Парадигма развития науки

Методологическое обеспечение

А. Е. Кононюк

ДИСКРЕТНО-НЕПРЕРЫВНАЯ МАТЕМАТИКА

Книга 9

Математическая логика

Часть 1

Киев
«Освіта України»

2017

УДК 51 (075.8)
ББК В161.я7
К213

Рецензенты:
В. В. Довгай — к-т физ.-мат. наук, доц. (Национальный тех—нический университет «КПІ»);
В. В. Гавриленко — д-р физ.-мат. наук, проф.,
О. П. Будя — к-т техн. наук, доц. (Киевский университет эко—номики, туризма и права);
Н. К. Печурин — д-р техн. наук, проф. (Национальный ави—ационный университет).

Кононюк А. Е.
К213 Дискретно-непрерывная математика. (Математическая логика). — В 12-и кн. Кн 9, ч.1— К.: 2017. — 580 с.

ISBN 978-966-373-693-8 (многотомное издание)
ISBN 978-966-373-694-5 (книга 9, ч.1)

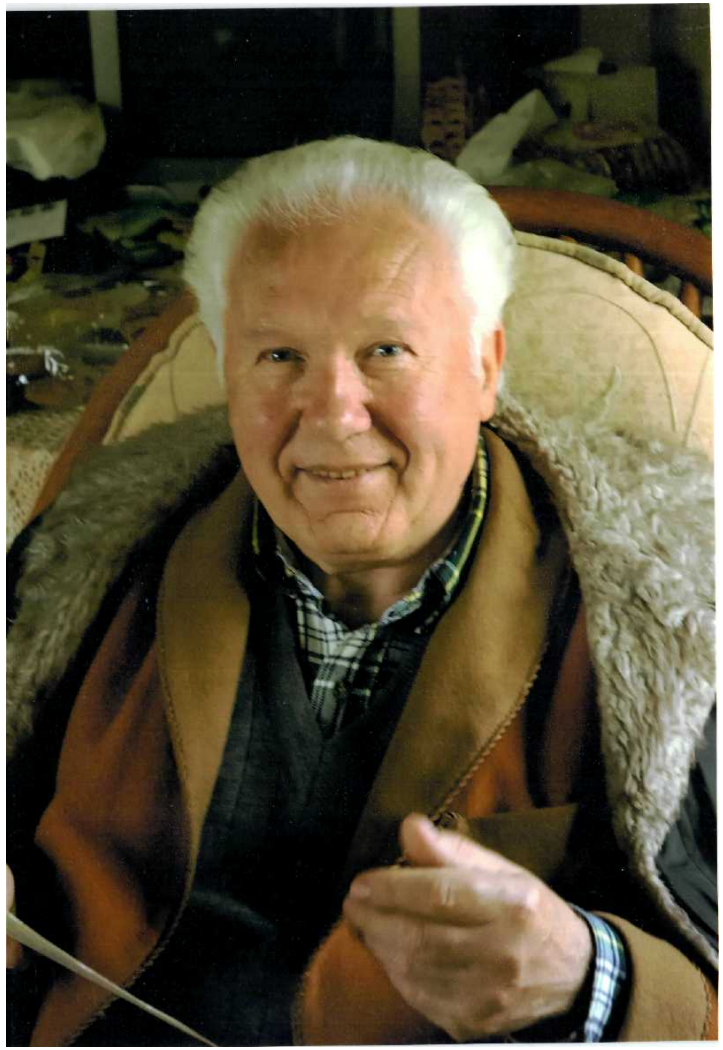
Многотомная работа содержит систематическое изложение математических дисциплин, исспользуемых при моделировании и исследованиях математических моделей систем.

В работе излагаются основы теории множеств, отношений, поверхностей, пространств, алгебраических систем, матриц, графов, математической логики, теории вероятностей и массового обслуживания, теории формальных грамматик и автоматов, теории алгоритмов, которые в совокупности образуют единную методолгически взамосвязанную математическую систему «Дискретно-непрерывная математика».

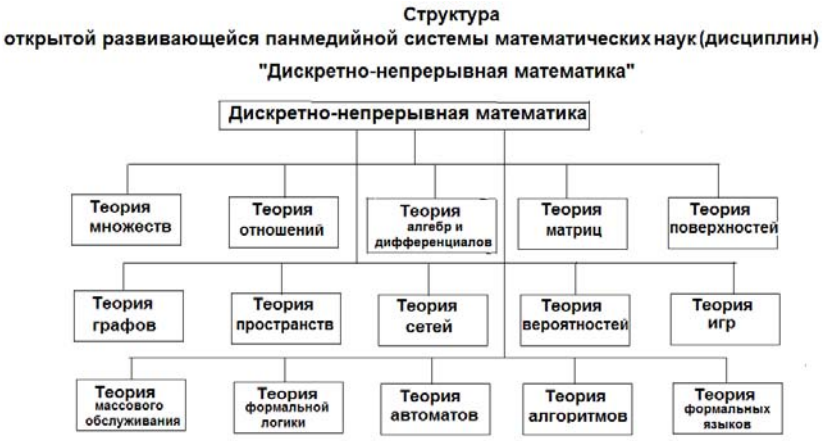
Для бакалавров, специалистов, магистров, аспирантов, докторантов и просто ученых и специалистов всех специальностей.

УДК 51 (075.8)
ББК В161.я7

ISBN 978-966-373-693-8 (многотомное издание)
ISBN 978-966-373-694-5 (книга 9, ч.1)



Кононюк Анатолий Ефимович



Оглавление

Часть первая
Двузначная логика

Часть вторая

Часть первая

Двузначная логика

1. Введение в математическую логику

1.1. Чем занимается математическая логика?

Логика как искусство рассуждений зародилась в глубокой древности. Начало науки о законах и формах мышления связывают с именем Аристотеля. Прошло два тысячелетия, прежде чем Лейбниц предложил ввести в логику математическую символику и использовать ее для общих логических построений. Эту идею последовательно реализовал в XIX столетии Джордж Буль и тем самым заложил основы *математической (символической) логики.*

Математическая логика — это математическая дисциплина, изучающая технику доказательств. Компьютеры, как и математики, требуют точности и строгости в определениях, описаниях, доказательствах и обоснованиях, чем они отличаются от обычных людей.

Главная цель применения в логике математической символики заключалась в том, чтобы свести операции с логическими заключениями к формальным действиям над символами. При этом исходные положения записываются формулами, которые преобразуются по определенным законам, а полученные результаты истолковываются в соответствующих понятиях.

Бурное развитие математической логики связано, прежде всего, с задачами обоснования математики, где она используется для доказательства непротиворечивости исходных понятий и правильности рассуждений и выводов математических теорий. Некоторые ученые даже склонны рассматривать логику как одну из наиболее общих наук, частью которой является сама математика.

Логика нашла широкое применение в технике при исследовании и разработке релейно-контактных схем, вычислительных машин, дискретных автоматов. Ее методы используются в теории

преобразования и передачи информации, теории вероятностей и комбинаторном анализе. Математическая логика внедрилась в такие нематематические области, как экономика, биология, медицина, психология, языкознание, право. Интенсивно развиваются специальные разделы математической логики, призванные обслуживать конкретные области науки и техники.

Столь энергичный выход математической логики за пределы математики объясняется тем, что ее аппарат легко распространяется на объекты самой общей природы, лишь бы только они характеризовались конечным числом состояний.

Математи́ческая ло́гика (*теоретическая логика*, *символическая логика*) — раздел математики, изучающий математические обозначения, формальные системы, доказуемость математических суждений, природу математического доказательства в целом, вычислимость и прочие аспекты оснований математики. В более широком смысле рассматривается как математизированная ветвь формальной логики — «*логика по предмету, математика по методу*», «*логика, развиваемая с помощью математических методов*».

На сегодня известны следующие виды существующих логик:

- двузначная или классическая логика, использующая для описания действительности только два крайних понятия: “истинно” и “ложно”;

- трехзначная логика, принесенная Я. Лукасевичем, он ввел понятие “парадоксальное” или “бессмысленное”, по которым нельзя сказать конкретно истинны они или ложны – промежуточные между крайностями предыдущего этапа (среднее между крайностями всегда неопределенно);

- многозначная логика, использующая такие неопределенные понятия, как “необходимо”, “возможно”, “случайно”.

Приведенная схема вполне однозначно показывает нам, что количество логических систем, существующих в Природе, определяется количеством цифр-координат в соответствующем ряду треугольника Паскаля, т.е. каждая цифра в соответствующем ряду является носителем того или иного вида логики, что позволяет сделать нам следующие выводы:

первичной является не двузначная, а **однозначная логика**, которая не рассматривается современной наукой;

предельное количество существующих логик определяется предельным количеством цифр-координат в 11-м ряду треугольника Паскаля – это значит, что количество существующих в Природе логических систем не должно превышать одиннадцати;

и то, что количество используемых системой логических систем (в данном случае человечеством) определяется этапом развития самой системы, т.е. уровень развития сознания любой системы на каждом очередном этапе будет определяться структурой самого этапа развития (монадность, диадность, триадность… многозначность). А это значит, что структура системы на том или ином этапе будет определять и структуру используемой ею логики. В свою очередь структура логики будет определять сущность и форму исповедуемых системой идей на соответствующем этапе ее развития. Прекрасной демонстрацией изложенного является изменение сознания ребенка в ходе его роста. В раннем детстве он исповедует одну логику и философию, в детском саду другую, в младших классах школы третью, в старших классах четвертую, после школы пятую и т.д.

Последнее обрисовывает нам следующий эволюционный процесс логического развития:

- на этапе монады (тезиса) системой используется однозначная логика. Последнюю можно определить простым принципом, характерным для любой вновь возникшей и бурно растущей системы – “вперед и только вперед”, невзирая ни на какие возражения, несогласия и препятствия окружающей среды. Других рассуждений на этом этапе не существует (именно так мыслит возникший из зерна росток, так мыслит появившийся на свет ребенок, так мыслил и любой завоеватель, основывающий новое государство, будь то Александр Македонский или Чингисхан, Франциско Писарро и Эрнан Кортес, Наполеон, Ленин, Сталин и пр., и так же до XX века мыслило и все человечество).

- на этапе Позитрониевой диады (антитезиса) используется двузначная логика крайностей – т.н. классическая логика, использующая только два полярно противоположных понятия: “истинно” и “ложно”, которые можно обозначить знаками “+” и “-“, “1” и “0”, “Да” и “Нет”, свет и

тьма, правое и левое и т.д. (количество знаков в двузначной математике). Этот этап ярко описан Аристотелем в его “принципе исключения третьего”, который утверждает что “нет ничего третьего (промежуточного) между членами противоречивой пары и предписывает считать истинным какую-либо одну из крайностей”. Следует отметить, что некоторые философские школы отрицают идентичность закона исключения третьего принципа с принципом двузначности. Хотя из структуры треугольника Паскаля четко видно, что это две различных формулировки одной и той же закономерности, а следовательно, мы вправе снять вопрос об их различии раз и навсегда!

- на этапе перехода к многозначной логике используется триадная логика третьего ряда треугольника Паскаля Я. Лукасевича, включающая наличие третьего принципа между двумя крайними суждениями в отличие от предыдущего этапа.

- многозначная логика от четвертой до десятой степени (от 4 до 11 ряда треугольника Паскаля) – логика использующая не только простые и однозначные определения, а все богатство современного человеческого мышления. Современная логика слагается из множества внутренне разнородных логических систем. Многозначные системы более богаты, чем двузначная логика: в первой имеются функции невыразимые во второй. Понятия в многозначной логике не кажутся достаточно ясными. Они неопределимы в двузначной логике. Так, если в двузначной логике имеется только четыре разные функции от одного аргумента (четыре координаты-элемента-пространства третьего ряда треугольника Паскаля), то в трехзначной логике их уже соответственно двадцать семь. Следовательно, в одиннадцатом ряду их будет 1111 = 285311670611, т.е. в 71327917652 раз больше чем четыре (285311670611 : 4). Из сопоставления этой величины с четырьмя функциями одного аргумента двузначной логики сразу же становится видна несовершенность привычной для нас классической логики, в рамки которой мы тщетно пытаемся запихать все существующие истины.

Все это позволяет сделать следующий кардинальный вывод: для каждого этапа эволюции существует своя логика и то, что истинно для одного этапа развития является ложным для другого. Любая система в ходе эволюции проходит последовательно все этапы своего развития в полном соответствии с рядами треугольника Паскаля.

Эволюция структуры логики по треугольнику Паскаля.

1 - однозначная (монадная) логика

1+1 - двузначная логика: либо “ложь”, либо “истина”

1+2+1 - трехзначная логика – введение “парадоксального” между крайностями

1+3+3+1

1+4+6+4+1

1+5+10+10+5+1

1+6+15+20+15+6+1 многозначная логика, использующая понятия,

1+7+21+35+35+21+7+1 которые не являются достаточно конкретными,

1+8+28+56+70+56+28+8+1 такие как необходимо, возможно, случайно и т.д.

1+9+36+84+126+126+84+36+9+1

1+10+45+120+210+252+210+120+45+10+1

Устоявшееся представление о математической логике как науке, изучающей законы мышления с применением аппарата математики,

главным образом, для нужд самой математики, в современных условиях становится слишком узким. С расширением областей применения и дальнейшим развитием математической логики изменяется и взгляд на нее. Объектами математической логики являются любые дискретные конечные системы, а ее главная задача — структурное моделирование таких систем.

1.2. Основные положения

Первое дошедшее до нас сочинение по формальной логике — "Аналитики" Аристотеля (384-322 гг. до нашей эры). В них рассматриваются основы силлогистики — правила вывода одних высказываний из других. Так из высказываний "Все христиане — люди" и "Все люди — живые существа" можно сделать вывод, что все христиане — живые существа. Однако на практике такие случаи встречаются крайне редко.

Вопрос о создании символической логики как универсального научного языка рассматривал Лейбниц в 1666 году в работе «Искусство комбинаторики» (*De arte combinatoria*). Он думал о записи высказываний на специальном языке, чтобы затем по логическим законам вычислять истинность других. В середине XIX века появились первые работы по алгебраизации аристотелевой логики, сформировавшие первооснову исчисления высказываний (Буль, де Морган, Шрёдер). В работах Фреге и Пирса (конец 1870-х — начало 1880-х) в логику введены предметные переменные, кванторы и, тем самым, основано исчисление предикатов. В конце 1880-х годов Дедекинд и Пеано применили эти инструменты в попытках аксиоматизации арифметики, при этом Пеано создал удобную систему обозначений, закрепившуюся и в современной математической логике.

Уайтхед и Рассел создают в 1910—1913 годах трактат Principia Mathematica, который оказал исключительное влияние на все последующее развитие математической логики. Ещё одной важной вехой в развитии логики стало обнаружение свойственных уровню развития логических исчислений и теории множеств конца XIX века парадоксов, в преодоление которых появилась концепция интуиционизма и интуиционистская логика (Брауэр, 1908) и, в качестве альтернативы, Гильбертом создана программа обоснования

математики посредством аксиоматической формализации с использованием строго ограниченных средств, не приводящих к противоречиям.

Применение в логике математических методов становится возможным тогда, когда суждения формулируются на некотором точном языке. Такие точные языки имеют две стороны: синтаксис и семантику. Синтаксисом называется совокупность правил построения объектов языка (обычно называемых формулами). Семантикой называется совокупность соглашений, описывающих наше понимание формул (или некоторых из них) и позволяющих считать одни формулы верными, а другие — нет.

Важную роль в математической логике играют понятия дедуктивной теории и исчисления. Исчислением называется совокупность правил вывода, позволяющих считать некоторые формулы выводимыми. Правила вывода подразделяются на два класса. Одни из них непосредственно квалифицируют некоторые формулы как выводимые. Такие правила вывода принято называть аксиомами. Другие же позволяют считать выводимыми формулы А, синтаксически связанные некоторым заранее определённым способом с конечными наборами $A_1,\ldots,A_n$ выводимых формул. Широко применяемым правилом второго типа является правило modus ponens: если выводимы формулы А и (А→В), то выводима и формула В.

Отношение исчислений к семантике выражается понятиями семантической пригодности и семантической полноты исчисления. Исчисление И называется семантически пригодным для языка Я, если любая выводимая в И формула языка Я является верной. Аналогично, исчисление И называется семантически полным в языке Я, если любая верная формула языка Я выводима в И.

Многие из рассматриваемых в математической логике языков обладают семантически полными и семантически пригодными исчислениями. В частности, известен результат Курта Гёделя о том, что классическое исчисление предикатов является семантически полным и семантически пригодным для языка классической логики предикатов первого порядка (теорема Гёделя о полноте). С другой стороны, имеется немало языков, для которых построение семантически полного и семантически пригодного исчисления невозможно. В этой области классическим результатом является

теорема Гёделя о неполноте, утверждающая невозможность семантически полного и семантически пригодного исчисления для языка формальной арифметики.

На практике множество элементарных логических операций является обязательной частью набора инструкций всех современных микропроцессоров и, соответственно, входит в языки программирования. Это является одним из важнейших практических приложений методов математической логики, изучаемых в современных учебниках информатики.

Разделы математической логики

В Математической предметной классификации математическая логика объединена в одну секцию верхнего уровня с основаниями математики, в которой выделены следующие разделы:

- общая логика (англ. *general logic*), включает классическую логику первого порядка, логики высших порядков (логику второго порядка), комбинаторную логику, λ-исчисление, временную логику, модальную логику, многозначные логики, нечёткую логику, логику в информатике;
- теория моделей;
- теория вычислимости и теория рекурсии;
- теория множеств;
- теория доказательств и конструктивная математика;
- алгебраическая логика (включает вопросы изучения булевых алгебр, алгебр Гейтинга, квантовых логик, цилиндрических и полиадических алгебр, алгебр Поста);
- нестандартные модели.

1.3. Булевы функции.

Объекты с двумя возможными состояниями характеризуются *булевыми переменными,* которые способны принимать лишь два различных значения. Для обозначения этих двух значений обычно используются цифры 0 и 1 или буквы Л (ложно) И (истинно).

Отношения между булевыми переменными представляются *булевыми функциями,* которые подобно числовым функциям могут зависеть от одной, двух и, вообще, *n* переменных (аргументов). Запись $y = f(x_1, x_2, ..., x_n)$ означает , что y — функция аргументов $x_1, x_2, ..., x_n$. Важнейшая особенность булевых функций состоит в том, что они, как и их аргументы, принимают свои значения из двухэлементного множества {0,1}, или {И, Л}, т. е. характеризуются одним из двух возможных состояний.

Функции небольшого числа переменных можно задавать с помощью таблиц, подобных таблицам сложения и умножения одноразрядных чисел. Для этого нужно только указать значения функции для каждой комбинации значений ее аргументов. Основными в двузначной логике являются следующие три функции.

Отрицание — функция $y = f(x)$, принимающая значения 1, когда $x = 0$, и значение 0, когда $x = 1$; она обозначается $y = \overline{x}$ (читается «не *x*»).

Дизъюнкция — функция $y = f(x_1, x_2)$, принимающая значение 0 тогда и только тогда, когда оба аргумента имеют значение 0; она обозначается $y = x_1 \vee x_2$ (читается «x_1 или x_2»).

Конъюнкция — функция $y = f(x_1, x_2)$, принимающая значение 0 тогда и только тогда, когда оба аргумента имеют значение 1; она обозначается $y = x_1 \wedge x_2$ (читается «x_1 и x_2»).

Таблицы для этих функций имеют вид:

x	$\overline{x}$
0	1
1	0

$x_1 \vee x_2$

x_1	x_2	
	0	1
0	0	1
1	1	1

$x_1 \wedge x_2$

x_1	x_2	
	0	1
0	0	0
1	0	1

1.4. Логические операции и формулы

Булевы функции можно рассматривать как *логические операции* над величинами, принимающими два значения — 0 и 1. Отрицание— это *одноместная* операция, а дизъюнкция и конъюнкция — *двухместные операции.* При этом выражения $\overline{x}$, $x_1 \vee x_2$, $x_1 \wedge x_2$ являются *логическими формулами.*

Более сложные формулы получаются замещением входящих в них переменных другими логическими формулами, которые обычно заключаются в скобки. Например, положив $x_1=\overline{a}$ и $x_2=b\wedge c$ из $x_1\vee x_2$, имеем $(\overline{a})\vee(b\wedge c)$. Каждая формула определяет некоторую булеву функцию. Ее значение при различных значениях переменных определяется на основании таблиц функций, приведенных в (2). Так, при $a=0$, $b=1$ и $c=0$ имеем: $x_1=a=0=1$,

$x_2=b\wedge c=1\wedge 0=0$ и $x_1\vee x_2=a\vee(b\wedge c)=1\vee 0=1$.

Аналогично получаем значения функции и при других комбинациях значений аргументов.

Две функции (как и определяющие их формулы) считаются *равносильными* при любых значениях аргументов эти функции (формулы) принимают одинаковые значения. Равносильные функции соединяются знаком равенства, например: $(x\wedge y)\vee\bar{z}=$

$=\overline{(\bar{x}\vee\bar{y})}\wedge z$ или $((x\vee\bar{x})\wedge y)\vee(y\vee x)=x\vee y$.

Равносильность функций проверяется по таблицам основных операций, причем необходимо сравнить их значения для всех комбинаций значений переменных.

1.5. Булева алгебра

Булевой алгеброй называется непустое множество A с двумя бинарными операциями $\wedge$ (аналог конъюнкции), $\vee$ (аналог дизъюнкции), одной унарной операцией $\neg$ (аналог отрицания) и двумя выделенными элементами: 0 (или Ложь) и 1 (или Истина) такими, что для всех a, b и c из множества A верны следующие аксиомы:

$a\vee(b\vee c)=(a\vee b)\vee c$ ассоциативность

$a\wedge(b\wedge c)=(a\wedge b)\wedge c$

$a\vee b=b\vee a$ $a\wedge b=b\wedge a$ коммутативность

$a\vee(a\wedge b)=a$ $a\wedge(a\vee b)=a$ законы поглощения

$a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c)$
$a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$ дистрибутивность

$a\vee\neg a=1$ $a\wedge\neg a=0$ дополнительность

Первые три аксиомы означают, что $(A,\wedge,\vee)$ является решёткой. Таким образом, булева алгебра может быть определена как дистрибутивная решётка, в которой выполнены две последние аксиомы. Структура, в которой выполняются все аксиомы, кроме предпоследней, называется псевдобулевой алгеброй. Названа в честь Джорджа Буля.

1.5.1. Некоторые свойства

Из аксиом видно, что наименьшим элементом является 0, наибольшим является 1, а дополнение $\neg a$ любого элемента a однозначно определено. Для всех a и b из A верны также следующие равенства:

$a\vee a=a$ $a\wedge a=a$

$a\vee 0=a$ $a\wedge 1=a$

$a\vee 1=1$ $a\wedge 0=0$

$\neg 0=1$ $\neg 1=0$ дополнение 0 есть 1 и наоборот

$\neg(b\vee c)=\neg a\wedge\neg b$ $\neg(b\wedge c)=\neg a\vee\neg b$ законы де Моргана

$\neg\neg a=a$ инволютивность отрицания, закон снятия двойного отрицания.

Примеры

- Самая простая нетривиальная булева алгебра содержит всего два элемента, 0 и 1, а действия в ней определяются следующей таблицей:

$\wedge$	**0**	**1**
0	0	0
1	0	1

$\vee$	**0**	**1**
0	0	1
1	1	1

a	**0**	**1**
¬a	1	0

- Эта булева алгебра наиболее часто используется в логике, так как является точной моделью классического исчисления высказываний. В этом случае 0 называют *ложью*, 1 — *истиной*. Выражения, содержащие булевы операции и переменные, представляют собой высказывательные формы.
- Алгебра Линденбаума — Тарского (фактормножество всех утверждений по отношению равносильности в данном исчислении с соответствующими операциями) какого-либо исчисления высказываний является булевой алгеброй. В этом случае истинностная оценка формул исчисления является гомоморфизмом алгебры Линденбаума — Тарского в двухэлементную булеву алгебру.
- Множество всех подмножеств данного множества S образует булеву алгебру относительно операций $\wedge := \bigcup$ (объединение), $\vee := \cap$ (пересечение) и унарной операции дополнения. Наименьший элемент здесь — пустое множество, а наибольший — всё S.
- Если R — произвольное кольцо, то на нём можно определить множество *центральных идемпотентов* так:
  $A = \{ e \in R : e^2 = e, ex = xe, \forall x \in R \}$,
  тогда множество A будет булевой алгеброй с операциями $e \vee f := e + f - ef$ и $e \wedge f := ef$.

1.5.2. Принцип двойственности

В булевых алгебрах существуют двойственные утверждения, они либо одновременно верны, либо одновременно неверны. Именно, если в

формуле, которая верна в некоторой булевой алгебре, поменять все конъюнкции на дизъюнкции, 0 на 1, $\leq$ на $>$ и наоборот или $<$ на $\geq$ и наоборот, то получится формула, также истинная в этой булевой алгебре. Это следует из симметричности аксиом относительно таких замен.

1.5.3. Представления булевых алгебр

Можно доказать, что любая конечная булева алгебра изоморфна булевой алгебре всех подмножеств какого-то множества. Отсюда следует, что количество элементов в любой конечной булевой алгебре будет степенью двойки.

Теорема Стоуна утверждает, что любая булева алгебра изоморфна булевой алгебре всех открыто-замкнутых множеств какого-то компактного вполне несвязного хаусдорфова топологического пространства.

1.5.4. Аксиоматизация

В 1933 году американский математик Хантингтон предложил следующую аксиоматизацию для булевых алгебр:

1. *Аксиома коммутативности*: $x + y = y + x$.
2. *Аксиома ассоциативности*: $(x + y) + z = x + (y + z)$.
3. *Уравнение Хантингтона*: $n(n(x) + y) + n(n(x) + n(y)) = x$.

Здесь использованы обозначения Хантингтона: + означает дизъюнкцию, n — отрицание.

Герберт Роббинс поставил следующий вопрос: можно ли сократить последнюю аксиому так, как написано ниже, то есть будет ли определённая выписанными ниже аксиомами структура булевой алгеброй?
Аксиоматизация алгебры Роббинса:

1. *Аксиома коммутативности*: $x + y = y + x$.
2. *Аксиома ассоциативности*: $(x + y) + z = x + (y + z)$.
3. *Уравнение Роббинса*: $n(n(x + y) + n(x + n(y))) = x$.

Этот вопрос оставался открытым с 1930-х годов и был любимым вопросом Тарского и его учеников.

В 1996 году Вильям МакКьюн, используя некоторые полученные до него результаты, дал утвердительный ответ на этот вопрос. Таким образом, любая алгебра Роббинса является булевой алгеброй.

Множество всех булевых функций вместе с операциями отрицания, конъюнкции и дизъюнкции образует *булеву алгебру.*

На основе определения основных операций нетрудно убедиться в справедливости следующих тождеств (свойств) булевой алгебры:

коммутативность

$$x \vee y = y \vee x, \quad x \wedge y = y \wedge x;$$

ассоциативность

$$x \vee (y \vee z) = (x \vee y) \vee z, \quad x \wedge (y \wedge z) = (x \wedge y) \wedge z;$$

дистрибутивность

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \quad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z);$$

свойство констант

$$x \vee 0 = x, \quad x \wedge 1 = x;$$

свойство отрицания

$$x \vee \bar{x} = 1, \quad x \wedge \bar{x} = 0.$$

1.6. Тождественные преобразования

Приведенные свойства позволяют получить ряд других важных законов и тождеств уже без обращения к таблицам соответствия: $\overline{x \vee y} = \bar{x} \wedge \bar{y}$, $\overline{x \wedge y} = \bar{x} \vee \bar{y}$ *(законы де Моргана),* $x \vee (x \wedge y) = x \wedge (x \vee y) = x$ *(законы поглощения)* $x \vee x = x \wedge x = x$ *)законы идемпотентности),* а также тождества

$$x \vee (\bar{x} \wedge y) = x \vee y; \; (x \wedge y) \vee (x \wedge z) \vee (y \wedge \bar{z}); = (x \wedge z) \vee (y \wedge \bar{z}); \quad \bar{\bar{x}} = x; \quad \bar{1} = 0; \quad \bar{0} = 1; \quad x \vee 1 = 1; \quad x \wedge 0 = 0$$

и т. д.

Так, законы идемпотентности доказываются следующими преобразованиями:

$$x \vee x = (x \vee x) \wedge 1 = (x \vee x) \wedge (x \vee \bar{x}) = x \vee (x \wedge \bar{x}) = x \vee 0 = x; \quad x \wedge x = (x \wedge x) \vee 0 = (x \wedge x) \vee (x \wedge \bar{x}) = x \wedge (x \vee x) = x \wedge 1 = x.$$

Используя

$$x \vee 1 = x \vee (x \vee \bar{x}) = (x \vee x) \vee \bar{x} = x \vee \bar{x} = 1; \quad x \wedge 0 = x \wedge (x \wedge \bar{x}) = x \wedge \bar{x} = 0.$$

полученные соотношения, имеем:

Доказательство законов поглощения имеет вид:

$$x \vee (x \wedge y) = (x \wedge 1) \vee (x \wedge y) = x \wedge (1 \vee y) = x \wedge 1 = x; \; x \wedge (x \vee y) = (x \vee 0) \wedge (x \vee y) = x \vee (0 \wedge y) = x \vee (y \wedge 0) = x \vee 0 = x.$$

Соотношение $\bar{x} = x$ доказывается следующим образом: из $x \vee \bar{x} = 1$ по закону коммутативности следует $\bar{x} \vee x = 1$, откуда сравнением с $\bar{x} \vee \bar{x} = 1$ имеем $x = \bar{x}$.

Интересно доказательство закона де-Моргана. На основании свойств отрицания равенство функций $\overline{x \vee y}$ и $\bar{x} \wedge \bar{y}$ должно означать, что $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = 1$ и $(x \vee y) \wedge (\bar{x} \wedge \bar{y}) = 0$. Действительно,

$$(x \vee y) \vee (\bar{x} \wedge \bar{y}) = ((x \vee y) \vee \bar{x}) \wedge ((x \vee y) \vee \bar{y}) = ((x \vee \bar{x}) \vee (y \vee \bar{y})) \wedge (x \vee (y \vee \bar{y})) = (1 \vee y) \wedge (x \vee 1) = 1 \wedge 1 = 1,$$

а также

$$(x \vee y) \wedge (\bar{x} \wedge \bar{y}) = (x \wedge (\bar{x} \wedge \bar{y})) \vee (y \wedge (\bar{x} \wedge \bar{y})) = ((x \wedge \bar{x}) \wedge \bar{y}) \vee ((y \wedge \bar{y}) \wedge \bar{x}) = (0 \wedge \bar{y}) \vee (0 \wedge \bar{x}) = (\bar{y} \wedge 0) \vee (\bar{x} \wedge 0) = 0 \vee 0 = 0.$$

Следовательно, соотношение

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}$$

доказано. Аналогично доказывается и второй закон.

1.7. Упрощение записи формул

Операции дизъюнкции и конъюнкции удовлетворяют законам коммутативности и ассоциативности. Поэтому если переменные или формулы связаны только посредством одной из этих операций, то их можно выполнять в любом порядке, а формулы записывать без скобок. Например: $((x_1 \vee x_2) \vee (x_3 \vee x_4)) \vee x_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$, а также $(x_1 \wedge x_2) \wedge (x_3 \wedge (x_4 \wedge x_5)) = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5$.

Если считать, что операция конъюнкции должна предшествовать операции дизъюнкции (конъюнкция связывает сильнее дизъюнкции), то можно опустить скобки, в которые заключены формулы со знаком конъюнкции. При наличии скобок в первую очередь должны выполняться операции внутри скобок, независимо от их старшинства. Обычно опускают также скобки, в которые заключены формулы со знаком отрицания.

Еще одно упрощение связано с символикой. Знак конъюнкции в формулах можно опустить и вместо $x \wedge y$ писать xy. Операцию

конъюнкции часто называют *логическим умножением,* а операцию дизъюнкции — *логическим сложением.*

С учетом приведенных условий запись существенно упрощается. Например, формуле $(x \wedge (y \wedge \bar{z})) \vee ((\overline{x \vee y}) \wedge z)$ соответствует запись $xy\bar{z} \vee \overline{x \vee y}\, z$.

1.8. Переключательные схемы

В качестве одной из интерпретаций булевых функций рассмотрим электрическую схему, состоящую из источника напряжения (батареи), лампочки и одного или двух ключей (x_1 и x_2). Ключи управляются кнопками с двумя состояниями: кнопка нажата (1) и кнопка отпущена (0). Если в исходном состоянии ключ разомкнут, то при нажатии кнопки он замыкается.

Ключ может быть сконструирован и так, что в исходном состоянии он замкнут, тогда нажатие кнопки означает его размыкание, т. е. приводит к противоположному результату. Поэтому нормально замкнутые ключи обозначим через $\overline{x}_1$ и $\overline{x}_2$.

При соответствующих состояниях кнопок лампочка принимает одно из двух состояний: горит (1) и не горит (0). Состояния кнопок отождествляются со значениями булевых переменных x_1 и x_2, а состояние лампочки — со значением функций этих переменных.

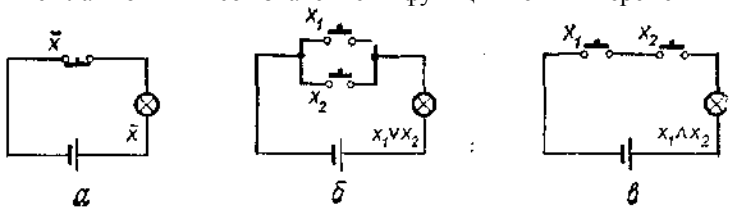


Рис. 1. Переключательные схемы, соответствующие операциям отрицания (а), дизъюнкции *(б)* и конъюнкции (в).

Операции отрицания соответствует схема с одним нормально замкнутым ключом (рис. 1, *а).* Если кнопка нажата *(x* = 1), ключ разомкнут и лампочка не горит, т. е. *f(x)*= 0; при отпущенной кнопке *(x* = 0) ключ замкнут и лампочка горит, т. е. *f(x)* = 1. Операциям дизъюнкции и конъюнкции соответствуют схемы с двумя нормально разомкнутыми ключами (рис. 1, *б, в*). Легко убедиться, что в схеме рис. 1, *б* лампочка горит при нажатии хотя бы одной из кнопок, а в схеме рис. 1, *в* — только при нажатии обеих кнопок одновременно.

Любую сложную булеву функцию можно представить некоторой переключательной схемой. На рис. 2,*а* показана схема, реализующая функцию $y = x_1\bar{x}_2 \vee \bar{x}_1x_2x_3 \vee x_2x_4$. Та же функция представляется равносильной формулой $y = x_1\bar{x}_2 \vee (\bar{x}_1x_2 \vee x_4)x_3$, которой соответствует другая более простая схема (рис. 2, *б).* Следует иметь в виду, что ключи, обозначенные одинаковыми буквами *(x* или $\overline{x}$), связаны между собой и управляются общей кнопкой.

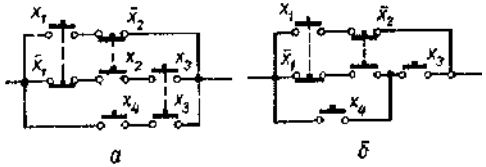


Рис. 2. Переключательная схема, реализующая логическую функцию *(а),* и упрощенная схема *(б).*

В реальных устройствах используются ключи различной конструкции и физической природы (механические, электромагнитные, электронные, гидравлические, пневматические в т. д.) Однако при реализации логических функций многие технические особенности не имеют значения. Существенными свойствами контактных схем являются исходные положения ключей (нормально разомкнуты или нормально замкнуты) и способ их соединения между собой и внешними устройствами. Эта информация полностью отображается графом, ребра которого соответствуют ключам, а вершины — точкам их соединения. Ребра нормально разомкнутых ключей обозначаются соответствующей переменной *(x),* а нормально замкнутых — отрицанием переменной ($\overline{x}_1$). Например, контактная схема (рис. 2, *б*) изображается графом, как показано на рис. 3, *а.*

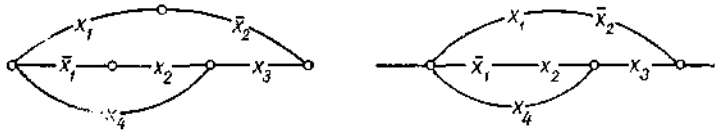


Рис. 3. Граф переключательной схемы *(а)* и его упрощенное изображение *(б).*

При изображении контактных схем графами принимаются некоторые специфические условия и упрощения. Обычно переменные обозначаются в разрывах линий, изображающих ребра. При этом ребрами считаются только такие линии, которые обозначены какой-либо переменной или ее отрицанием. Другие линии, не являющиеся ребрами графа, могут изображать входы и выходы схемы, связи с другими схемами и т. п. Кроме того, вершины второй степени могут не изображаться, так как им инцидентны пары последовательно соединенных ребер, из которых каждое обозначено соответствующей переменной. На рис. 3, *б* показана контактная схема в обычно принятом виде.

1.9. Высказывания

Пусть x_1 и x_2 — некоторые *высказывания,* которые могут быть истинными (1) или ложными (0), например: «Я пойду в театр» *(x_1)* и «Я встречу друга» *(x_2).* Дизъюнкцией $x_1 \vee x_2$ является сложное высказывание «Я пойду в театр *или* встречу друга», а конъюнкцией $x_1 \wedge x_2$ — высказывание «Я пойду в театр *и* встречу друга».

Ясно, что если высказывание истинно, то его отрицание ложно. Сложное высказывание, образованное дизъюнкцией двух высказываний, истинно при условии, что истинно хотя бы одно из них. Сложное высказывание, образованное конъюнкцией двух истинных высказываний истинно, если истинны оба эти высказывания одновременно.

Итак, высказывания можно рассматривать как двоичные переменные, а связки «не», «или», «и», с помощью которых образуются сложные высказывания,— как операции над этими переменными. В алгебре высказываний используются еще две операции: *имплинация* $x_1 \rightarrow x_2$, соответствующая связке «если, то» и *эквиваленция* $x_1 \sim x_2$, соответствующая связке «если и только если». Они задаются следующими таблицами:

$x_1 \rightarrow x_2$

x_1	x_2 = 0	x_2 = 1
0	1	1
1	0	1

$x_1 \sim x_2$

x_1	x_2 = 0	x_2 = 1
0	1	0
1	0	1

В нашем примере импликацией будет высказывание: *«Если* я пойду в театр, *то* встречу друга», а эквиваленцией— «Я пойду в театр, *если и только если* встречу друга». Как видно из таблиц, импликация высказываний ложна только в случае, когда первое из простых высказываний истинно, а второе ложно. Эквиваленция является истинным высказыванием, если оба простые высказывания истинны или ложны одновременно.

Обозначив буквами простые высказывания, можно представить сложное высказывание формулой с помощью соответствующих связок. Например, высказыванию *«Если* давление масла на шарик клапана больше усилия его пружины (x_1), *то* масло открывает клапан (x_2) *и* частично перетекает из нагнетательной полости во впускную (x_3)*»* соответствует формула $x_1 \rightarrow x_2x_3$.

1.10. Предикаты

Обычно высказывания выражают свойства одного или нескольких объектов. Содержательная часть высказывания играет роль определяющего свойства совокупности объектов, для которых это высказывание истинно, и называется *предикатом.* Например, высказывание «Иванов — отличник» истинно или ложно в зависимости от оценок, которые имеет данный студент. В то же время предикат *«х* — отличник» определяет подмножество отличников на некотором множестве студентов (группа, курс, факультет). Подставив вместо *х* фамилии студентов, получим множество высказываний. Совокупность истинных высказываний и будет соответствовать подмножеству отличников,

Предикат представляет собой логическую функцию *P(x),* принимающую, как и булевы функции, значение 0 или 1, но в отличие от них, значения аргумента *х* выбираются из некоторого множества *M*

объектов *(x ∈ M)*. В общем случае такая функция может зависеть от многих аргументов $x_1, x_2, ..., x_n$, принимающих значения из одного и того же или различных множеств. Ее записывают $P(x_1,x_2, ..., x_n)$ и называют *n-местным предикатом*. Например: «*x*— четное число», «*x* — компонент цепи» —одноместные предикаты *P(x); «x* брат *y», «x* меньше *y»* — двуместные предикаты *P(x, y); «x* и *y* — родители *z*», «*x* — сумма *y* и *z*» — трехместные предикаты *P(x,y, z)* и т. д. Если аргументы $x_1, x_2.....x_n$ замещены конкретными значениями (объектами), то предикат, переходит в высказывание, которое рассматривают как *O-местный предикат*.

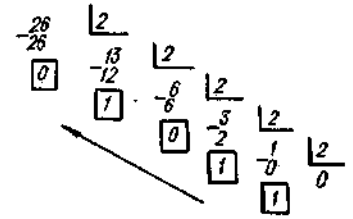
Так как предикаты способны принимать только значения 0 и 1, то их, как и булевы переменные, можно связывать логическими операциями. В результате получаем формулы, определяющие более сложные предикаты. Так, если *P(x)* означает *«x* — инженер», а *Q(x)* — «*x* — сотрудник нашего отдела», то $P(x) \wedge Q(x) = R(x)$ есть одноместный предикат «*x* — инженер и сотрудник нашего отдела» или проще «*x* — инженер нашего отдела». Очевидно, если *P* — множество инженеров, а *Q* — множество сотрудников данного отдела, то этот предикат соответствует пересечению $P \bigcap Q$. Таким образом, имеет место тесная связь между логикой предикатов и операциями над множествами.

1.11. Двоичная арифметика.

В позиционной системе счисления с основанием *m* любое целое неотрицательное число *a* записывается последовательностью различных цифр $x_1x_2 ... x_n$, что означает $a = x_1m^{n-1} + x_2m^{n-2} + ... + x_nm^0$. Десятичная система использует цифры 0, 1, ..., 9, например: $2907 = 2 \cdot 10^3 + 9 \cdot 10^2 + 0 \times 10 + 7 \cdot 10^0$. Для двоичной системы счисления достаточно двух цифр, которые обозначаются 0 и 1. При этом последовательность $x_1x_2 ... x_n$ таких цифр является записью двоичного *n*-разрядного числа

$$x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + ... + x_n \cdot 2^0.$$

Перевод целых десятичных чисел в двоичные осуществляется последовательным делением исходного числа и каждого частного на два. Получаемые при этом остатки (0 или 1), записанные в обратном порядке, и дают представление десятичного числа в двоичной системе счисления. Например:

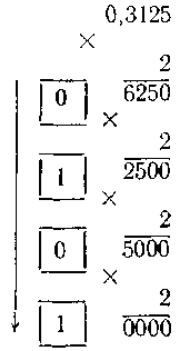


$$26_{10} = 11010_2.$$

Действительно, проверяя полученный результат, получаем

$$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 8 + 2 = 26.$$

Дробное число переводится в двоичную систему счисления методом последовательного умножения на два. При этом каждый раз после запятой двоичного числа записывается 0 или 1 соответственно целой части результата умножения. Последовательное умножение продолжается до тех пор, пока дробная часть не обратится в нуль или пока не получим требуемое количество двоичных знаков после запятой. Например, двоичное представление числа 0,3125 получается следующим образом:



$$0{,}3125_{10} = 0{,}0101_2.$$

Проверка полученного результата дает:

$$0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = \frac{1}{4} + \frac{1}{16} = \frac{5}{16} = 0{,}3125.$$

Если число является смешанным, т. е. его целая и дробная части отличны от нуля, то оно переводится в двоичную систему раздельно: целая часть — последовательным делением, а дробная — последовательным умножением.

Арифметические операции над числами сводятся к операциям сложения и умножения одноразрядных чисел. В двоичной системе счисления умножение задается таблицей конъюнкции:

$$0 \cdot 0 = 0;\ 1 \cdot 0 = 0;\ 0 \cdot 1 = 0 \text{ и } 1 \cdot 1 = 1.$$

Сложение выполняется по правилу:

$0 + 0 = 0;\ 1 + 0 = 1;\ 0 + 1 = 1$ и $1 + 1 = 10$

(10— это двоичное число, соответствующее десятичному числу 2). Операции над двоичными числами выполняются по правилам, аналогичным для десятичных чисел, но эти правила предельно упрощаются (особенно для умножения). Например, десятичные операции $4 + 27 = 68$ и $41 \cdot 5 = 205$ выгладят следующим образом:

$$\begin{array}{r} 101001 \\ +\ 11011 \\ \hline 1000100 \end{array} \qquad \begin{array}{r} \times\ 101001 \\ \quad\ 101 \\ \hline \quad 101001 \\ +\ 101001 \\ \hline 11001101 \end{array}.$$

Как видно, умножение двоичных чисел сводится к сложению чисел, образованных сдвигом влево первого сомножителя. Поразрядное сложение осуществляется в соответствии с таблицей

x_1 \ x_2	0	1
0	0	1
1	1	0

,

причем в случае $x_1 = x_2 = 1$ образуется единица переноса в старший разряд. Операция, задаваемая этой таблицей, называется *сложением по модулю 2.* Если при сложении перенос не учитывается, то эта операция вместе с операцией умножения определяет на множестве двоичных чисел *арифметику по модулю 2.*

1.12. Логическая арифметика

Строго говоря, булева арифметика оперирует на множествах $\mathbf{Z}_2$ и $\mathbf{Z}^n_2$ и, следовательно, включает только числа 0 и 1. Для того чтобы подчеркнуть такую структуру, начнем с рассмотрения логической арифметики на «относительно большом» множестве $\mathbf{Z}_5$. Она дает основу многозначной логики. Отсюда легко получить более простой случай $\mathbf{Z}_2$. Возьмем множество $\mathbf{Z}_5 = (0, 1, 2, 3, 4\}$ и операции $\vee$ и $\wedge$, которые определены в табл. 1.

Таблица 1

$\vee$	0	1	2	3	4
0	0	1	2	3	4
1	1	1	2	3	4
2	2	2	2	3	4
3	3	3	3	3	4
4	4	4	4	4	4

$\wedge$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

Упорядочивая $\mathbf{Z}_5$ обычным образом (порядок индуцируется **Z** и **R**), видим, что

$$a \vee b = \max\{a, b\},$$
$$a \wedge b = \min\{a, b\}.$$

Обе операции коммутативны и ассоциативны, 0 является единицей для $\vee$, а 4 является единицей для $\wedge$; $\wedge$ дистрибутивна по отношению к $\vee$, но не наоборот.

Пример 1. Возьмем множество $\mathbf{Z}_m$ с естественным порядком элементов. Введем операции $\wedge$ и $\vee$. Рассмотрим шесть возможных случаев упорядочивания трех произвольных элементов *a, b, c* с $\mathbf{Z}_m$:

(I) $a \le b \le c;$
(II) $a \le c \le b;$
(III) $b \le a \le c;$
(IV) $b \le c \le a;$
(V) $c \le a \le b;$
(VI) $c \le b \le a.$

Использование символа $\le$ является интуитивным, однако может быть обосновано с помощью следующего определения:

$a \le b$ тогда и только тогда, когда $a \vee b = b.$

Для проверки условия дистрибутивности нужно показать, что

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

Это можно сделать проверкой того, что обе части выражения совпадают для каждого из наборов *a, b* и *c*. Будем одновременно вычислять и сопоставлять соответствующие выражения:

(I) $a \wedge (b \vee c) = a \wedge c = a,$
$(a \wedge b) \vee (a \wedge c) = a \vee a = a;$
(II) $a \wedge (b \vee c) = a \vee b = a,$
$(a \wedge b) \vee (a \wedge c) = a \vee a = a;$
(III) $a \wedge (b \vee c) = a \wedge c = a,$
$(a \wedge b) \vee (a \wedge c) = b \vee a = a;$
(IV) $a \wedge (b \vee c) = a \wedge c = c,$
$(a \wedge b) \vee (a \wedge c) = b \vee c = c;$

(V) $a \wedge (b \vee c) = a \wedge b = a$,
$(a \wedge b) \vee (a \wedge c) = a \vee c = a$;
(VI) $a \wedge (b \vee c) = a \wedge b = b$,
$(a \wedge b) \vee (a \wedge c) = b \vee c = b$.

Следовательно, $\wedge$ дистрибутивна по отношению к $\vee$.

Можно также показать, что $\vee$ дистрибутивна по отношению к $\wedge$, т.е. что

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

Проверку этого свойства оставляем как упражнение.

Перед тем как закончить обсуждение общего случая, давайте вернемся к табл. 11, определяющим $\vee$ и $\wedge$. Элементы, имеющие одинаковые значения в таблицах, расположены относительно единичных элементов так, как показано на рис. 4.

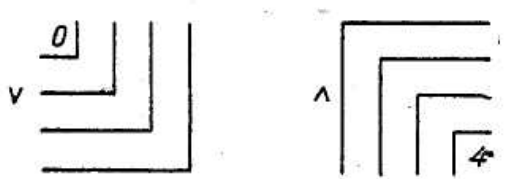


Рис. 4.

На самом деле каждая из этих операций является «отражением» другой и связь, которая позволяет одну операцию менять на другую, определяется (в $\mathbf{Z}_5$) парами (0, 4), (1, 3), (2, 2), (3, 1), (4, 0). В сущности, это принцип двойственности, который будет обсуждаться в следующих разделах. Возвращаясь к $\mathbf{Z}_2$, имеем

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

В $\mathbf{Z}_2$ операцию $\vee$ обычно интерпретируют как ***или*** (результат равен 1, если один из операндов равен 1, включая случай, когда они оба равны 1). Аналогично $\wedge$ читается как ***и***. Число 0 является единичным элементом по отношению к ***или***, число 1 является единичным элементом по отношению к ***и***. Можно распространить эти результаты

на более высокие размерности (переходя от $\mathbf{Z}_2$ к $\mathbf{Z}^n_2$), расширяя компоненты и учитывая то, что не существует переноса из одной копии $\mathbf{Z}_2$ к другой.

Пример 2.

$$\begin{array}{r} 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1 \\ \wedge \\ \underline{0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1} \\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1. \end{array}$$

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Подстановкой в формулу $a \vee b$ переменных запишите новые формулы и упростите их, если это возможно:
а) $a = \bar{x}y$, $b = z$; б) $a = xy$, $b = \bar{x}\bar{y}$;
в) $a = x$, $b = xy$; г) $a = x$, $b = \bar{x}\bar{y}$; д) $a = xy$, $b = c \vee d$, $c = xz$, $d = y\bar{z}$.

2. Запишите таблицы соответствия для следующих формул:
а) $x\bar{x}$;
б) $xy \vee \bar{x}$; в) $(p \vee q)(\bar{p} \vee \bar{q})$; г) $\overline{x \vee y}$.

3. Проверьте с помощью таблиц соответствия следующие тождества:
а) $\overline{x \vee y} = \bar{x}\bar{y}$; б) $x(x \vee y) = x$; в) $x \vee \bar{x}y = x \vee y$.

4. Постройте переключательные схемы для обеих частей приведенных ниже тождеств и убедитесь в том, что эти схемы функционируют одинаково:
а) $xy \vee \bar{x}y \vee \bar{x}\bar{y} = y \vee \bar{x}\bar{y}$;
б) $(x \vee y)(\bar{x} \vee z) = x \vee yz$;
в) $xyz \vee x\bar{y}z \vee x\bar{y} = x$.

5. Упростите следующие формулы:
а) $\bar{x}yz \vee x\bar{y}z \vee xy\bar{z}$;
б) $xy \vee z \vee \overline{xy \vee z}\,(zv \vee x)$;
в) $x\bar{y}z \vee xy\bar{z} \vee \bar{x}yz \vee xyz$;
г) $(x \vee y)(\bar{x}\bar{y} \vee z) \vee \bar{z} \vee (x \vee y)(u \vee v)$.

6. Комитет, состоящий из трех членов, принимает решения большинством голосов. Постройте такую схему, чтобы голосование каждого члена комитета производилось нажатием своей кнопки и чтобы лампочка загоралась, если и только если решение принято. Какое наименьшее количество ключей необходимо?

7. Постройте схему освещения так, чтобы лампочка могла независимо включаться и выключаться двумя выключателями

8. Преобразуйте формулы к такому виду, чтобы операция отрицания применялась только к логическим переменным;

а) $\overline{xy \vee \bar{z}}$; б) $x\,(xy \vee \overline{y\bar{z}}) \vee \overline{y \vee z\bar{v}}$.

9. Убедитесь с помощью таблиц соответствия в справедливости выражений для импликации и эквиваленции:

а) $x_1 \to x_2 = \bar{x}_1 \vee x_2$; б) $x_1 \sim x_2 =$

$= x_1x_2 \vee \bar{x}_1\bar{x}_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)$; в) $x_1 \sim x_2 = (x_1 \to x_2)(x_2 \to x_1)$.

10. Постройте переключательные схемы для импликации и эквиваленции в соответствии с тождествами, приведенными в задаче 9.

11. Запишите формулу, соответствующую переключательной схеме рис. 4. Упростите эту формулу и постройте более простую схему.

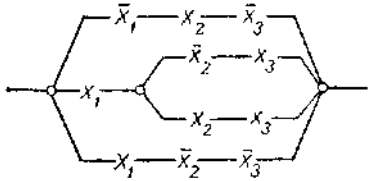


Рис. 4. Граф переключательной схемы к задаче 11.

12. Постройте переключательные схемы по формулам:

а) $(x_1 \vee x_2\bar{x}_3)(x_1x_2 \vee x_3x_4)$;

б) $(\bar{x}_1 (x_2 \vee \bar{x}_3) \vee \bar{x}_4)\, x_1$.

13. Из простых высказываний x_1 — «испытания проведены» и x_2 — «программа выполнена» образуйте сложные высказывания по формулам:

а) $x_1 \vee \bar{x}_2$; б) x_1x_2; в) $x_1 \to x_2$; г) $x_1 \sim x_2$.

14. Запишите формулы для следующих высказываний, обозначив буквами входящие в них простые высказывания:

а) Давление падает и система не работает.

б) Вычисления выполнены точно или конструкция несовершенна.

в) Проект разработал Андрей или Петр, а эксперимент выполнил Иван.

г) Если будет хорошая погода, мы отправимся на стадион или пойдем за грибами.

д) Программа может быть выполнена, если и только если материалы поступят своевременно.

е) Если я поеду на автобусе, то опоздаю на работу, или я воспользуюсь такси.

ж) Андрей помогает Петру или Петр помогает Андрею, или они помогают друг другу.

15. Запишите формулу, соответствующую высказыванию: «Программа будет выполнена тогда и только тогда, когда закончатся испытания и показатели будут удовлетворительны; если программа не будет выполнена, сотрудники не получат премию или будут пересмотрены технические условия».

16. Даны простые высказывания: x_1 — «идет дождь», x_2 — «очень жарко».

а) Запишите формулу сложного высказывания «Неверно, что идет дождь и очень жарко».

б) Преобразуйте формулу по закону де-Моргана и составьте соответствующее высказывание.

в) Убедитесь в тождественности исходного и преобразованного высказываний.

17. Путешественник остановился у развилки дорог, ведущих в пункты *A* и *B,* и ему нужно выяснить, в какой именно пункт ведет каждая из дорог. Находившиеся у развилки два человека заявили, что они могут ответить только на один вопрос и что один из них всегда правдив, а другой лжец. Какой вопрос должен задать путешественник, чтобы в любом случае ответ на него содержал необходимую информацию?

а) Решите задачу путем непосредственных рассуждений без применения алгебры логики.

б) Представьте ситуацию в виде сложного высказывания, составленного из простых.

в) Запишите соответствующую формулу и таблицу соответствия.

г) По таблице соответствия сформулируйте искомый вопрос.

18. Высказывание является логически истинным, если соответствующая ему формула тождественно равна единице, и логически ложным, если формула равна нулю. Определите с помощью таблиц соответствия, каким высказываниям соответствуют приведенные ниже формулы (истинным, ложным или ни тем и не другим): а) $p \sim p$; б) $p \to \bar{p}$; в) $(p \vee q) \sim pq$; г) $(p \to \bar{q}) \to (q \to \bar{p})$;

д) $(p \to q) \to p$; е) $((p \to q) \to p) \to p$; ж) $\overline{p \vee q} \sim pq$.

19. При $x_1 = 1$; $x_2 = 0$; $x_3 = 0$ и $x_4 = 1$ найдите значения каждой из следующих функций:

а) $\overline{x_1 \vee x_2} \vee x_3 x_4$;
б) $x_1 x_2 \vee x_3 (x_1 \vee x_4) \vee x_4 (x_2 \vee x_3)$;
в) $x_1 \to (x_2 \vee x_3)$;
г) $(x_1 \vee x_2) \sim x_2 x_3$;
д) $x_1 x_2 \to (x_2 \sim x_3)$;
е) $x_1 x_2 \to (x_3 \to x_2 x_4)$.

20. Пусть X — множество сотрудников отдела и на этом множестве определены относительно переменной $x \in X$ одноместные предикаты $P(x)$, $Q(x)$, $R(x)$, означающие соответственно: x — занимается спортом, изучает иностранный язык, имеет изобретения. Расшифруйте предикаты, образованные с помощью следующих логических операций: а) $P(x) \vee Q(x)$; б) $P(x)\,Q(x)$;
в) $\overline{P(x)}\,Q(x) \to R(x)$; г) $Q(x) \sim R(x)$; д) $\overline{R(x)} \sim (Q(x) \vee R(x))$.

21. Пусть V — множество вершин и E — множество ребер графа, причем ребро $e \in E$ соединяет вершины $x, y \in V$. Что означают предикаты $P(x, y)$, $Q(e, x, y)$, $R(x,$ е)?

22. Каким десятичным числам соответствуют следующие двоичные числа: а) 1011; б) 1000110; в) 110100111?

23. Переведите в двоичную систему счисления десятичные числа: а) 51; б) 64; в) 125; г) 1000.

24. Выполните в двоичной системе следующие операции над десятичными числами: а) 21 + 37; б) 31 + 105; в) 25 • 8; г) (8 + 19)11; д) 24 • 8 — 17. Проверьте полученные результаты в десятичной системе.

25. Переведите в двоичную систему счисления с точностью до пяти двоичных знаков после запятой числа: а) 0,131; б) 0,25; в) 175,26.

26. Дайте обоснование правил перевода десятичных чисел в двоичные.

27. Сложите двоичные числа 11001110 и 11010111 по обычному правилу и по модулю два. Найдите разность полученных результатов и объясните ее смысл.

2. Логические функции

2.1. Логические функции как отображения

Логи́ческая фу́нкция (или **Бу́лева функция**, или **функция а́лгебры ло́гики**) от n аргументов — в дискретной математике — отображение $B^n \to B$, где $B = \{0,1\}$ — *булево множество*. Элементы булева множества {1, 0} обычно интерпретируют как логические значения

«истинно» и «ложно», хотя в общем случае они рассматриваются как формальные символы, не несущие определённого смысла.
Неотрицательное целое число n называют арностью или местностью функции, в случае $n = 0$ булева функция превращается в *булеву константу*. Элементы декартова произведения (n-я прямая степень) B^n называют *булевыми векторами*. Множество всех булевых функций от любого числа аргументов часто обозначается P_2, а от n аргументов — $P_2(n)$. Переменные, принимающие значения из булева множества называются *булевыми переменными*. Булевы функции названы по фамилии математика Джорджа Буля.

При работе с булевыми функциями происходит полное абстрагирование от содержательного смысла, который имелся в виду в алгебре высказываний. Тем не менее, между булевыми функциями и формулами алгебры высказываний можно установить взаимно-однозначное соответствие, если:

- установить взаимно-однозначное соответствие между булевыми переменными и пропозициональными переменными,
- установить связь между булевыми функциями и логическими связками,
- оставить расстановку скобок без изменений.

Каждая булева функция арности n полностью определяется заданием своих значений на своей области определения, то есть на всех булевых векторах длины n. Число таких векторов равно 2^n. Поскольку на каждом векторе булева функция может принимать значение либо 0, либо 1, то количество всех n-арных булевых функций равно $2^{(2n)}$. Поэтому в этом разделе рассматриваются только простейшие и важнейшие булевы функции.
Практически все булевы функции малых арностей (0, 1, 2 и 3) сложились исторически и имеют конкретные имена. Если значение функции не зависит от одной из переменных (то есть строго говоря для любых двух булевых векторов, отличающихся лишь в значении этой переменной, значение функции на них совпадает), то эта переменная называется *фиктивной*.

2.2. Однородные функции

Логические функции могут зависеть от одной, двух и, вообще, любого числа переменных (аргументов) $x_1, x_2, ..., x_n$. В отличие от самой функции, аргументы могут принимать значения из элементов как конечных, так и бесконечных множеств.

В теоретико-множественном смысле логическая функция n переменных $y = f(x_1, x_2, ..., x_n)$ представляет собой отображение множества *наборов* (n-мерных векторов, кортежей, последовательностей) вида $(x_1, x_2, ..., x_n)$, являющегося областью ее определения, на множество ее значений $N = \{\alpha_1, \alpha_2, ..., \alpha_n\}$. Логическую функцию можно также рассматривать как операцию, заданную законом композиции $X_1 \times X_2 \times ... \times X_n \to N$, где. $X_1, X_2, ..., X_n$ — множества, на которых определены аргументы $x_1 \in X_1, x_2 \in X_2, ..., x_n \in X_n$.

Если аргументы принимают значения из того же множества, что и сама функция, то ее называют *однородной функцией*. В этом случае $X_1 = X_2 = ... = X_n = N$ и однородная функция, рассматриваемая как закон композиции $N'' \to N$, определяет некоторую *n-местную операцию* на конечном множестве N.

Областью определения однородной функции $y = f(x_1, x_2, ..., x_n)$ служит множество наборов $(x_1, x_2, ..., x_n)$, называемых *словами,* где каждый из аргументов $x_1, x_2, ..., x_n$ замещается буквами k-ичного алфавита $\{0, 1, ..., k — 1\}$. Количество n букв в данном слове определяет его *длину*.

Очевидно, число всевозможных слов длины n в k-ичном алфавите равной k^n. Так как каждому такому слову имеется возможность предписать одно из k значений множества N, то общее количество однородных функций от n переменных выражается числом $k^{(kn)}$.

Если буквами алфавита служат числа от 0 до k — 1, то каждое слово $(x_1, x_2, ..., x_n)$ символически представляется упорядоченной последовательностью n таких чисел и рассматривается как запись n-разрядного числа в позиционной системе счисления с основанием k, т. е. $x_1k^{n-1} + x_2k^{n-2} + ... + x_{n-1}k^1 + x_nk^0 = q$. Числа $q = 0, 1, ..., k^n — 1$ служат *номерами слов* и тем самым на множестве всех слов вводится естественная упорядоченность (отношение строгого порядка).

Аналогично *номерами функций* можно считать k^n-разрядные числа в той же системе счисления.

Различные слова длины n в данном алфавите образуются как n-перестановки с повторениями. Так, в трехзначном алфавите $\{0, 1, 2\}$ словами длины 4 будут все четырехразрядные числа с основанием $k = 3$, т. е. 0000, 0001, 0002, 0010, 0011, ..., 2221, 2222, которые

соответствуют десятичным числам от 0 до $80 = 2 \cdot 3^3 + 2 \cdot 3^2 + 2 \cdot 3^1 + 2 \cdot 3^0$.

Поставив каждому такому четырехразрядному числу в соответствие одну из букв алфавита $\{0, 1, 2\}$, получим некоторую функцию четырех переменных $f_i(x_1, x_2, x_3, x_4)$, причем количество таких функций выражается огромным числом 3^{81}.

Пусть алфавит состоит из трех букв русского алфавита {о, п, т}. Множество пятибуквенных слов в этом алфавите состоит из $3^6 = 243$ элементов. Наряду с такими имеющими прямой смысл словами, как «топот» и «потоп», оно также включает все другие 5-перестановки, например: «ооппт», «поппп», «тттоп» и др.

Примерами однородных логических функций двух переменных могут служить операции сложения и умножения одноразрядных m-значных чисел по модулю m, внутренние операции поля Галуа с четырехзначным алфавитом {0, 1, *A*, B} и т. п.

2.3. Табличное задание функций.

Как и бинарный закон композиции , однородная функция двух переменных может быть задана таблицей соответствия (матрицей), строки и столбцы которой соответствуют буквам алфавита. Таким способом представлялись ранее функции одной и двух переменных. Для представления функций трех и большего числа переменных потребовались бы трехмерные и, вообще, n-мерные таблицы. Этого можно избежать, если столбцы матрицы поставить в соответствие не буквам алфавита, а словам, т. е. образовать k^n столбцов. Для каждой функции отводится строка, клетки которой заполняются буквами из данного алфавита. Матрица всех функций n переменных в k-значном алфавите содержит $k^{(k^n)}$ строк и называется *общей таблицей соответствия*. Например, для $k = 3$ и $n = 2$ такая матрица имеет вид:

x_1	0	0	0	1	1	1	2	2	2
x_2	0	1	2	0	1	2	0	1	2
y_0	0	0	0	0	0	0	0	0	0
y_1	0	0	0	0	0	0	0	0	1
y_2	0	0	0	0	0	0	0	0	2
...	.	.	.	.	.	.	.	.	.
y_{2361}	0	1	0	0	1	2	2	0	1
...	.	.	.	.	.	.	.	.	.
y_{19682}	2	2	2	2	2	2	2	2	2

Номера столбцов определяются расположенными над ними

n-разрядными числами с основанием *k*, каждое из которых читается сверху вниз. Номера функций отождествляются с k^n-разрядными числами, которые соответствуют строкам матрицы в той же системе счисления.

Двузначные однородные функции. Наиболее простым и в то же время важнейшим классом однородных функций являются *двузначные (булевы) функции,* частично рассмотренные ранее. Областью определения булевых функций от *n* переменных служит множество слов длины *n.* Они представляют собой всевозможные наборы из *n* двоичных цифр и их общее количество равно 2^n.

Число всевозможных булевых функций *n* переменных $v = 2^n$ быстро возрастает с увеличением *n* (при *n* = 3 оно равно 256, а при *n* = 5 превышает 4 миллиарда). Но функции одной и двух переменных еще можно перечислить и подробно исследовать, так как их количество сравнительно невелико (v = 4 при *n* = 1 и v = 16 при *n* = 2).

Булевы функции одной переменной. Общая таблица соответствия для булевых функций одной переменной имеет вид (справа указаны обозначения функций):

x	0	1	y
y_0	0	0	0
y_1	0	1	x
y_2	1	0	$\bar{x}$
y_3	1	1	1

Две функции $y_0 = 0$ и $y_3 = 1$ представляют собой *функции-константы* (тождественный нуль и тождественная единица), так как они не изменяют своих значений при изменении аргумента. Функция $y_1 = x$ повторяет значения переменной *x* и потому просто совпадает с ней. Единственной нетривиальной функцией является $y_2 = \bar{x}$, называемая *отрицанием* или *инверсией* ($\bar{x}$ читается «не *x*»). Она равна 1, когда аргумент принимает значение 0, и равна 0 при аргументе 1.

Булевы функции двух переменных. Все 16 функций двух переменных приведены в табл. 1, где указаны условные обозначения, названия и чтения функций (в скобках даны встречающиеся в литературе варианты).

Шесть из приведенных функций не зависят от x_1 или x_2 (или от обоих вместе). Это две константы ($y_0 = 0$ и $y_{15} = 1$), повторения ($y_3 = x_1$ и $y_5 = x_2$) и отрицания ($y_{10} = \bar{x}_2$, $y_{12} = \bar{x}_1$), являющиеся функциями одной переменной (x_1 или x_2). Из остальных десяти функций две (y_4 и y_{11}) отличаются от соответствующих им (y_2 и y_{13}) лишь порядком расположения аргументов и поэтому не являются самостоятельными.

Поэтому из 16 булевых функций двух переменных только восемь являются оригинальными ($y_1, y_2, y_6, y_7, y_8, y_9, y_{13}, y_{14}$).

Рассмотрение булевых функций одной, двух и большего числа переменных показывает, что всякая функция от меньшего числа переменных содержится среди функций большего числа переменных.

Таблица 1

Булевы функции двух переменных

x_1 / x_2	0 0 1 1 / 0 1 0 1	Обозначения	Названия	Чтение
y_0	0 0 0 0	0	Константа 0 (тождественный нуль, всегда ложно)	Любое 0
y_1	0 0 0 1	x_1x_2; $x_1 \wedge x_2$ ($x_1 \,\&\, x_2$; $x_1 \cap x_2$)	Конъюнкция (совпадение, произведение, пересечение, логическое «и»)	x_1 и x_2 (x_1 и x_2)
y_2	0 0 1 0	$x_1 \not\leftarrow x_2$ ($x_1 \not\supset x_2$; $x_1 \setminus x_2$)	Отрицание импликации (совпадение с запретом, антисовпадение, запрет)	x_1, но не x_2
y_3	0 0 1 1	x_1	Повторение (утверждение, доминация) первого аргумента	Как x_1
y_4	0 1 0 0	$x_2 \not\leftarrow x_1$ ($x_1 \not\subset x_2$; $x_2 \setminus x_1$)	Отрицание обратной импликации (обратное антисовпадение)	Не x_1, но x_2
y_5	0 1 0 1	x_2	Повторение (утверждение, доминация) второго аргумента	Как x_2
y_6	0 1 1 0	$x_1 + x_2$ ($x_1 \nabla x_2$; $x_1 \oplus x_2$)	Сумма по модулю 2 (неравнозначность, антиэквивалентность)	x_1 не как x_2 (или x_1 или x_2)
y_7	0 1 1 1	$x_1 \vee x_2$ ($x_1 + x_2$; $x_1 \cup x_2$)	Дизъюнкция (разделение, логическая сумма, сборка, логическое «или»)	x_1 или x_2 (x_1 или хотя бы x_2)
y_8	1 0 0 0	$x_1 \downarrow x_2$ ($x_1 \overline{\vee} x_2$; $x_1 \circ x_2$)	Стрелка Пирса (функция Вебба, отрицание дизъюнкции, логическое «не — или»)	Ни x_1, ни x_2

Продолжение табл. 1

x_1 / x_2	0 0 1 1 / 0 1 0 1	Обозначения	Названия	Чтение
y_9	1 0 0 1	$x_1 \sim x_2$ ($x_1 \equiv x_2$; $x_1 \leftrightarrow x_2$)	Эквиваленция (равнозначность, эквивалентность, взаимозависимость)	x_1 как x_2 (x_1, если и только если x_2)
y_{10}	1 0 1 0	$\bar{x}_2$ ($\bar{x}_2$; $\sim x_2$; $\neg x_2$)	Отрицание (инверсия) второго аргумента (дополнение к первой переменной)	Не x_2
y_{11}	1 0 1 1	$x_2 \rightarrow x_1$ ($x_1 \subset x_2$; $x_1 < x_2$)	Обратная импликация (обратное разделение с запретом, обратная селекция)	Если x_2, то x_1 (x_1 или не x_2)
y_{12}	1 1 0 0	$\bar{x}_1$ ($\bar{x}_1$; $\sim x_1$; $\neg x_1$)	Отрицание (инверсия) первого аргумента (дополнение ко второй переменной)	Не x_1
y_{13}	1 1 0 1	$x_1 \rightarrow x_2$ ($x_1 \supset x_2$; $x_1 > x_2$)	Импликация (разделение с запретом, следование, селекция)	Если x_1, то x_2 (не x_1 или x_2)
y_{14}	1 1 1 0	$\overline{x_1/x_2}$ ($\overline{x_1 \wedge x_2}$; $\overline{x_1 \,\&\, x_2}$)	Штрих Шеффера (отрицание конъюнкции, несовместность, логическое «не—и»)	Не x_1 или не x_2
y_{15}	1 1 1 1	1	Константа 1 (тождественная единица, всегда истинно)	Любое 1

Функции, которые сводятся к зависимости от меньшего числа переменных, называют *вырожденными,* а функции, существенно зависящие от всех переменных, являются *невырожденными.* Так, среди функций одной переменной имеются две вырожденные (константы 0 и 1, которые можно рассматривать как функции от нуля переменных), функции двух переменных содержат те же константы и четыре функции одной переменной и т. д.

Зависимость между булевыми функциями. Из табл. 1 видно, что между функциями имеются зависимости $y_i = \bar{y}_{15-i}$ ($i = 0, 1, \ldots, 15$), на основании которых можно записать соотношения для констант $0 = \bar{1}$ и $1 = \bar{0}$, для функции одной переменной $x = \bar{\bar{x}}$

и для функций двух переменных:

$$x_1x_2 = \overline{x_1/x_2};\ x_1 \leftarrow x_2 = \overline{x_1 \to x_2};\ x_1 + x_2 = \overline{x_1 \sim x_2};\ x_1 \vee x_2 = \overline{x_1 \downarrow x_2},$$

или

$$x_1/x_2 = \overline{x_1x_2};\ x_1 \to x_2 = \overline{x_1 \leftarrow x_2};\ x_1 \sim x_2 = \overline{x_1 + x_2};\ x_1 \downarrow x_2 = \overline{x_1 \vee x_2}.$$

Из этих зависимостей следует, что любая функция двух переменных (включая константы) выражается в аналитической форме через совокупность шести функций, содержащей отрицание $\bar{x}$ и любую из каждой пары функций $\{y_0, y_{15}\}$, $\{y_1, y_{14}\}$, $\{y_2, y_{13}\}$, $\{y_6, y_9\}$, $\{y_7, y_8\}$. Например, такой совокупностью могут служить функции: константа 0, отрицание $\bar{x}$, конъюнкция x_1x_2, дизъюнкция $x_1 \vee x_2$, эквиваленция $x_1 \sim x_2$ и импликация $x_1 \to x_2$. Как уже упоминалось, они используются в исчислении высказываний.

Выбранная таким способом совокупность шести функций является избыточной. Можно показать, что импликация и эквиваленция выражаются через остальные функции этой совокупности:

$$x_1 \to x_2 = \bar{x}_1 \vee x_2;$$
$$x_1 \sim x_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2).$$

Для этого достаточно построить таблицу соответствия и сравнить ее с табл. 1:

x_1	0	0	1	1	
x_2	0	1	0	1	
$\bar{x}_1$	1	1	0	0	
$\bar{x}_2$	1	0	1	0	
$\bar{x}_1 \vee x_2$	1	1	0	1	$x_1 \to x_2$
$x_1 \vee \bar{x}_2$	1	0	1	1	
$(x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)$	1	0	0	1	$x_1 \sim x_2$

Таким образом, комплект элементарных функций сокращается до четырех: константа 0, отрицание $\bar{x}$, конъюнкция x_1x_2 и дизъюнкция $x_1 \vee x_2$. Этот комплект обладает существенными удобствами и часто применяется на практике, но и он может быть сокращен. Так, из законов де Моргана и свойства двойного отрицания вытекают тождества:

$$x_1 \vee x_2 = \overline{\bar{x}_1\bar{x}_2};\quad x_1x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}.$$

Отсюда следует, что булевы функции выражаются через отрицание и конъюнкцию или через отрицание и дизъюнкцию.

Более того, для записи любой булевой функции достаточно только одной из двух элементарных функций — стрелки Пирса или штриха Шеффера. Это вытекает из соотношений (их доказательство приводится аналогично с помощью таблиц соответствия):

$$\bar{x} = x \downarrow x = x/x;$$
$$x_1x_2 = (x_1/x_2)/(x_1/x_2);\ x_1 \vee x_2 = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2).$$

Булевы функции многих переменных. С помощью суперпозиции функций, т. е. подстановки в логические формулы вместо переменных некоторых булевых функций, можно получить более сложные функции от любого числа переменных. Например, подставляя в выражение ab формулы $a = x_1 \vee x_2$ и $b = x_2 \to c$, а также $c = \bar{x}_3$, получаем $(x_1 \vee x_2)(x_2 \to \bar{x}_3)$. Таблица соответствия для сложных формул записывается на основании общей таблицы для элементарных функций. Для данного примера она имеет вид:

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$x_1 \vee x_2$	0	0	1	1	1	1	1	1
$\bar{x}_3$	1	0	1	0	1	0	1	0
$x_2 \to \bar{x}_3$	1	1	1	0	1	1	1	0
$(x_1 \vee x_2)(x_2 \to \bar{x}_3)$	0	0	1	0	1	1	1	0

Если на всех наборах значений переменных функция принимает значение 0 или 1, то она вырождается в соответствующую константу и называется *тождественным нулем* или *тождественной единицей.* Например, $x \vee \bar{x} = 1$; $x\bar{x} = 0$; $x\bar{x} \vee x\bar{x}y = 0$; $((xy \vee \bar{y}z) \to \bar{z}) \vee \vee (x \vee \bar{y})\, z = 1$; $x(x \to y) \to y = 1$ и т. п.

Геометрическое представление. Область определения булевых функций от n переменных $y = f(x_1, x_2, \ldots, x_n)$ можно рассматривать как совокупность n-мерных векторов (слов длины n). компонентами которых являются буквы 0 и 1 двоичного алфавита. При $n = 3$ каждый вектор представляется вершиной единичного куба в трехмерном пространстве (рис. 1).

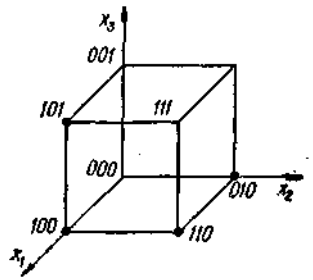


Рис.1. Отображение булевой функции $y = (x_1 \vee x_2) \times (x_2 \to \bar{x}_3)$ на трехмерном кубе.

В общем случае совокупность векторов $(x_1x_2, \ldots x_n)$ отображается на множество вершин n-мерного куба. Все такие вершины образуют *логическое пространство.*
Булева функция отображается на nмерном кубе путем выделения вершин, соответствующих векторам $(x_1x_2, \ldots x_n)$, на которых булева функция $y = f(x_1, x_2, \ldots, x_n)$ принимает значения 1. Обычно такие вершины отмечают жирными точками. Так, на рис. 1 отображена функция $(x_1 \vee x_2)(x_2 \to \bar{x}_3)$ в соответствии с выше приведенной таблицей.

2.4. Неоднородные функции.

Аргументы *неоднородных функций,* в отличие от однородных, могут принимать значения из любых конечных или бесконечных множеств, но область значений самих функций ограничена конечными множествами.

Важным примером неоднородных функций являются двузначные *n-местные предикаты*. Предикат $P(x_1, x_2, \ldots, x_n)$ принимает одно из двух значений — «истинно» (1) или «ложно» (0) в зависимости от конкретных значений, приписываемых переменным $x_1, x_2, \ldots, x_n$. Если значения переменных выбираются из некоторого множества M (универсума), то n-местный предикат можно рассматривать как n-местное отношение, определенное на этом множестве.
Одноместный предикат $P(x)$ задает некоторое свойство элементов множества M и вполне определяется подмножеством $P \subset M$ тех объектов $x \in M$, на которых он принимает значение «истинно».

Множество объектов, на которых предикат $P(x)$ принимает значение «ложно», соответствует дополнению множества P, т. е. $\bar{P}$. Очевидно, если $P(x)$ истинно, то $\bar{P}(x)$— ложно и наоборот. Например, если на множестве натуральных чисел определен предикат $P(x) =$ $=$ «x — четное число», то $\bar{P}(x) =$ «x — нечетное число». Таким образом, одноместный предикат, определенный на множестве M, разбивает это множество на два подмножества P и $\bar{P}$. Подмножество $P \subset M$, на котором предикат $P(x)$ принимает значение «истинно», называется *характеристическим подмножеством.*
Пусть на M определены два предиката P(x) п Q(x), характеистическими подмножествами которых являются соответственно P и Q.
Рассматривая предикаты как двузначные функции, можно с помощью операций алгебры логики строить новые одноместные предикаты на множестве M. *Конъюнкция* $P(x)$ и $Q(x)$ — это предикат $R(x) = P(x) \wedge Q(x)$, который истинен для тех и только тех объектов из M, для которых оба предиката $P(x)$ и Q(x) истинны.
Характеристическим множеством предиката $R(x)$ является пересечение $P \cap Q$. Подобным образом вводятся и операции дизъюнкции $P(x) \vee Q(x)$, импликации $P(x) \to Q(x)$, эквиваленции $P(x) \sim Q(x)$ и др. На рис. 2 показаны соответствующие этим операциям характеристические подмножества (область истинных значений заштрихована).

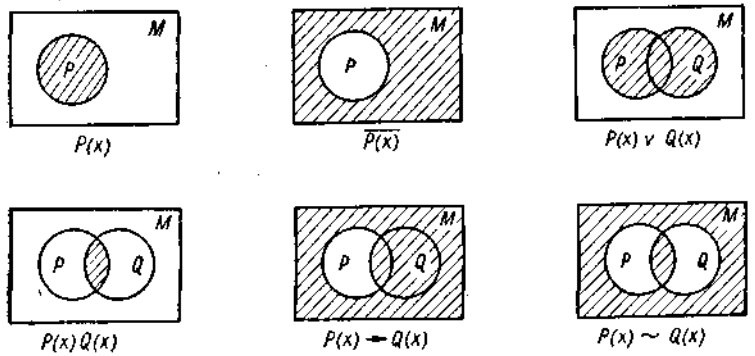


Рис. 2. Характеристические подмножества, соответствующие операциям над предикатами (область истинных значений заштрихована).

Их легко получить из таблиц соответствия для функций двух переменных. Имеют место также соответствия между различными операциями, вытекающие из зависимостей между булевыми функциями: $P(x) \to Q(x)$ соответствует $\overline{P}(x) \vee Q(x)$, $P(x) \sim Q(x)$ соответствует $(P(x) \vee \overline{Q(x)})(\overline{P(x)} \wedge Q(x))$ или $P(x)Q(x) \vee \overline{P(x)}\,\overline{Q(x)}$ и т. п.

2.5. Таблицы истинности

Таблица истинности — это таблица, описывающая логическую функцию.

Под «логической функцией» в данном случае понимается функция, у которой значения переменных (параметров функции) и значение самой функции выражают логическую истинность. Например, в двузначной логике они могут принимать значения «истина» либо «ложь» (true либо false, 1 либо 0).

Табличное задание функций встречается не только в логике, но для логических функций таблицы оказались особенно удобными, и с начала XX века за ними закрепилось это специальное название. Особенно часто таблицы истинности применяются в булевой алгебре и в аналогичных системах многозначной логики.

Булева функция задаётся конечным набором значений, что позволяет представить её в виде таблицы истинности, например:

x_1	x_2	…	x_{n-1}	x_n	$f(x_1,x_2,\ldots,x_n)$
0	0	…	0	0	0
0	0	…	0	1	0
0	0	…	1	0	1
0	0	…	1	1	0
1	1	…	0	0	1

1	1	…	0	1	0
1	1	…	1	0	0
1	1	…	1	1	0

Таблицы истинности для основных двоичных логических функций приведены в табл.1

Таблицы истинности для некоторых троичных логических функций

x	2 1 0 2 1 0 2 1 0
y	2 2 2 1 1 1 0 0 0
min(x,y)	2 1 0 1 1 0 0 0 0
x	2 1 0 2 1 0 2 1 0
y	2 2 2 1 1 1 0 0 0
max(x,y)	2 2 2 2 1 1 2 1 0
x	2 1 0 2 1 0 2 1 0
y	2 2 2 1 1 1 0 0 0
F2TN22310	0 0 0 0 2 2 0 2 1

Нульарные функции

При $n = 0$ количество булевых функций сводится к двум $2^{2^0} = 2^1 = 2$, первая из них тождественно равна 0, а вторая 1. Их называют булевыми константами — тождественный нуль и тождественная единица.
Таблица значений и названий нульарных булевых функций:

Значение	Обозначение	Название
0	F0,0 = 0	тождественный ноль
1	F0,1 = 1	тождественная единица, тавтология

Унарные функции

При $n = 1$ число булевых функций равно $2^{2^1} = 2^2 = 4$. Определение этих функций содержится в следующей таблице.

Таблица значений и названий булевых функций от одной переменной:

$x_0=x$	1	0	Обозначение	Название
0	0	0	F1,0 = 0	тождественный ноль
1	0	1	F1,1 = $x = \neg x = x' =$ NOT(x)	отрицание, логическое «НЕТ», «НЕ», «НИ», инвертор, SWAP (обмен)
2	1	0	F1,2 = x	тождественная функция, логическое "ДА", повторитель
3	1	1	F1,3 = 1	тождественная единица, тавтология

Бинарные функции

При $n = 2$ число булевых функций равно $2^{2^2} = 2^4 = 16$.

Таблица значений и названий булевых функций от двух переменных:

$x_0=x$	1	0	1	0		
$x_1=y$	1	1	0	0	Обозначение	Название
0	0	0	0	0	F2,0 = 0	тождественный ноль
1	0	0	0	1	F2,1 = $x \downarrow y = x$ NOR $y =$ NOR(x,y) = x НЕ-ИЛИ $y =$ НЕ-ИЛИ(x,y) = NOT(MAX(X,Y))	стрелка Пи́рса - "↓" (кинжал Куайна - "†"), функция Ве́бба - "○"[5], НЕ-ИЛИ, 2ИЛИ-НЕ, антидизъюнкция, инверсия максимума
2	0	0	1	0	F2,2 = $x \rightarrow y = x > y = x$ GT y = GT(x,y)	инверсия прямой импликации, больше
3	0	0	1	1	F2,3 = $y = y' = \neg y =$ NOT2(x,y) = НЕ2(x,y)	отрицание (негация, инверсия) второго операнда
4	0	1	0	0	F2,4 = $x \leftarrow y = x < y = x$ LT y = LT(x,y)	инверсия обратной импликации, меньше
5	0	1	0	1	F2,5 = $x = x' = \neg x =$ NOT1(x,y) = НЕ1(x,y)	отрицание (негация, инверсия) первого операнда
6	0	1	1	0	F2,6 = x □ $y = x$ XOR $y =$	сложение по модулю 2,

					XOR(x,y) = x >< $y = x$ <> $y = x$ NE y = NE(x, y)	исключающее «или», сумма Жегалкина[6], не равно
7	0	1	1	1	F2,7 = $x \mid y = x$ NAND $y =$ NAND(x,y) = x НЕ-И $y =$ НЕ-И(x,y) = NOT(MIN(X,Y))	штрих Ше́ффера, НЕ-И, 2И-НЕ, антиконъюнкция, инверсия минимума
8	1	0	0	0	F2,8 = $x \wedge y = x \cdot y = xy = x$ & $y = x$ AND $y =$ AND(x,y) = x И y = И(x,y) = min(x,y)	конъюнкция, 2И, минимум
9	1	0	0	1	F2,9 = $(x \equiv y) = x \sim y = x \leftrightarrow y = x$ EQV y = EQV(x,y)	эквивалентность, равенство
10	1	0	1	0	F2,10 = YES1(x,y) = ДА1(x,y) = x	первый операнд
11	1	0	1	1	F2,11 = $x \leftarrow y = x \subset y = x \geq y = x$ GE y = GE(x,y)	обратная импликация (от второго аргумента к первому), больше или равно
12	1	1	0	0	F2,12 = YES2(x,y) = ДА2(x,y) = y	второй операнд
13	1	1	0	1	F2,13 = $x \rightarrow y = x \supset y = x \leq y = x$ LE y = LE(x,y)	прямая (материальная) импликация (от первого аргумента ко второму), меньше или равно
14	1	1	1	0	F2,14 = $x \vee y = x + y = x$ OR y = OR(x,y) = x ИЛИ y = ИЛИ(x,y) = max(x,y)	дизъюнкция, 2ИЛИ, максимум
15	1	1	1	1	F2,15 = 1	тождественная единица, тавтология

Аналогичная таблица в английской Википедии. (Из-за отсутствия правила очерёдности перебора значений функций в таблице англоязычного автора функция F2 получила номер и колонку 4, функция F4 получила номер и колонку 2, функция F11 получила номер и колонку 13, а функция F13 получила номер и колонку 11. У нас же перебор начинают строго с младшего по порядку (нулевого или первого) аргумента.)
При двух аргументах префиксная, инфиксная и постфиксная записи, по экономичности, почти одинаковы.

Тернарные функции

При $n = 3$ число булевых функций равно $2^{(23)} = 2^8 = 256$. Некоторые из них определены в следующей таблице:
Таблица значений и названий некоторых булевых функций от трех переменных, имеющих собственное название:

x_0=x	1	0	1	0	1	0	1	0		
x_1=y	1	1	0	0	1	1	0	0		
x_2=z	1	1	1	1	0	0	0	0	**Обозначения**	**Названия**
1	0	0	0	0	0	0	0	1	F3,1 = $x\downarrow y\downarrow z$ = ↓(x,y,z) = $Webb_2$(x,y,z) = NOR(x,y,z)	3ИЛИ-НЕ, функция Вебба, функция Даггера, стрелка Пирса
23	0	0	0	1	0	1	1	1	F3,23 = = ≥2(x,y,z)	Переключатель по большинству с инверсией, 3ППБ-НЕ, мажоритарный клапан с инверсией
126	0	1	1	1	1	1	1	0	F3,126 = (x≠y≠z) = [≠(x,y,z)] = NE(x,y,z)	Неравенство
127	0	1	1	1	1	1	1	1	F3,127 = x|y|z = |(x,y,z) = NAND(x,y,z)	3И-НЕ, штрих Шеффера
128	1	0	0	0	0	0	0	0	F3,128 = x&y&z = &(x,y,z) = (x AND y AND z) = AND(x,y,z) = (x И y И z) = И(x,y,z) = min(x,y,z)	3И, минимум
129	1	0	0	0	0	0	0	1	F3,129 = (x=y=z) = [=(x,y,z)] = EQV(x,y,z)	Равенство
150	1	0	0	1	0	1	1	0	F3,150 = x□y□z = x□$_2$y□$_2$z = □$_2$(x,y,z)	Тернарное сложение по модулю 2
216	1	1	0	1	1	0	0	0	F3,216 = f_1	Разряд займа при тернарном вычитании
232	1	1	1	0	1	0	0	0	F3,232 = f_2 = [>=2(x,y,z)] = ≥2(x,y,z) = (x И y) ИЛИ (y И z) ИЛИ (z И x)	Разряд переноса при тернарном сложении, переключатель по большинству, 3ППБ, мажоритарный клапан
254	1	1	1	1	1	1	1	0	F3,254 = (x+y+z) =	ИЛИ, максимум

+(x,y,z) = (x OR y OR z) = OR(x,y,z) = (x ИЛИ y ИЛИ z) = ИЛИ(x,y,z) = max(x,y,z)

При трёх и более аргументах префиксная (и постфиксная) запись экономичнее инфиксной записи.
Обычный вид записи функций — префиксный (перед операндами). При инфиксной (между операндами) записи функций функции называются операторами, а аргументы функции — операндами.

2.6. Полные системы булевых функций

2.6.1.Суперпозиция и замкнутые классы функций

Результат вычисления булевой функции может быть использован в качестве одного из аргументов другой функции. Результат такой операции суперпозиции можно рассматривать как новую булеву функцию со своей таблицей истинности. Например, функции $f(x, y, z) = \overline{x(\overline{y} \vee z)}$ (суперпозиция конъюнкции, дизъюнкции и двух отрицаний) будет соответствовать следующая таблица:

$x_2 = x$	$x_1 = y$	$x_0 = z$	$f(x, y, z)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Говорят, что множество функций замкнуто относительно операции суперпозиции, если любая суперпозиция функций из данного

множества тоже входит в это же множество. Замкнутые множества функций называют также замкнутыми классами.

Замкнутый класс в теории булевых функций — такое множество P функций алгебры логики, замыкание которого относительно операции суперпозиции совпадает с ним самим: [P]=P. Другими словами, любая функция, которую можно выразить формулой с использованием функций множества P, снова входит в это же множество.

В 1941 году Эмиль Пост представил полное описание системы замкнутых классов, называемое также решёткой Поста.

Эмиль Пост показал, что любой замкнутый класс булевых функций является пересечением конечного числа описанных выше классов, приведя полное описание структуры замкнутых классов двузначной логики. Также Пост установил, что любой замкнутый класс может быть порожден конечным базисом. В качестве простейших примеров замкнутых классов булевых функций можно назвать множество $\{x\}$, состоящее из одной тождественной функции, или множество $\{0\}$, все функции из которого тождественно равны нулю вне зависимости от своих аргументов. Замкнуты также множество функций $\{x, \overline{x}\}$ и множество всех унарных функций. А вот объединение замкнутых классов может таковым уже не являться. Например, объединив классы $\{0\}$ и $\{x, \overline{x}\}$, мы с помощью суперпозиции $\overline{0}$ сможем получить константу 1, которая в исходных классах отсутствовала. Разумеется, множество P_2 всех возможных булевых функций тоже является замкнутым.

Свойства замыкания функции с переменными

1. Любое множество является подмножеством своего замыкания: $A \subseteq [A]$.
2. Замыкание подмножества является подмножеством замыкания: $A \subseteq B \Rightarrow [A] \subseteq [B]$.
   Следует заметить, что из строгого вложения множеств следует лишь нестрогое вложение их замыканий: : $A \subset B \Rightarrow [A] \subseteq [B]$.
3. Многократное применение операции замыкания эквивалентно однократному: [[A]]=[A].

Примеры замкнутых классов

Множество P_2 всех возможных булевых функций замкнуто.

Особо важны для теории булевых функций следующие замкнутые классы, называемые предполными классами:

- Класс T_0 функций, сохраняющих константу 0: $T_0=\{ f(x_1,\dots,x_n) \mid f(0,\dots,0)=0\}$.
- Класс T_1 функций, сохраняющих константу 1: $T_1=\{ f(x_1,\dots,x_n) \mid f(1,\dots,1)=1\}$.
- Класс S самодвойственных функций: $S=\{ f(x_1,\dots,x_n) \mid f(\overline{x}_1,\dots,\overline{x}_n) = \overline{f(x_1,\dots,x_n)}\}$.
- Класс M монотонных булевых функций: $M=\{ f(x_1,\dots,x_n) \mid \forall i(a_i \le b_i) \Rightarrow f(a_1,\dots,a_n) \le f(b_1,\dots,b_n)\}$.
- Класс L линейных булевых функций: $L=\{ f(x_1,\dots,x_n) \mid f(x_1,\dots,x_n) = a_0 \oplus a_1x_1 \oplus \dots \oplus a_nx_n,\ a_i\{0, 1\}\}$.

Любой замкнутый класс булевых функций, отличный от P_2, целиком содержится хотя бы в одном из пяти предполных классов.

Другими важными замкнутыми классами являются:

- Класс конъюнкций K, являющийся замыканием множества $\{\wedge, 0, 1\}$. Он представляет собой множество функций вида

  $c_0 \wedge (c_1 \vee x_1) \wedge \dots \wedge (c_n \wedge x_n)$.

- Класс дизъюнкций D, являющийся замыканием множества $\{\vee, 0, 1\}$. Он представляет собой множество функций вида

  $c_0 \vee (c_1 \wedge x_1) \vee \dots \vee (c_n \wedge x_n)$.

- Класс функций одной переменной U, содержащий только константы, отрицание и селектор (функцию, равную одному из своих аргументов на всех наборах их значений).

- Класс O^m *функций* (m — любое натуральное, большее единицы число), удовлетворяющих следующему условию: для любых m наборов, на которых функция принимает нулевое значение, найдется переменная, также принимающая нулевое значение на всех этих наборах.
- Класс O^∞ функций, для которых выполнено условие $f(x_1,\ldots,x_n) \geq x_i$, где x_i — одна из переменных функции.
- Класс I^m *функций* (m — любое натуральное, большее единицы число), удовлетворяющих следующему условию: для любых m наборов, на которых функция принимает единичное значение, найдется переменная, также принимающая единичное значение на всех этих наборах.
- Класс I^∞ функций, для которых выполнено условие , $f(x_1,\ldots,x_n) \leq x_i$, где x_i — одна из переменных функции.

В 1941 году Эмиль Пост показал, что любой замкнутый класс булевых функций является пересечением конечного числа описанных выше классов, приведя полное описание структуры замкнутых классов двузначной логики. Также Пост установил, что любой замкнутый класс может быть порожден конечным базисом.

Некоторые свойства замкнутых классов

- Непустое пересечение замкнутых классов снова является замкнутым классом.
- Объединение замкнутых классов может замкнутым классом не являться.
- Замкнутый класс булевых функций, содержащий не только константы, обязательно содержит тождественную функцию.
- Дополнение замкнутого класса булевых функций до множества всех булевых функций P_2 замкнутым классом не является.

Полные системы функций

Множество А функций алгебры логики называется **полной системой**, если замыкание этого множества совпадает с множеством всех функций. (В частности, для двузначной логики $[A]=P_2$.) Другими

словами, должна быть возможность любую функцию алгебры логики выразить формулой с использованием функций множества А.

Критерий Поста формулирует необходимое и достаточное условие полноты системы булевых функций:
Система булевых функций полна тогда и только тогда, когда она не содержится целиком ни в одном из классов T_0, T_1, S, M, L частности, если функция не входит ни в один из классов Поста, она сама по себе формирует полную систему. В качестве примера можно назвать функцию Шеффера (отрицание конъюнкции).

Широко известны такие полные системы булевых функций:

- $\{ \wedge, \vee , \neg \}$ (конъюнкция, дизъюнкция, отрицание);
- $\{ \wedge , \oplus , 1\}$(конъюнкция, сложение по модулю 2, константа 1);
- $\{ \wedge , \neg \}$(конъюнкция, отрицание);
- $\{ \vee , \neg \}$(дизъюнкция, отрицание);
- $\{ \downarrow \}$ (стрелка Пирса);
- {|}(штрих Шеффера).

Первая система используется, например, для представления функций в виде дизъюнктивных и конъюнктивных нормальных форм, вторая — для представления в виде полиномов Жегалкина.

Менее известные другие полные системы булевых функций:

- $\{\rightarrow, \neg \}$ (импликация, отрицание);
- $\{\rightarrow, \oplus \}$(импликация, сложение по модулю 2);
- $(\rightarrow, 0\}$(импликация, константа 0);

Полная система функций называется **базисом**, если она перестаёт быть полной при исключении из неё любого элемента. Первая из упоминавшихся выше полных систем базисом не является, поскольку согласно законам де Моргана либо дизъюнкцию, либо конъюнкцию можно исключить из системы и восстановить с помощью остальных двух функций. Вторая система является базисом — все три её элемента

необходимы для полноты. Максимально возможное число булевых функций в базисе — 4.

Иногда говорят о системе функций, полной в некотором замкнутом классе, и соответственно о базисе этого класса. Например, систему $\{\oplus, 1\}$ можно назвать базисом класса линейных функций.

2.6.2. Тождественность и двойственность

Две булевы функции тождественны друг другу, если на любых одинаковых наборах аргументов они принимают равные значения. Тождественность функций f и g можно записать, например, так:

$f(x_1,x_2, \ldots. x_n) = g(x_1,x_2, \ldots. x_n)$

Функция $g(x_1,x_2, \ldots. x_n)$ называется двойственной функции $f(x_1,x_2, \ldots. x_n)$, если $f(\overline{x}_1,\overline{x}_2,\ldots.\overline{x}_n) = \overline{g(x_1,x_2,\ldots.x_n)}$. Легко показать, что в этом равенстве f и g можно поменять местами, то есть функции f и g двойственны друг другу. Из простейших функций двойственны друг другу константы 0 и 1, а из законов де Моргана следует двойственность конъюнкции и дизъюнкции. Тождественная функция, как и функция отрицания, двойственна сама себе.

Если в булевом тождестве заменить каждую функцию на двойственную ей, снова получится верное тождество. В приведённых выше формулах легко найти двойственные друг другу пары.

2.6.3. Полнота системы, критерий Поста

Система булевых функций называется *полной*, если можно построить их суперпозицию, тождественную любой заранее заданной функции. Говорят ещё, что замыкание данной системы совпадает с множеством P_2.

Критерий Поста — одна из центральных теорем в теории булевых функций, устанавливающая необходимое и достаточное условие для

того, чтобы некоторый набор булевых функций обладал достаточной выразительностью, чтобы представить любую булеву функцию. Впервые сформулирован американским математиком Эмилем Постом.

В середине 60-х годов почти одновременно в СССР и во Франции появились работы, где с иных позиций и в более доступной форме излагались результаты Поста. В 80-е годы сразу целому ряду исследователей с использованием различных подходов и различной техники удалось получить достаточно компактные доказательства результатов Поста. Алгебраический подход в изучении замкнутых классов булевых функций (подалгебр итеративной алгебры Поста над множеством B={0, 1}) принадлежит *А. И. Мальцеву*.

Пост ввёл в рассмотрение следующие замкнутые классы булевых функций:

- Функции, сохраняющие константу P_0 и P_1;
- Самодвойственные функции S;
- Монотонные функции M;
- Линейные функции L.

Им было доказано, что любой замкнутый класс булевых функций, не совпадающий с P_2, целиком содержится в одном из этих пяти так называемых *предполных классов*, но при этом ни один из пяти не содержится целиком в объединении четырёх других. Таким образом, критерий Поста для полноты системы сводится к выяснению, не содержится ли вся эта система целиком в одном из предполных классов. Если для каждого класса в системе найдётся функция, не входящая в него, то такая система будет полной, и с помощью входящих в неё функций можно будет получить любую другую булеву функцию. Пост доказал, что множество замкнутых классов булевых функций — счётное множество.

2.7. Представление булевых функций

Теорема Поста открывает путь к представлению булевых функций синтаксическим способом, который в ряде случаев оказывается намного удобнее, чем таблицы истинности. Отправной точкой здесь служит нахождение некоторой полной системы функций $\sum = \{f_1,\ldots,f_n\}$.

Тогда каждая булева функция может быть представлена некоторым термом в сигнатуре $\sum$, который в данном случае называют также формулой. Относительно выбраной системы функций полезно знать ответы на следующие вопросы:

- Как построить по данной функции представляющую её формулу?
- Как проверить, что две разные формулы эквивалентны, то есть задают одну и ту же функцию?
  - В частности: существует ли способ приведения произвольной формулы к эквивалентной ей *канонической* форме такой, что две формулы эквивалентны тогда и только тогда, когда их канонические формы совпадают?
- Как по данной функции построить представляющую её формулу с теми или иными заданными свойствами (например, наименьшего размера), и возможно ли это?

Положительные ответы на эти и другие вопросы существенно увеличивают прикладное значение выбранной системы функций.

2.7.1. Дизъюнктивная нормальная форма (ДНФ)

Простой конъюнкцией или *конъюнктом* называется конъюнкция некоторого конечного набора переменных или их отрицаний, причём каждая переменная встречается не более одного раза. *Дизъюнктивной нормальной формой* или *ДНФ* называется дизъюнкция простых конъюнкций. Элементарная конъюнкция

- правильная, если каждая переменная входит в неё не более одного раза (включая отрицание);
- полная, если каждая переменная (или её отрицание) входит в неё ровно 1 раз;
- монотонная, если она не содержит отрицаний переменных.

Совершенной дизъюнктивной нормальной формой или *СДНФ* относительно некоторого заданного конечного набора переменных называется такая ДНФ, у которой в каждую конъюнкцию входят все переменные данного набора, причём в одном и том же порядке.

Легко убедиться, что каждой булевой функции соответствует некоторая ДНФ, а функции, отличной от тождественного нуля — даже СДНФ. Для этого достаточно в таблице истинности этой функции найти все булевы векторы, на которых её значение равно 1, и для каждого такого вектора построить конъюнкцию . Дизъюнкция этих конъюнкций является СДНФ исходной функции, поскольку на всех булевых векторах её значения совпадают со значениями исходной функции.

Дизъюнкти́вная норма́льная фо́рма (ДНФ) в булевой логике — нормальная форма, в которой булева формула имеет вид дизъюнкции конъюнкций литералов. Любая булева формула может быть приведена к ДНФ. Для этого можно использовать закон двойного отрицания, закон де Моргана, закон дистрибутивности. Дизъюнктивная нормальная форма удобна для автоматического доказательства теорем.

Примеры и контрпримеры

Формулы **в ДНФ**:

$A \vee B$
$(A \wedge B) \vee \neg A$
$(A \wedge B \wedge \neg C) \vee (\neg D \wedge E \wedge F) \vee (C \wedge D) \vee B$

Формулы **не в ДНФ**:

$\neg (A \vee B)$
$A \vee (B \wedge (C \vee D)$

Построение ДНФ

Алгоритм построения ДНФ

1) Избавиться от всех логических операций, содержащихся в формуле, заменив их основными: конъюнкцией, дизъюнкцией, отрицанием. Это можно сделать, используя равносильные формулы:

$A \to B = \neg A \vee B$
$A \leftrightarrow B = (A \wedge B) \vee (\neg A \wedge \neg B)$

2) Заменить знак отрицания, относящийся ко всему выражению, знаками отрицания, относящимися к отдельным переменным высказываниям на основании формул:

$$\neg (A \vee B) = \neg A \wedge \neg B$$
$$\neg (A \wedge B) = \neg A \vee \neg B$$

3) Избавиться от знаков двойного отрицания.

4) Применить, если нужно, к операциям конъюнкции и дизъюнкции свойства дистрибутивности и формулы поглощения.

Пример построения ДНФ

Приведем к ДНФ формулу

$$F = \neg ((X \to Y) \vee \neg (Y \to Z))$$

Выразим логическую операцию → через $\vee \wedge \neg$

$$F = \neg ((\neg X \vee Y) \vee \neg (\neg Y \vee Z))$$

В полученной формуле перенесем отрицание к переменным и сократим двойные отрицания:

$$F = (\neg\neg X \wedge \neg Y) \wedge (\neg\neg Y \vee Z) = (X \wedge \neg Y) \wedge (\neg\neg Y \vee Z)$$

Используя закон дистрибутивности, получаем:

$$F = (X \wedge \neg Y \wedge \neg\neg Y) \vee (X \wedge \neg\neg Y \wedge Z)$$

Используя идемпотентность конъюкции, получаем ДНФ:

$$F = (X \wedge \neg Y) \vee (X \wedge \neg\neg Y \wedge Z)$$

***k*-дизъюнктивная нормальная форма**

k-дизъюнктивной нормальной формой называют дизъюнктивную нормальную форму, в которой каждая конъюнкция содержит ровно *k* литералов.

Например, следующая формула записана в 2-ДНФ:

$$(A \wedge B) \vee (\neg\neg B \wedge C) \vee (B \wedge \neg\neg C)$$

Переход от ДНФ к СДНФ

Если в какой-то простой конъюнкции недостаёт переменной, например, Z, вставляем в неё выражение

$$Z \vee \neg\neg Z = 1,$$

после чего раскрываем скобки (при этом повторяющиеся дизъюнктные слагаемые не пишем, так как $Z \vee Z = Z$ по акону идемпотентности). Например:

$$X \vee \neg\neg Y \neg\neg Z = X(Y \vee \neg\neg Y)(Z \vee \neg\neg Z) \vee (X \vee \neg\neg X) \neg\neg Y \neg\neg Z =$$
$$XYZ \vee X \neg\neg YZ \vee XY \neg\neg Z \vee X \neg\neg Y \neg\neg Z \vee X \neg\neg Y \neg\neg Z \vee$$
$$\neg\neg X \neg\neg Y \neg\neg Z = XYZ \vee X \neg\neg YZ \vee XY \neg\neg Z \vee \neg\neg X \neg\neg Y \neg\neg Z$$

Таким образом, из ДНФ получили СДНФ.

Формальная грамматика, описывающая ДНФ

Следующая формальная грамматика описывает все формулы, приведенные к ДНФ:

<ДНФ> → <конъюнкт>
<ДНФ> → <ДНФ> ∨ <конъюнкт>
<конъюнкт> → <литерал>
<конъюнкт> → (<конъюнкт> ∧ <литерал>)
<литерал> → <терм>
<литерал> → ¬<терм>

где <терм> обозначает произвольную булеву переменную.

2.7.2. Конъюнктивная нормальная форма (КНФ)

Конъюнктивная нормальная форма1 (КНФ) определяется двойственно к ДНФ. *Простой дизъюнкцией* или *дизъюнктом* называется дизъюнкция одной или нескольких переменных или их отрицаний, причём каждая переменная входит в неё не более одного раза. КНФ — это конъюнкция простых дизъюнкций.

Совершенной конъюнктивной нормальной формой (СКНФ), относительно некоторого заданного конечного набора переменных, называется такая КНФ, у которой в каждую дизъюнкцию входят все переменные данного набора, причём в одном и том же порядке. Поскольку (С)КНФ и (С)ДНФ взаимодвойственны, свойства (С)КНФ повторяют все свойства (С)ДНФ, грубо говоря, «с точностью до наоборот».

КНФ может быть преобразована к эквивалентной ей ДНФ путём раскрытия скобок по правилу:

которое выражает дистрибутивность конъюнкции относительно дизъюнкции. После этого необходимо в каждой конъюнкции удалить повторяющиеся переменные или их отрицания, а также выбросить из дизъюнкции все конъюнкции, в которых встречается переменная вместе со своим отрицанием. При этом результатом не обязательно будет СДНФ, даже если исходная КНФ была СКНФ. Точно также можно всегда перейти от ДНФ к КНФ. Для этого следует использовать правило

выражающее дистрибутивность дизъюнкции относительно конъюнкции. Результат нужно преобразовать описанным выше способом, заменив слово «конъюнкция» на «дизъюнкция» и наоборот.

Конъюнкти́вная норма́льная фо́рма (**КНФ**) в булевой логике — нормальная форма, в которой булева формула имеет вид конъюнкции

дизъюнкций литералов. Конъюнктивная нормальная форма удобна для автоматического доказательства теорем. Любая булева формула может быть приведена к КНФ. Для этого можно использовать: закон двойного отрицания, закон де Моргана, дистрибутивность.

Примеры и контрпримеры

Формулы **в КНФ**:

$$\neg A \wedge (B \vee C),$$
$$(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E),$$
$$A \wedge B$$

Формулы **не в КНФ**:

$$\neg (B \vee C)$$
$$(A \vee B) \vee C$$
$$A \vee (B \vee (D \wedge E))$$

Но эти 3 формулы не в КНФ эквивалентны следующим формулам в КНФ:

$$\neg B \wedge \neg C,$$
$$(A \vee C) \wedge (B \vee C)$$
$$A \wedge (B \vee D) \wedge (B \vee E)$$

Построение КНФ

Алгоритм построения КНФ

1) Избавиться от всех логических операций, содержащихся в формуле, заменив их основными: конъюнкцией, дизъюнкцией, отрицанием. Это можно сделать, используя равносильные формулы:

$$A \to B = \neg A \vee B$$
$$A \leftrightarrow B = (\neg A \vee B) \wedge (A \vee \neg B)$$

2) Заменить знак отрицания, относящийся ко всему выражению, знаками отрицания, относящимися к отдельным переменным высказываниям на основании формул:

$\neg (A \vee B) = \neg A \wedge \neg B$
$\neg (A \wedge B) = \neg A \vee \neg B$

3) Избавиться от знаков двойного отрицания.

4) Применить, если нужно, к операциям конъюнкции и дизъюнкции свойства дистрибутивности и формулы поглощения.

Пример построения КНФ

Приведем к КНФ формулу

$F=(X \to Y) \wedge ((\neg Y \to Z) \to \neg X))$.

Преобразуем формулу F к формуле, не содержащей →:

$F= (\neg X \vee Y) \wedge (\neg\neg(\neg\neg Y \to Z) \vee \neg X) =$

$=(\neg X \vee Y) \wedge (\neg (\neg\neg \neg Y \vee Z) \vee \neg X)$.

В полученной формуле перенесем отрицание к переменным и сократим двойные отрицания:

$F= (\neg X \vee Y) \wedge ((\neg\neg Y \wedge \neg Z) \vee \neg X)$.

По закону дистрибутивности получим КНФ:

$F= (\neg X \vee Y) \wedge (\neg\neg X \vee \neg Y) \wedge (\neg\neg X \vee \neg\neg Z)$

***k*-конъюнктивная нормальная форма**

k-конъюнктивной нормальной формой называют конъюнктивную нормальную форму, в которой каждая дизъюнкция содержит ровно *k* литералов.

Например, следующая формула записана в 2-КНФ:

$(A \vee B) \wedge (\neg B C) \wedge (B \vee \neg C)$

Переход от КНФ к СКНФ

Если в простой дизъюнкции не хватает какой-то переменной (например, z), то добавляем в неё выражение $:Z \wedge \neg Z=0$(это не меняет самой дизъюнкции), после чего раскрываем скобки с использованием распределительного закона:

$(X \vee Y) \wedge (X \vee \neg\neg Y \vee \neg\neg Z) = ((X \vee Y \vee (Z \wedge \neg\neg Z)) \wedge$
$\wedge (X \vee \neg\neg Y \vee \neg\neg Z) = (X \vee Y \vee Z) \wedge (X \vee Y \vee \neg\neg Z) \wedge$
$\wedge (X \vee \neg\neg Y \vee \neg\neg Z)$

Таким образом, из КНФ получена СКНФ.

Формальная грамматика, описывающая КНФ

Следующая формальная грамматика описывает все формулы, приведенные к КНФ:

<КНФ> → <дизъюнкт>
<КНФ> → <КНФ> ∧ <дизъюнкт>
<дизъюнкт> → <литерал>;
<дизъюнкт> → (<дизъюнкт> ∨ <литерал>)
<литерал> → <терм>
<литерал> → ¬<терм>

где <терм> обозначает произвольную булеву переменную.

Задача выполнимости формулы в КНФ

В теории вычислительной сложности важную роль играет задача выполнимости булевых формул в конъюнктивной нормальной форме. Согласно теореме Кука, эта задача NP-полна, и она сводится к задаче о выполнимости формул в 3-КНФ, которая сводится и к которой в свою очередь сводятся другие NP-полные задачи.

Задача о выполнимости 2-КНФ формул может быть решена за линейное время.

2.7.3. Алгебраическая нормальная форма (АНФ или полином Жегалкина)

Алгебраическая нормальная форма (общепринятое название в зарубежной литературе) или полином Жегалкина (название, используемое в отечественной литературе) — это форма представления логической функции в виде полинома с коэффициентами вида 0 и 1, в котором в качестве произведения используется операция конъюнкции («И», AND), а в качестве сложения — сложение по модулю 2 (исключающее «ИЛИ», XOR). Для получения полинома Жегалкина следует выполнить следующие действия:

1. Получить СДНФ функции
2. Все ИЛИ заменить на Исключающее ИЛИ
3. Во всех термах заменить элементы с отрицанием на конструкцию: («элемент» «исключающее ИЛИ» 1)
4. Раскрыть скобки по правилам алгебры Жегалкина и привести попарно одинаковые термы

Полином Жегалкина — многочлен над кольцом $\mathbf{Z}_2$, то есть полином с коэффициентами вида 0 и 1, где в качестве произведения берётся конъюнкция, а в качестве сложения — исключающее или. Полином был предложен в 1927 году Иваном Жегалкиным в качестве удобного средства для представления функций булевой логики. В зарубежной литературе представление в виде полинома Жегалкина обычно называется алгебраической нормальной формой (АНФ).

Теорема Жегалкина — утверждение о существовании и единственности представления всякой булевой функции в виде полинома Жегалкина.
Полином Жегалкина представляет собой сумму по модулю два произведений неинвертированных переменных, а также (если необходимо) константы 1. Формально полином Жегалкина можно представить в виде

$P(X_1 \ldots X_n)=a \oplus a_1X_1 \oplus a_2X_2 \oplus \ldots \oplus a_nX_n \oplus a_{12}X_1X_2 \oplus a_{13}X_1X_3 \oplus \ldots \oplus a_{1..n}X_1 \ldots X_n,$
$a, \ldots, a_{1..n} \in \{0, 1\}$

или в более формализованном виде как:

$$P = a \oplus \bigoplus_{\substack{1 \le i_1 < \ldots < i_k \le n \\ k \in 1,\ldots,n}} a_{i_1,\ldots,i_k} \wedge x_{i_1} \wedge \ldots \wedge x_{i_k},$$
$$a, a_{i_1,\ldots,i_k} \in \{0,1\}.$$

Примеры полиномов Жегалкина:

$P=B \oplus AB;$
$P=X \oplus YZ \oplus ABX \oplus ABDYZ;$
$P=1 \oplus A \oplus ABD.$

Предпосылки

По теореме Поста, чтобы система булевых функций была полной, надо, чтобы в ней существовали:

1. Хотя бы одна функция, не сохраняющая 0.
2. Хотя бы одна функция, не сохраняющая 1.
3. Хотя бы одна нелинейная функция.
4. Хотя бы одна немонотонная функция.
5. Хотя бы одна несамодвойственная функция.

Этому требованию отвечает, в частности, система функций $<\wedge, \oplus, 1>$(конъюнкция, сложение по модулю два, константа 1). На её основе и строятся полиномы Жегалкина.

Существование и единственность представления

По теореме Жегалкина каждая булева функция единственным образом представляется в виде полинома Жегалкина. Теорема доказывается следующим образом. Заметим, что различных булевых функций от n

переменных 2^{2^n} штук. При этом конъюнкций вида $x_{i_1}...x_{i_k}$ существует ровно 2^n, так как из *n* возможных сомножителей каждый или входит в конъюнкцию, или нет. В полиноме у каждой такой конъюнкции стоит 0 или 1, то есть существует 2^{2^n} различных полиномов Жегалкина от *n* переменных. Теперь достаточно лишь доказать, что различные полиномы реализуют различные функции. Предположим противное. Тогда приравняв два различных полинома и перенеся один из них в другую часть равенства, получим полином, тождественно равный нулю и имеющий ненулевые коэффициенты. Тогда рассмотрим слагаемое с единичным коэффициентом наименьшей длины, то есть с наименьшим числом переменных, входящих в него (любой один, если таких несколько). Подставив единицы на места этих переменных, и нули на места остальных, получим, что на этом наборе только одно это слагаемое принимает единичное значение, то есть нулевая функция на одном из наборов принимает значение 1. Противоречие. Значит, каждая булева функция реализуется полиномом Жегалкина единственным образом.

Представление функции в виде полинома Жегалкина

С помощью эквивалентных преобразований ДНФ

По сравнению с ДНФ в полиноме Жегалкина отсутствуют операции ИЛИ и НЕ. Таким образом, полином Жегалкина можно получить из ДНФ, выразив операции ИЛИ и НЕ через операции сложение по модулю два, и константу 1. Для этого применяются следующие соотношения:

$A \vee B = A \oplus B \oplus AB$;

$\overline{A} = A \oplus 1$.

Ниже приведён пример преобразования ДНФ в полином Жегалкина:

$XY \vee \overline{X}\overline{Y} = XY \oplus \overline{X}\overline{Y} \oplus XY\,\overline{X}\overline{Y} = XY \oplus \overline{X}\overline{Y} = XY \oplus (X \oplus 1)(Y \oplus 1) = XY \oplus XY \oplus X \oplus Y \oplus 1 = X \oplus Y \oplus 1$.

При преобразованиях использованы соотношения:

$A \oplus A = 0$;

$(A \oplus B)C = AC \oplus BC$.

С помощью эквивалентных преобразований СДНФ

СДНФ обладает тем свойством, что при любых значениях входных переменных в единицу обращается не более одного члена выражения. Для таких выражений операция дизъюнкции эквивалентна операции сложение по модулю два.

При преобразовании СДНФ в полином Жегалкина, достаточно заменить все дизъюнкции на операции сложение по модулю два и избавиться от инверсий при помощи эквивалентного преобразования

$\overline{A} = A \oplus 1$.

С помощью карты Карно

Логическая функция трёх переменных P(A,B,C), представленная в виде карты Карно, преобразуется в полином Жегалкина следующими шагами:

- Рассматриваем все ячейки карты Карно в порядке возрастания количества единиц в их кодах. Для функции трёх переменных последовательность ячеек будет 000—100 — 010—001 — 110—101 — 011—111. Каждой ячейке карты Карно сопоставляем член полинома Жегалкина в зависимости от позиций кода, в которых стоят единицы. Например, ячейке 111 соответствует член ABC, ячейке 101 — член AC, ячейке 010 — член B, ячейке 000 — член 1.
- Если в рассматриваемой ячейке находится 0, переходим к следующей ячейке.
- Если в рассматриваемой ячейке находится 1, добавляем в полином Жегалкина соответствующий член, инвертируем в карте Карно все ячейки, где этот член равен 1 и переходим к следующей ячейке. Например, если при рассмотрении ячейки

110 в ней оказывается единица, то в полином Жегалкина добавляется член AB и инвертируются все ячейки карты Карно, где A=1 и B=1. Если единице равна ячейка 000, то в полином Жегалкина добавляется член 1 и инвертируется вся карта Карно.
- Процесс преобразования можно считать законченным, когда после очередной инверсии все ячейки карты Карно становятся нулевыми.

Метод треугольника

Метод треугольника (также называется методом треугольника Паскаля) позволяет преобразовать таблицу истинности в полином Жегалкина путём построения вспомогательной треугольной таблицы в соответствии со следующими правилами:

- Строится полная таблица истинности, в которой строки идут в порядке возрастания двоичных кодов от 000…00 до 111…11.
- Строится вспомогательная треугольная таблица, в которой первый столбец совпадает со столбцом значений функции в таблице истинности.
- Ячейка в каждом последующем столбце получается путём суммирования по модулю 2 двух ячеек предыдущего столбца — стоящей в той же строке и строкой ниже.
- Столбцы вспомогательной таблицы нумеруются двоичными кодами в том же порядке, что и строки таблицы истинности.
- Каждому двоичному коду ставится в соответствие один из членов полинома Жегалкина в зависимости от позиций кода, в которых стоят единицы. Например, ячейке 111 соответствует член ABC, ячейке 101 — член AC, ячейке 010 — член B, ячейке 000 — член 1 и т. д.
- Если в верхней строке какого-либо столбца стоит единица, то соответствующий член присутствует в полиноме Жегалкина.

Метод Паскаля

Наиболее экономным с точки зрения объёма вычислений и целесообразным для построения полинома Жегалкина вручную является метод Паскаля.

Строим таблицу, состоящую из 2^N столбцов и N+1 строк, где N — количество переменных в функции. В верхней строке таблицы размещаем вектор значений функции, то есть последний столбец таблицы истинности.

Каждую строку полученной таблицы разбиваем на блоки (чёрные линии на рисунке). В первой строке блок занимает одну клетку, во второй строке — две, в третьей — четыре, в четвёртой — восемь и т. д. Каждому блоку в некоторой строке, который мы будем называть «нижний блок», всегда соответствует ровно два блока в предыдущей строке. Будем называть их «левый верхний блок» и «правый верхний блок».

Построение начинается со второй строки. Содержимое левых верхних блоков без изменения переносится в соответствующие клетки нижнего блока (зелёные стрелки на рисунке). Затем над правым верхним и левым верхним блоками побитно производится операция «Исключающее ИЛИ» и полученный результат переносится в соответствующие клетки правой части нижнего блока (красные стрелки на рисунке). Это операция проводится со всеми строками сверху вниз и со всеми блоками в каждой строке. После окончания построения в нижней строке оказывается строка чисел, которая является коэффициентами полинома Жегалкина, записанными в той же последовательности, что и в описанном выше методе треугольника.

Метод суммирования

Графическое представление коэффициентов полинома Жегалкина для функций с разным числом переменных

По таблице истинности легко вычислить отдельные коэффициенты полинома Жегалкина. Для этого нужно просуммировать по модулю 2 значения функции в тех строках таблицы, где переменные, отсутствующие в конъюнкции, принимают нулевые значения.

Предположим для примера, что нужно найти коэффициент при конъюнкции xz для функции трёх переменных f(x, y,z). В этой конъюнкции отсутствует переменная y. Найдём входные наборы, в

которых переменная y принимает нулевое значение. Это наборы 0, 1, 4, 5 (000, 001, 100, 101). Тогда коэффициент при конъюнкции xz равен

$a_5=f_0 \oplus f_1 \oplus f_4 \oplus f_5=f(0,0,0) \oplus f(0,0,1) \oplus f(1,0,0) \oplus f(1,0,1)$.

Поскольку с свободном члене отсутствуют все переменные, то

$a_0=f_0$.

Для члена, куда входят все переменные, в сумму входят все значения функции:

$a_{N-1}=f_0+f_1+f_2+\ldots+f_{N-2}+f_{N-1}$.

Представим графически коэффициенты полинома Жегалкина как суммы по модулю 2 значений функций в некоторых точках. Для этого построим квадратную таблицу, где каждый столбец представляет собой значение функции в одной из точек, а строка — коэффициент полинома Жегалкина. Точка на пересечении некоторого столбца и строки означает, что значение функции в данной точке входит в сумму для данного коэффициента полинома (см. рисунок). Назовём эту таблицу T_N, где N — число переменных функции.

Существует закономерность, которая позволяет получить таблицу для функции N переменных, имея таблицу для функции N-1 переменных. Новая таблица T_{N+1} компонуется как матрица 2×2 таблиц T_N, причём правый верхний блок матрицы очищается.

2.8. Классификация булевых функций

- По количеству n входных операндов, от которых зависит значение на выходе функции, различают нульарные ($n = 0$), унарные ($n = 1$), бинарные ($n = 2$), тернарные ($n = 3$) булевы функции и функции от большего числа операндов.

- По количеству единиц и нулей в таблице истинности отличают узкий класс сбалансированных булевых функций (также называемых уравновешенными или равновероятностными, поскольку при равновероятных случайных значениях на входе

или при переборе всех комбинаций по таблице истинности вероятность получения на выходе значения **1** равна 1/2) от более широкого класса несбалансированных булевых функций (так же называемых неуравновешенными, поскольку вероятность получения на выходе значения **1** отлична от 1/2). Сбалансированные булевы функции в основном используются в криптографии.

- По зависимости значения функции от перестановки её входных битов различают симметричные булевы функции (значение которых зависит только от количества единиц на входе) и несимметричные булевы функции (значение которых так же зависит от перестановки её входных бит).

- По значению функции на противоположных друг другу наборах значений аргументов отличают самодвойственные функции (значение которых инвертируется при инвертировании значения всех входов) от остальных булевых функций, не обладающих таким свойством. Нижняя часть таблицы истинности для самодвойственных функций является зеркальным отражением инвертированной верхней части (если расположить входные комбинации в таблице истинности в естественном порядке).

- По алгебраической степени нелинейности отличают линейные булевы функции (АНФ которых сводится к линейной сумме по модулю 2 входных значений) и нелинейные булевы функции (АНФ которых содержит хотя бы одну нелинейную операцию конъюнкции входных значений). Примерами линейных функций являются: сложение по модулю 2 (исключающее «ИЛИ», XOR), эквивалентность, а также все булевы функции, АНФ которых содержит лишь линейные операции сложения по модулю 2 без конъюнкций. Примерами нелинейных функций являются: конъюнкция («И», AND), штрих Шеффера («НЕ-И», NAND), стрелка Пирса («НЕ-ИЛИ», NOR), а также все булевы функции, АНФ которых содержит хотя бы одну нелинейную операцию конъюнкции.

3. Алгебра логики

Не следует путать с булевой алгеброй.

Алгебра логики (алгебра высказываний) — раздел математической логики, в котором изучаются логические операции над высказываниями. Чаще всего предполагается, что высказывания могут быть только истинными или ложными, то есть используется так называемая *бинарная* или *двоичная* логика, в отличие от, например, троичной логики.

3.1.Определение

Базовыми элементами, которыми оперирует алгебра логики, являются высказывания.

Высказывания строятся над множеством $\{B, \neg, \wedge, \vee, 0, 1\}$, где B — непустое множество, над элементами которого определены три операции:

$\neg$ *отрицание* (унарная операция),
$\wedge$ *конъюнкция* (бинарная),
$\vee$ *дизъюнкция* (бинарная),

а логический ноль **0** и логическая единица **1** — константы.

Так же используются названия

- *Дизъю́нкт* — пропозициональная формула, являющаяся дизъюнкцией одного или более литералов (например $x_1 \vee \neg x_2 \vee x_4$).
- *Конъюнкт* — пропозициональная формула, являющаяся конъюнкцией одного или более литералов (например $x_1 \wedge \neg x_2 \wedge x_4$).

Унарная операция отрицания в тексте формул оформляется либо в виде значка перед операндом ($\neg x$) либо в виде черты над операндом ($\overline{x}$), что компактнее, но в целом менее заметно.

3.2. Аксиомы

1. $\overline{\overline{x}} = x$, инволютивность отрицания, закон снятия двойного отрицания
2. $x \vee \overline{x} = 1$
3. $x \vee 1 = 1$
4. $x \vee x = x$
5. $x \vee 0 = x$
6. $x \wedge \overline{x} = 0$
7. $x \wedge z = x$
8. $x \wedge 0 = 0$
9. $x \wedge 1 = x$

3.3. Логические операции

Простейший и наиболее широко применяемый пример такой алгебраической системы строится с использованием множества B, состоящего всего из двух элементов:

B = { Ложь, Истина }

Как правило, в математических выражениях **Ложь** отождествляется с логическим нулём, а **Истина** — с логической единицей, а операции отрицания (НЕ), конъюнкции (И) и дизъюнкции (ИЛИ) определяются в привычном нам понимании. Легко показать, что на данном множестве B можно задать четыре унарные и шестнадцать бинарных отношений и все они могут быть получены через суперпозицию трёх выбранных операций.

Опираясь на этот математический инструментарий, логика высказываний изучает высказывания и предикаты. Также вводятся дополнительные операции, такие как эквиваленция $\leftrightarrow$(«тогда и только тогда, когда»), импликация $\rightarrow$(«следовательно»), сложение по модулю два $\oplus$ («исключающее или»), штрих Шеффера |, стрелка Пирса $\downarrow$и другие.

Логика высказываний послужила основным математическим инструментом при создании компьютеров. Она легко преобразуется в битовую логику: истинность высказывания обозначается одним битом

(0 — ЛОЖЬ, 1 — ИСТИНА); тогда операция $\neg$ приобретает смысл вычитания из единицы; $\vee$ — немодульного сложения; & — умножения; $\leftrightarrow$ — равенства; $\oplus$ — в буквальном смысле сложения по модулю 2 (исключающее Или — XOR); | — непревосходства суммы над 1 (то есть

A |B = (A + B) <= 1).

Впоследствии булева алгебра была обобщена от логики высказываний путём введения характерных для логики высказываний аксиом. Это позволило рассматривать, например, логику кубитов, тройственную логику (когда есть три варианта истинности высказывания: «истина», «ложь» и «не определено»), комплексную логику и др.

3.4. Свойства логических операций

1. Коммутативность: $x \circ y = y \circ x$. $\circ \in \{\wedge, \vee, \oplus, \sim, |, \downarrow\}$.
2. Идемпотентность: $x \circ x = x$, $\circ \in \{\wedge, \vee\}$.
3. Ассоциативность: $(x \circ y) \circ z = x \circ (y \circ z)$, $\circ \in \{\wedge, \vee, \oplus, \sim\}$.
4. Дистрибутивность конъюнкций и дизъюнкции относительно дизъюнкции, конъюнкции и суммы по модулю два соответственно:
   - $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$,
   - $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$,
   - $x \wedge (y \oplus z) = (x \wedge y) \oplus (x \wedge z)$.
5. Законы де Мóргана:
   - $\overline{x \wedge y} = \overline{x} \vee \overline{y}$,
   - $\overline{x \vee y} = \overline{x} \wedge \overline{y}$.
6. Законы поглощения:
   - $x \wedge (x \vee y) = x$,
   - $x \vee (x \wedge y) = x$.
7. Другие (1):
   - $x \wedge \overline{x} = x \wedge 0 = x \oplus x = 0$.
   - $x \vee \overline{x} = x \vee 1 = x \sim x = x \rightarrow x = 1$
   - $x \vee x = x \wedge x = x \wedge 1 = x \vee 0 = x \oplus 0 = x$.
   - $x \oplus 1 = x \rightarrow 0 = x \sim 0 = x | x = x \downarrow x = \overline{x}$.

   - $\overline{\overline{x}} = x$, инволютивность отрицания, закон снятия двойного отрицания.
8. Другие (2):
   - $x \oplus y = x \wedge \overline{y} \vee \overline{x} \wedge y = (x \vee y) \wedge (\overline{x} \vee \overline{y})$
   - $x \sim y = \overline{x \oplus y} = 1 \oplus x \oplus y = x \wedge y \vee \overline{x} \wedge \overline{y} = (x \vee \overline{y}) \wedge (\overline{x} \vee y)$.
   - $x \rightarrow y = y = x \wedge y \oplus x \oplus 1$.
   - $x \vee y = x \oplus y \oplus x \wedge y$.
9. Другие (3) (Дополнение законов де Мóргана):
   - $x|y = \overline{x \wedge y} = \overline{x} \vee \overline{y}$,
   - $x \downarrow y = \overline{x \vee y} = \overline{x} \wedge \overline{y}$.

Существуют методы упрощения логической функции: например, Карта Карно, метод Куайна - Мак-Класки

4. Булевы функции

4.1. Булевы функции

Булевы функции находят применение в конструировании и упрощении логических схем. Такие схемы встречаются в электронных устройствах, используемых в компьютерах, калькуляторах, телефонных системах и ряде других устройств.

Обозначим множество {0;1} через $\mathbb{Z}_2$, т. е. $\mathbb{Z}_2 = \{0;1\}$.

Функция f из множества $E_2^n \to E_2$ называется ***булевой функцией n переменных***. Напомним, что

$$E_2^n = \underbrace{E_2 \times E_2 \times \ldots \times E_2}_{n \text{ раз}}$$

Переменные булевых функций могут принимать только значения 0 или 1 и называются ***булевыми переменными***.

Множества всех булевых функции n переменных обозначается P_n, т.е.

$$P_n = \{f \mid f : E_2^n \to E_2\}.$$

Количество всех булевых функции n переменных находится по формуле

$$|P_n| = 2^{2^n}.$$

Например, булевых функции 1 переменной

$$|P_1| = 2^{2^1} = 4,$$

булевых функции 2 переменных

$$|P_2| = 2^{2^2} = 16,$$

булевых функции 3 переменных

$$|P_3| = 2^{2^3} = 256.$$

Булевы функции часто задаются таблично. Эти таблицы напоминают таблицы истинности логических операций, поэтому сами булевы функции часто называют ***булевыми операциями***, а соответствующие им таблицы - ***таблицами истинности.***

Булевы функции одной переменной

		Значения переменной x	0	1
	Название функции	**Обозначение функции**	**Значения функции**	
f_1	Тождественный нуль	0	0	0
f_2	Тождественная	x	0	1
f_3	Отрицание	$\overline{x}$	1	0

f_4	Тождественная единица	1	1	1

Булевы функции двух переменных

		Значения переменных	x_1	0	0	1	1
			x_2	0	1	0	1
	Название функции	**Обозначение функции**		**Значения функции**			
f_1	Тождественный нуль	0		0	0	0	0
f_2	Конъюнкция	&, $\wedge$, ·		0	0	0	1
f_3	Отрицание импликации	$\overline{x_1 \to x_2}$		0	0	1	0
f_4	Тождественная первой переменной	x_1		0	0	1	1
f_5	Отрицание импликации	$\overline{x_2 \to x_1}$		0	1	0	0
f_6	Тождественная второй переменной	x_2		0	1	0	1
f_7	Сумма по модулю два, строгая дизъюнкция	$\oplus$, $\veebar$		0	1	1	0
f_8	Дизъюнкция	$\vee$		0	1	1	1
f_9	Стрелка Пирса	$\downarrow$		1	0	0	0

f_{10}	Эквиваленция	$\leftrightarrow$, $\equiv$, ~	1	0	0	1
f_{11}	Инверсия второй переменной	$\overline{x_2}$	1	0	1	0
f_{12}	Импликация	$x_2 \to x_1$	1	0	1	1
f_{13}	Инверсия первой переменной	$\overline{x_1}$	1	1	0	0
f_{14}	Импликация	$x_1 \to x_2$	1	1	0	1
f_{15}	Штрих Шеффера	$x_1 \mid x_2$	1	1	1	0
f_{16}	Тождественная единица	1	1	1	1	1

Как уже говорилось ранее, имеется 256 булевых функции 3 переменных. Перечислять их все нет необходимости, приведем лишь примеры задания такой функции:

$$f(x_1 x_2 x_3) = 10110011,$$

$$f(x_1 x_2 x_3) = 11111111$$ (тождественная единица) и др.

4.2. Реализация функций формулами

Так же, как составные высказывания строятся из более простых, с помощью логических операций, можно комбинировать булевы переменные с помощью булевых операций, получая булевы выражения, которые называются ***формулами***.

Всякой формуле однозначно соответствует некоторая функция, при этом говорят, что ***формула реализует функцию.***

ПРИМЕР

Построить таблицу истинности для формулы $(x_1 \wedge x_2) \to x_1$.

x_1	x_2	$(x_1 \wedge x_2)$	$(x_1 \wedge x_2) \to x_1$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Таким образом, формула $(x_1 \wedge x_2) \to x_1$ реализует функцию $f_{16}(x_1 x_2) = 1111$ (тождественная единица).

ПРИМЕР

Построить таблицу истинности для формулы $((x_1 \wedge x_2) \oplus x_1) \oplus x_2$.

x_1	x_2	$(x_1 \wedge x_2)$	$((x_1 \wedge x_2) \oplus x_1)$	$((x_1 \wedge x_2) \oplus x_1) \oplus x_2$
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	0	1

Таким образом, формула $((x_1 \wedge x_2) \oplus x_1) \oplus x_2$ реализует функцию $f_8(x_1 x_2) = 0111$ (дизъюнкция).

4.3. Равносильные формулы

Формулы называются ***равносильными***, если реализуют одну и ту же функцию.

Формула называется ***тождественно-истинной*** или ***тавтологией***, если она реализует тождественную единицу.

Формула называется ***тождественно-ложной***, если она реализует тождественный ноль.

4.4.Законы булевой алгебры

Законами булевой алгебры называются следующие равносильности:

1. Идемпотентность

$$a + a = a, \quad a \cdot a = a$$

2. Коммутативность

$$a + b = b + a, \quad a \cdot b = a \cdot b$$

3. Ассоциативность

$$a + (b + c) = (a + b) + c = a + b + c,$$
$$a + (b + c) = (a + b) + c = a + b + c$$

4. Дистрибутивность

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c),$$
$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

5. Закон поглощения

$$a \vee (a \wedge b) = a, \quad a \wedge (a \vee b) = a$$

6. Закон склеивания

$$(a + b) \cdot (a + \bar{b}) = a, \quad (a \cdot b) + (a \cdot \bar{b}) = a$$

7. Закон нуля

$$a + 0 = a, \quad a \cdot 0 = 0$$

8. Закон единицы

$$a + 1 = 1, \quad a \cdot 1 = a$$

9. Закон дополнения

$$a + \bar{a} = 1, \quad a \cdot \bar{a} = 0$$

10. Инволютивность

$$\overline{\overline{a}} = a$$

11. Законы де Моргана

$$\overline{a+b}=\overline{a}\cdot\overline{b},\quad \overline{a\cdot b}=\overline{a}+\overline{b}$$

4.5. Принцип двойственности

Двойственной для булевой функции $f(x_1x_2\ldots x_n)$ называется булева функция

$$f^*(x_1x_2\ldots x_n)=\overline{f(\overline{x}_1\overline{x}_2\ldots\overline{x}_n)}$$

ПРИМЕР

$$0^*=\overline{0}=1,\ 1^*=\overline{1}=0,\ x^*=\overline{\overline{x}}=x,$$

$$(x\wedge y)^*=\overline{\overline{x}\wedge\overline{y}}\overset{\text{закон}}{\underset{\text{де Моргана}}{=}}x\vee y,$$

$$(x\vee y)^*=\overline{\overline{x}\vee\overline{y}}\overset{\text{закон}}{\underset{\text{де Моргана}}{=}}x\wedge y$$

Функция f называется ***самодвойственной*** если

$$f^*=f$$

ПРИМЕР

Функция $f(x)=x$ является самодвойственной, т.к.

$$x^*=\overline{\overline{x}}=x$$

ТЕОРЕМА (***Закон двойственности***)

Если формула f_1 равносильна формуле f_2, то формула f_1^* равносильна формуле f_2^*.

(Если две равносильные формулы заменить двойственными, то равносильность сохранится).

ТЕОРЕМА (***Принцип двойственности***)

Двойственная к булевой формуле может быть получена заменой констант **0** на **1**, **1** на **0**, **Ù** на **Ú**, **Ú** на **Ù** и сохранением структуры формулы (т.е. соответствующего порядка действий).

4.6. СДНФ и СКНФ

Определим степень следующим образом:

$$x^{\sigma}=\begin{cases} x, & \text{если } \sigma=1 \\ \bar{x}, & \text{если } \sigma=0 \end{cases}$$

, т.е. $x^0=\bar{x}$, $x^1=x$.

Выражение вида

$$x_1^{\sigma_1}\cdot x_2^{\sigma_2}\cdot\ldots\cdot x_n^{\sigma_n}$$

называется ***полной совершенной элементарной конъюнкцией.***

Можно дать другое определение: ***полной совершенной элементарной конъюнкцией*** называется конъюнкция переменных

функции или их отрицаний, причем никакая из переменных не входит вместе с отрицанием этой переменной.

Выражение вида

$$x_1^{\sigma_1}+x_2^{\sigma_2}+\ldots+x_n^{\sigma_n}$$

называется ***полной совершенной элементарной дизъюнкцией***.

Можно дать другое определение: ***полной совершенной элементарной дизъюнкцией*** называется дизъюнкция переменных функции или их отрицаний, причем никакая из переменных не входит вместе с отрицанием этой переменной.

Совершенной нормальной конъюнктивной формой (СКНФ) функции называется конъюнкция полных совершенных элементарных дизъюнкций.

Совершенной нормальной дизъюнктивной формой (СДНФ) функции называется дизъюнкция полных совершенных элементарных конъюнкций.

ПРИМЕР

Составим СДНФ и СКНФ для функции $x_1 \to x_2$.

В первой главе была приведена формула:

$$x_1 \to x_2 = \overline{x_1} + x_2,$$

таким образом, получили СКНФ для функции, состоящую из одной элементарной дизъюнкции.

Продолжим преобразования, получим

$$x_1 \to x_2 = \overline{x_1} + x_2 \overset{\text{закон единицы}}{=} \overline{x_1} \cdot 1 + x_2 \cdot 1 \overset{\text{закон дополнения}}{=} \overline{x_1} \cdot (x_2 + \overline{x_2}) + x_2 \cdot (x_1 + \overline{x_1}) \overset{\text{дистрибутивный закон}}{=}$$

$$= \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2} + x_2 \cdot x_1 + x_2 \cdot \overline{x_1} \overset{\text{идемпотентность}}{=} \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2} + x_2 \cdot x_1 =$$

$$\overset{\text{коммутативный закон}}{=} x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$

Таким образом, получили СДНФ для функции, состоящую из трех элементарной конъюнкции.

На этом примере покажем связь между таблицей истинности функции и ее совершенными нормальными формами:

x_1	x_2	$x_1 \to x_2$
0	0	1
0	1	1

1	0	0
1	1	1

СДНФ:

$$x_1 \to x_2 = x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2} =$$

$$x_1^{\,1} \cdot x_2^{\,1} + x_1^{\,0} \cdot x_2^{\,1} + x_1^{\,0} \cdot x_2^{\,0}$$

СКНФ:

$$x_1 \to x_2 = \overline{x_1} + x_2 = x_1^{\,0} + x_2^{\,1} = x_1^{\,\overline{1}} + x_2^{\,\overline{0}}.$$

При нахождении **СДНФ** пользуемся правилом: каждый набор аргументов определяет элементарную конъюнкцию, в которой значению 0 соответствует инверсия переменной, а значению 1 – сама переменная. СДНФ функции образуют те элементарные конъюнкции, которые соответствуют наборам аргументов, дающим 1.

x_1	x_2	$x_1 \to x_2$	элементарные конъюнкции
0	0	1	$\overline{x_1} \cdot \overline{x_2}$
0	1	1	$\overline{x_1} \cdot x_2$
1	0	0	$x_1 \cdot \overline{x_2}$
1	1	1	$x_1 \cdot x_2$

При нахождении **СКНФ** пользуемся правилом: каждый набор аргументов определяет элементарную дизъюнкцию, в которой значению 1 соответствует инверсия переменной, а значению 0 – сама переменная. СКНФ функции образуют те элементарные конъюнкции, которые соответствуют наборам аргументов, дающим 0.

x_1	x_2	$x_1 \to x_2$	элементарные дизъюнкции
0	0	1	$x_1 + x_2$
0	1	1	$x_1 + \overline{x_2}$
1	0	0	$\overline{x_1} + x_2$

1	1	1	$\overline{x_1} + \overline{x_2}$

5. Битовые операции

Би́товая опера́ция в программировании — некоторые операции над цепочками битов. В программировании, как правило, рассматриваются лишь некоторые виды этих операций: логические побитовые операции и битовые сдвиги. Битовые операции применяются в языках программирования и цифровой технике, изучаются в дискретной математике.

5.1. Побитовые логические операции

Ряд источников по языкам низкого уровня называет побитовые логические операции просто *логическими*, но в терминологии программирования на языках высокого уровня в названиях битовых операций присутствуют прилагательные *битовый*, *побитовый* (например: «побитовое логическое И», оно же «побитовое умножение»), *поразрядный*.

В некоторых языках программирования названия операторов, соответствующих логическим и побитовым логическим операциям, похожи. Кроме того, язык программирования может допускать неявное приведение числового типа к логическому и наоборот. В таких языках программирования необходимо внимательно следить за использованием логических и побитовых операций, перемешивание которых может привести к ошибкам. Например, в C++ результатом выражения «2 && 1» (логическое И) является булево значение *true*, а результатом выражения «2 & 1» (побитовое И) — целое значение *0*.

Побитовое отрицание (NOT)

Побитовое отрицание (или **побитовое НЕ**, или **дополнение**) — это унарная операция, действие которой эквивалентно применению

логического отрицания к каждому биту двоичного представления операнда. Другими словами, на той позиции, где в двоичном представлении операнда был 0, в результате будет 1, и, наоборот, где была 1, там будет 0. Например:

НЕ 01

10

Побитовое И (AND)

Побитовое И — это бинарная операция, действие которой эквивалентно применению логического И к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 1, результирующий двоичный разряд равен 1; если же хотя бы один бит из пары равен 0, результирующий двоичный разряд равен 0.

Пример:

И 0011
0101

0001

Побитовое ИЛИ (OR)

Побитовое ИЛИ — это бинарная операция, действие которой эквивалентно применению логического ИЛИ к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 0, двоичный разряд результата равен 0; если же хотя бы один бит из пары равен 1, двоичный разряд результата равен 1.

Пример:

ИЛИ 0011
0101

0111

Исключающее ИЛИ (XOR)

Исключающее ИЛИ (или **сложение по модулю 2**) — это бинарная операция, результат действия которой равен 1, если число складываемых единичных битов нечётно и равен 0, если чётно. Другими словами, если оба соответствующих бита операндов равны между собой, двоичный разряд результата равен 0; в противном случае, двоичный разряд результата равен 1.

Пример:

Искл. ИЛИ 0011
0101

0110

Первое русское название операции обусловлено тем, что результат данной операции отличается от результата «ИЛИ» только в одном случае из 4 случаев входа — обоих 1 (случай одновременной истинности аргументов «исключается»). Ещё в русской грамматике значение данной логической связки передаётся союзом «либо».

Второе название — тем, что действительно является сложением в кольце вычетов по модулю два, из чего следуют некоторые интересные свойства. Например, в отличие от вышеописанных «И» и «ИЛИ», данная операция является обратимой, или инволютивной:

$(x \oplus y) \oplus y = x.$

В компьютерной графике «сложение по модулю два» применяется при выводе спрайтов на картинку — повторное её применение убирает спрайт с картинки. Благодаря инволютивности эта же операция нашла применение в криптографии как простейшая реализация абсолютно стойкого шифра (шифра Вернама). «Сложение по модулю два» также может использоваться для обмена двух переменных, используя алгоритм обмена при помощи исключающего ИЛИ.

Также данная операция может называться «инверсией по маске», то есть у исходного двоичного числа инвертируются биты, которые совпадают с 1 в маске.

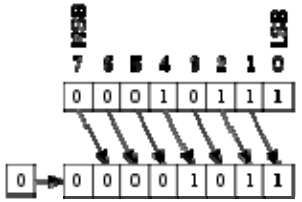
Другие побитовые логические операции

В распространённых языках программирования встроенными средствами реализуются только четыре побитовые логические операции: И, ИЛИ, НЕ и исключающее ИЛИ. Для задания произвольной побитовой логической операции вполне достаточно перечисленных, и, более того, как следует из теории булевых функций, можно ограничиться ещё меньшим набором базовых операций. Есть также языки программирования, где существует встроенная возможность выполнить любую бинарную логическую операцию побитово. Например, в PL/I есть встроенная функция BOOL, третий аргумент которой предназначен для указания произвольной логической операции, которую необходимо побитово применить к первым двум аргументам.

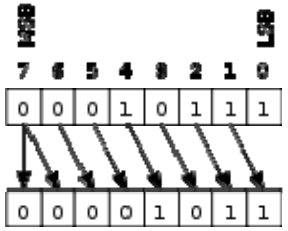
5.2. Битовые сдвиги

К битовым операциям также относят битовые сдвиги. При сдвиге значения битов копируются в соседние по направлению сдвига. Различают несколько видов сдвигов — *логический*, *арифметический* и *циклический*, в зависимости от обработки крайних битов.

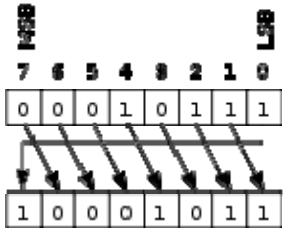
Также различают сдвиг *влево* (в направлении от младшего бита к старшему) и *вправо* (в направлении от старшего бита к младшему).



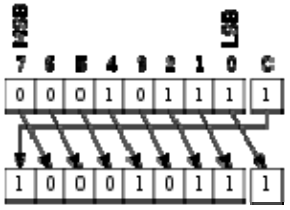
Логический сдвиг



Арифметический сдвиг (правый)



Циклический сдвиг



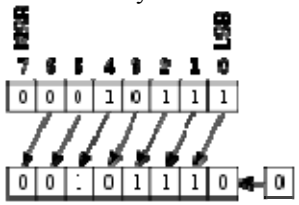
Циклический сдвиг через перенос

Логический сдвиг

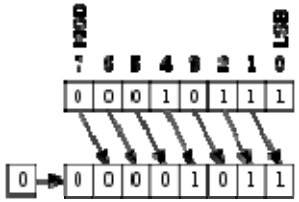
Сдвиг, при котором уходящий бит исчезает, не влияя на оставшиеся биты, а на месте появившегося бита записывается бит **0**.

Пример работы операции сдвига:

Пусть у нас есть число 10101010b (в двоичной системе).
Если сделать сдвиг влево на 1 бит, то получим число 01010100b.
Если сделать сдвиг исходного числа вправо на 1 бит, то получим число 01010101b.



Логический сдвиг влево



Логический сдвиг вправо

В большинстве процессоров уходящий бит сохраняется во флаге переноса. Эта функция широко используется при работе со многобайтовыми числами.

При логическом сдвиге значение последнего бита по направлению сдвига теряется (копируясь в бит переноса), а первый приобретает нулевое значение.

Арифметический сдвиг

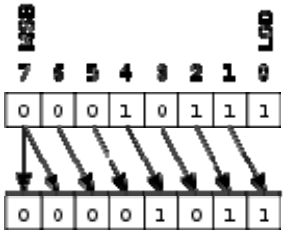
Арифметический сдвиг аналогичен логическому, но число считается знаковым, представленным в дополнительном коде. Так, при правом

сдвиге старший бит сохраняет своё значение. Левый арифметический сдвиг идентичен логическому.

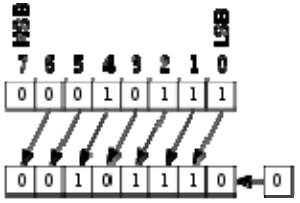
При этом сдвиге слово рассматривается не просто как группа битов, а как целое число в дополнительном коде. При сдвиге влево ведёт себя как логический сдвиг, при сдвиге вправо уходящий бит исчезает, не влияя на оставшиеся биты, а на месте появившегося бита устанавливается бит, соответствующий знаку.

Пример работы операции сдвига:

Пусть у нас есть число 11111010b = −6 (в двоичной системе).
Если сделать сдвиг влево на 1 бит, то получим число 11110100b = −12.
Если сделать сдвиг исходного числа вправо на 1 бит, то получим число 11111101b = −3.



Арифметический сдвиг влево



Арифметический сдвиг вправо

Легко заметить, что при арифметическом сдвиге сдвиг влево соответствует умножению на 2, а сдвиг вправо — делению на 2 (в общем случае — на основание системы счисления) **с округлением к** $-\infty$. Например:

```
 1011 = -5          1111 = -1
>>a 1             >>a 1
 ----               ----
 1101 = -3          1111 = -1
```

Схемотехническая реализация операций сдвига очень проста. Именно поэтому эти операции рекомендуют использовать для операций умножения и деления целых чисел на числа, равные степени 2 (2, 4, 8, 16, 32, 64 и т. д.) — если, конечно, такое округление отрицательных чисел не мешает.

Арифметические сдвиги влево и вправо используются для быстрого умножения и деления на 2.

Циклический сдвиг

При циклическом сдвиге, значение последнего бита по направлению сдвига копируется в первый бит (и копируется в бит переноса).

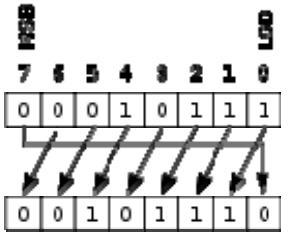
Также различают циклический сдвиг *через бит переноса* — при нём первый бит по направлению сдвига получает значение из бита переноса, а значение последнего бита сдвигается в бит переноса.

При этом сдвиге уходящий бит появляется на месте появившегося свободного на другом конце числа.

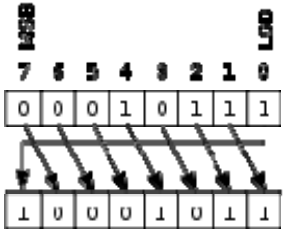
Пример работы операции сдвига:

Пусть у нас есть число 11111010b (в двоичной системе).

Если сделать сдвиг влево на 1 бит, то получим число 11110101b.
Если сделать сдвиг вправо на 1 бит, то получим число 01111101b.



Циклический сдвиг влево



Циклический сдвиг вправо

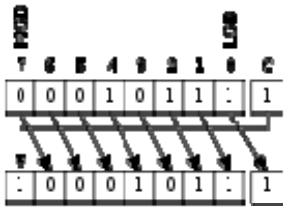
Циклический сдвиг через бит переноса

В архитектуру многих процессоров входит флаг переноса в следующий разряд (например, `cf` на x86). Данная операция выполняет циклический сдвиг над (n+1)-битным числом, состоящим из регистра и флага переноса.

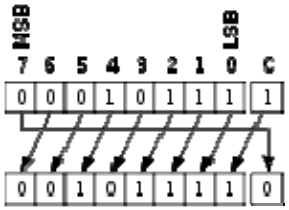
Например, если у нас в регистре число 11111010b, флаг переноса равен 0.

После сдвига влево на 1 бит в регистре 11110101b, флаг переноса равен 1.

После сдвига вправо на 1 бит в регистре 01111101b, флаг переноса равен 0.



Циклический сдвиг влево через бит переноса



Циклический сдвиг вправо через бит переноса

Операция циклического сдвига через бит переноса используется при работе с многобайтовыми числами. В частности, чтобы сдвинуть вправо на 1 бит длинное число, нужно очистить `cf` (в случае деления числа со знаком нужно записать в `cf` старший бит старшего слова) и циклически сдвинуть на единицу через `cf` каждое слово, начиная с верхнего. Например, пусть у нас есть число 011000111100b, занимающее три 4-битных слова:

```
Было:              HI=0110, MED=0011, LO=1100, cf=0
После сдвига HI:   HI=0011, MED=0011, LO=1100, cf=0
После сдвига MED:  HI=0011, MED=0001, LO=1100, cf=1
После сдвига LO:   HI=0011, MED=0001, LO=1110, cf=0
```

Сдвиги через регистр флагов более чем на 1 бит практически не используются.

В языках программирования

В следующей таблице для некоторых языков программирования приведены встроенные операторы и функции, реализующие побитовые логические операции.

Язык	НЕ	И	ИЛИ	Искл. ИЛИ	Сдвиг влево	Сдвиг вправо	Другие
C/C++, Java, C#, Ruby	~	&	|	^	<<	>>	
Pascal	not	and	or	xor	shl	shr	
PL/I	INOT ¬	IAND &	IOR |	IEOR ¬			BOOL
Prolog	\	/\	\/				

5.3. Битовая операция в теории сложности алгоритмов

Термин *битовая операция*, часто используется в области вычислений так называемых быстрых алгоритмов, которые изучают алгоритмы вычисления заданной функции с заданной точностью с использованием как можно меньшего числа битовых операций.

Битовая операция в теории алгоритмов запись знаков 0, 1, плюс, минус, скобка; сложение, вычитание и умножение двух битов (числа записаны в двоичной системе счисления). Используется для оценки сложности алгоритма.

5.4. Связь с другими науками

Битовые операции и математическая логика

Битовые операции очень близки (хотя и не тождественны) *логическим связкам* в классической логике. Бит можно рассматривать как логическое суждение — его значениями являются 1 «истина» и 0 «ложь». При такой интерпретации известные в логике связки конъюнкции, дизъюнкции, импликации, отрицания и другие имеют

представление на языке битов. И наоборот, битовые операции легко описываются на языке исчисления высказываний.

Однако, связкам математической логики более соответствуют логические операции в том числе в программировании, нежели собственно битовые операции.

Обобщение операций на булеву алгебру

Вместо одиночных битов мы можем рассмотреть векторы из фиксированного количества битов (в программировании их называют регистрами), например, байты. В программировании регистры рассматривают как двоичное разложение целого числа: $b=b_0+2b_1+2^2b_2+....+2^{N-1}b_{N-1}$, где N — количество битов в регистре.

Тем не менее, ничто не мешает рассматривать эти регистры именно как битовые векторы и проводить булевые операции покомпонентно (бит номер k значения есть результат операции от битов номер k аргументов). Кстати, математически говоря, булевы операции распространяются таким образом на произвольную булеву алгебру. Таким образом мы получаем операции *побитового* И, ИЛИ, НЕ, искл. ИЛИ и т. д. Как арифметические, данные операции не обладают хорошими свойствами за исключением побитового НЕ, которое для чисел в дополнительном коде совпадает с вычитанием из −1 (`~x == -1-x`). Однако, они очень полезны в программировании.

2-адическая интерпретация

Целое число, записанное (в дополнительном коде) в бесконечный (в сторону положительных степеней двойки) двоичный регистр является

естественным объектом для теории р-адических чисел при . Множество целых 2-адических чисел (то есть произвольных бесконечных битовых последовательностей) может быть рассмотрено как булева алгебра точно так же как и множество значений битового регистра конечной длины. Все вышеперечисленные битовые операции оказываются непрерывными отображениями. Хотя практическое программирование не располагает регистрами бесконечной длины, это

не мешает использовать данный теоретический факт в криптографии для создания быстродействующих алгоритмов шифрования.

Битовые операции как основа цифровой техники

Битовые операции лежат в основе обработки цифровых сигналов. А именно, посредством них мы можем из одного или нескольких сигналов на входе получить новый сигнал, который в свою очередь может быть подан на вход одной или нескольким таким операциям. По сути, именно битовые операции в сочетании с запоминающими элементами (напр. триггерами) реализуют всё богатство возможностей современной цифровой техники.

5.5. Практические применения

С точки зрения применения отдельная битовая операция мало интересна. Поэтому практическое применение основывается на способах комбинирования различных битовых операций, для реализации более сложного вычисления. Можно отметить два аспекта:

1. увеличение размера регистров, в которых битовые операции выполняются не по одной, а сразу на множестве 8, 16, 32, 64 битах
2. экспериментальные устройства, где обобщают битовые операции с двоичной системы, на троичные и прочие системы счисления (так например, разработана теория работы с четверичной системой (ДНК-компьютер), так же делаются исследования в области квантового компьютера).

ДНК-компьютер

ДНК-компьютер — вычислительная система, использующая вычислительные возможности молекул ДНК.

В 1994 году Леонард Адлеман, профессор университета Южной Калифорнии, продемонстрировал, что с помощью пробирки с ДНК можно весьма эффективно решать классическую комбинаторную «задачу о коммивояжере» (кратчайший маршрут обхода вершин

графа). Классические компьютерные архитектуры требуют множества вычислений с опробованием каждого варианта.

Метод ДНК позволяет сразу сгенерировать все возможные варианты решений с помощью известных биохимических реакций. Затем возможно быстро отфильтровать именно ту молекулу-нить, в которой закодирован нужный ответ.

Проблемы, возникающие при этом:

1. Требуется чрезвычайно трудоёмкая серия реакций, проводимых под тщательным наблюдением.
2. Существует проблема масштабирования задачи.

Биокомпьютер Адлемана отыскивал оптимальный маршрут обхода для 7 вершин графа. Но чем больше вершин графа, тем больше биокомпьютеру требуется ДНК-материала.

Было подсчитано, что при масштабировании методики Адлемана для решения задачи обхода не 7 пунктов, а около 200, масса количества ДНК, необходимого для представления всех возможных решений превысит массу нашей планеты.

В 2002 году исследователи из Института Вейцмана в Реховоте, Израиль, представили программируемую молекулярную вычислительную машину, состоящую из ферментов и молекул ДНК. 28 апреля 2004 года, Эхуд Шапиро, Яаков Бененсона, Биньямин Гил, Ури Бен-Дор и Ривка Адар из Института Вейцмана сообщили в журнале «Nature» о создании ДНК-компьютера с модулем ввода-вывода данных.

В январе 2013 года исследователи смогли записать в ДНК-коде несколько фотографий JPEG, набор шекспировских сонетов и звуковой файл. В марте 2013 года исследователи создали транскриптор (биологический транзистор).

Принцип работы

Нити ДНК имеют в своём составе четыре азотистых основания: цитозин, гуанин, аденин, тимин. Их последовательность кодирует

информацию. С помощью ферментов эту информацию можно изменять: полимеразы достраивают цепочки ДНК, а нуклеазы их разрезают и укорачивают. Некоторые ферменты способны разрезать и соединять цепи ДНК в местах, указываемых другими ферментами — лигазами. Таким образом, ДНК-компьютеры могут хранить и обрабатывать информацию. Также, химические реакции на разных частях молекул проходят независимо, параллельно, что обеспечивает высокую скорость вычислений.

Конечный биоавтомат Бененсона-Шапиро

Конечный биоавтомат Бененсона-Шапиро — технология многоцелевого ДНК-компьютера, разрабатываемая израильским профессором Эхудом Шапиро (en:Ehud Shapiro) и Яаковом Бененсоном из Вейцмановского института.

Его основой являются уже известные свойства биомолекул, таких как ДНК и ферменты. Функционирование ДНК-компьютера сходно с функционированием теоретического устройства, известного в математике как «конечный автомат» или машина Тьюринга.

Квантовый компьютер

Квантовый компьютер — вычислительное устройство, которое использует явления квантовой суперпозиции и квантовой запутанности для передачи и обработки данных. Хотя появление транзисторов, классических компьютеров и множества других электронных устройств связано с развитием квантовой механики и физики конденсированного состояния, информация между элементами таких систем обычно передается в виде классических величин, обычно, электрического напряжения.

Полноценный универсальный квантовый компьютер является пока гипотетическим устройством, сама возможность построения которого связана с серьёзным развитием квантовой теории в области многих частиц и сложных экспериментов; разработки в данной области связаны с новейшими открытиями и достижениями современной физики. На настоящий момент были практически реализованы лишь

единичные экспериментальные системы, исполняющие фиксированный алгоритм небольшой сложности.

Первым практическим высокоуровневым языком программирования для такого вида компьютеров считается язык Quipper, основанный на Haskell.

Идея о квантовых вычислениях была высказана Юрием Маниным в 1980 году.

Одна из первых моделей квантового компьютера была предложена Ричардом Фейнманом в 1981 году. Вскоре Пол Бениофф описал теоретические основы построения такого компьютера.

Так же концепцию квантового компьютера в 1983 предлагал Стивен Визнер в статье, которую он пытался опубликовать в течение более десяти лет до этого.

Необходимость в квантовом компьютере возникает тогда, когда мы пытаемся исследовать методами физики сложные многочастичные системы, подобные биологическим. Пространство квантовых состояний таких систем растет как экспонента от числа *n составляющих* их реальных частиц, что делает невозможным моделирование их поведения на классических компьютерах уже для *n*=10. Поэтому Визнер и Фейнман высказали идею построения квантового компьютера.

Квантовый компьютер использует для вычисления не обычные (классические) алгоритмы, а процессы квантовой природы, так называемые квантовые алгоритмы, использующие квантовомеханические эффекты, — такие как квантовый параллелизм и квантовая запутанность.

Если классический процессор в каждый момент может находиться ровно в одном из состояний |0>, |1>, ,…, |N-1> (обозначения Дирака), то квантовый процессор в каждый момент находится одновременно во всех этих базисных состояниях, при этом в каждом состоянии $|j\rangle$ — со своей комплексной амплитудой λ_l. Это квантовое состояние называется «квантовой суперпозицией» данных классических состояний и обозначается как

$$|\Psi\rangle = \sum_{j=0}^{N-1} \lambda_j |j\rangle$$

Базисные состояния могут иметь и более сложный вид. Тогда квантовую суперпозицию можно проиллюстрировать, например, так: «Вообразите атом, который мог бы подвергнуться радиоактивному распаду в определённый промежуток времени. Или не подвергнуться. Мы можем ожидать, что у этого атома есть только два возможных состояния: „распад“ и „не распад“, <…> но в квантовой механике у атома может быть некое объединённое состояние — „распада — не распада“, то есть ни то, ни другое, а как бы между. Вот это состояние и называется „суперпозицией“».

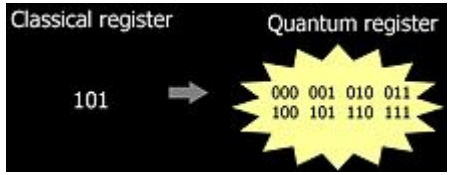
Квантовое состояние $|\Psi\rangle$ может изменяться во времени двумя принципиально различными путями:

1. Унитарная квантовая операция (квантовый вентиль, англ. *quantum gate*), в дальнейшем просто операция.
2. Измерение (наблюдение).

Если классические состояния $|j\rangle$ есть пространственные положения группы электронов в квантовых точках, управляемых внешним полем V, то унитарная операция есть решение уравнения Шрёдингера для этого потенциала.

Измерение есть случайная величина, принимающая значения $|j\rangle$, j=0, 1, …,N-1 с вероятностями $|\lambda_j|^2$ соответственно. В этом состоит квантовомеханическое правило Борна. Измерение есть единственная возможность получения информации о квантовом состоянии, так как значения λ_j нам непосредственно недоступны. Измерение квантового состояния не может быть сведено к унитарной шрёдингеровской эволюции, так как, в отличие от последней, оно необратимо. При измерении происходит так называемый коллапс волновой функции $|\Psi\rangle$, физическая природа которого до конца не ясна. Спонтанные вредоносные измерения состояния в ходе вычисления ведут к декогерентности, то есть отклонению от унитарной эволюции, что является главным препятствием при построении квантового компьютера (см. физические реализации квантовых компьютеров).

Квантовое вычисление есть контролируемая классическим управляющим компьютером последовательность унитарных операций простого вида (над одним, двумя или тремя кубитами).



3 кубита квантового регистра против 3 битов обычного

В конце вычисления состояние квантового процессора измеряется, что и даёт искомый результат вычисления.

Содержание понятия «квантовый параллелизм» в вычислении может быть раскрыто так: «Данные в процессе вычислений представляют собой квантовую информацию, которая по окончании процесса преобразуется в классическую путём измерения конечного состояния квантового регистра. Выигрыш в квантовых алгоритмах достигается за счёт того, что при применении одной квантовой операции большое число коэффициентов суперпозиции квантовых состояний, которые в виртуальной форме содержат классическую информацию, преобразуется одновременно».

Кубиты

Идея квантовых вычислений состоит в том, что квантовая система из L двухуровневых квантовых элементов (квантовых битов, кубитов) имеет 2^L линейно независимых состояний, а значит, вследствие принципа квантовой суперпозиции, пространство состояний такого квантового регистра является 2^L-мерным гильбертовым пространством. Операция в квантовых вычислениях соответствует повороту вектора состояния регистра в этом пространстве. Таким образом, квантовое вычислительное устройство размером L кубит фактически задействует одновременно 2^L классических состояний.

Физическими системами, реализующими кубиты, могут быть любые объекты, имеющие два квантовых состояния: поляризационные

состояния фотонов, электронные состояния изолированных атомов или ионов, спиновые состояния ядер атомов, и т. д.

Один классический бит может находиться в одном и только в одном из состояний $|0\rangle$ или $|1\rangle$. Квантовый бит, называемый кубитом, находится в состоянии $|\psi\rangle = a|0\rangle + b|1\rangle$, так что $|a|^2$ и $|b|^2$ — вероятности получить 0 или 1 соответственно при измерении этого состояния: $a, b \in \mathbf{C}$; $|a|^2 + |b|^2 = 1$. Сразу после измерения кубит переходит в базовое квантовое состояние, соответствующее классическому результату.

Пример:

Имеется кубит в квантовом состоянии $\frac{4}{5}|0\rangle - \frac{3}{5}|1\rangle$

В этом случае, вероятность получить при измерении
0 составляет $(4/5)^2=16/25 = 0{,}64$,
1 $(-3/5)^2=9/25 = 0{,}36$.

В данном случае, при измерении мы получили 0 с 0,64 вероятностью.

В результате измерения кубит переходит в новое квантовое состояние $|0\rangle$, то есть, при следующем измерении этого кубита мы получим 0 со стопроцентной вероятностью (предполагается, что по умолчанию унитарная операция тождественна; в реальных системах это не всегда так).

Приведем для объяснения два примера из квантовой механики: 1) фотон находится в состоянии $|\psi\rangle$ суперпозиции двух поляризаций. Это состояние есть вектор в двумерной плоскости, систему координат в которой можно представлять как две перпендикулярные оси, так что a и b *есть* проекции $|\psi\rangle$ на эти оси; измерение раз и навсегда коллапсирует состояние фотона в одно из состояний $|0\rangle$ или $|1\rangle$, причём вероятность коллапса равна квадрату соответствующей проекции. Полная вероятность получается по теореме Пифагора.

Перейдем к системе из двух кубитов. Измерение каждого из них может дать 0 или 1. Поэтому у системы есть 4 классических состояния: 00, 01, 10 и 11. Аналогичные им базовые квантовые состояния: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. И наконец, общее квантовое состояние системы имеет вид $|\Psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$. Теперь $|a|^2$ — вероятность измерить 00 и т. д. Отметим, что $|a|^2+|b|^2+|c|^2+|d|^2=1$ как полная вероятность.

Если мы измерим только первый кубит квантовой системы, находящейся в состоянии $|\Psi\rangle$, у нас получится:

1. С вероятностью $p_0 = |a|^2 + |b|^2$ первый кубит перейдет в состояние $|0\rangle$, а второй — в состояние
$$\frac{1}{\sqrt{|a|^2+|b|^2}}(a|0\rangle + b|1\rangle),$$
2. С вероятностью $p_1 = |c|^2 + |d|^2$ первый кубит перейдет в состояние $|1\rangle$, а второй — в состояние
$$\frac{1}{\sqrt{|c|^2+|d|^2}}(c|0\rangle + d|1\rangle).$$

В первом случае измерение даст состояние
$|\Psi_0\rangle = |0\rangle \oplus \frac{1}{\sqrt{|a|^2+|b|^2}}(a|0\rangle + b|1\rangle)$, во втором — состояние
$|\Psi_1\rangle = |1\rangle \oplus \frac{1}{\sqrt{|c|^2+|d|^2}}(c|0\rangle + d|1\rangle)$

Мы снова видим, что результат такого измерения невозможно записать как вектор в гильбертовом пространстве состояний. Такое состояние, в котором участвует наше незнание о том, какой же результат получится на первом кубите, называют смешанным состоянием. В нашем случае такое смешанное состояние называют проекцией исходного состояния $|\Psi\rangle$ на второй кубит, и записывают в виде матрицы плотности вида

$$\rho_2 = p_0\rho_{\Psi_0} + p_1\rho_{\Psi_1}$$

где матрица плотности состояния $|\psi\rangle$ определяется как $|\psi\rangle\langle\psi|$.

В общем случае системы из L кубитов, у неё 2^L классических состояний (00000…(L-нулей), …00001(L-цифр), … , 11111…(L-единиц)), каждое из которых может быть измерено с вероятностями 0—1.

Таким образом, одна операция над группой кубитов вычисляется сразу над всеми возможными ее значениями, в отличие от группы классических битов, когда может быть использовано лишь одно текущее значение. Это и обеспечивает беспрецедентный параллелизм вычислений.

Вычисление

Упрощённая схема вычисления на квантовом компьютере выглядит так: берётся система кубитов, на которой записывается начальное состояние. Затем состояние системы или её подсистем изменяется посредством унитарных преобразований, выполняющих те или иные логические операции. В конце измеряется значение, и это результат работы компьютера. Роль проводов классического компьютера играют кубиты, а роль логических блоков классического компьютера играют унитарные преобразования. Такая концепция квантового процессора и квантовых логических вентилей была предложена в 1989 году Дэвидом Дойчем. Также Дэвид Дойч в 1995 году нашёл универсальный логический блок, с помощью которого можно выполнять любые квантовые вычисления.

Оказывается, что для построения любого вычисления достаточно двух базовых операций. Квантовая система даёт результат, только с некоторой вероятностью являющийся правильным. Но за счёт небольшого увеличения операций в алгоритме можно сколь угодно приблизить вероятность получения правильного результата к единице.

С помощью базовых квантовых операций можно симулировать работу обычных логических элементов, из которых сделаны обычные

компьютеры. Поэтому любую задачу, которая решена сейчас, квантовый компьютер решит, и почти за такое же время. Следовательно, новая схема вычислений будет не слабее нынешней.

Большая часть современных ЭВМ работают по такой же схеме: n бит памяти хранят состояние и каждый такт времени изменяются процессором. В квантовом случае система из n кубитов находится в состоянии, являющемся суперпозицией всех базовых состояний, поэтому изменение системы касается *всех* 2^n базовых состояний одновременно. Теоретически новая схема может работать намного (в экспоненциальное число раз) быстрее классической. Практически (квантовый) алгоритм Гровера поиска в базе данных показывает квадратичный прирост мощности против классических алгоритмов

Квантовые алгоритмы

- Алгоритм Гровера позволяет найти решение уравнения $f(x)=1$, $0 \le x < N$ *за* время $O(\sqrt{N})$.
- Алгоритм Шора позволяет разложить натуральное число *n* на простые множители за полиномиальное от *log(n)* время.
- Алгоритм Залки — Визнера позволяет моделировать унитарную эволюцию квантовой системы *n частиц* за почти линейное время с использованием *O(n) кубит*.
- Алгоритм Дойча — Йожи позволяет «за одно вычисление» определить, является ли функция двоичной переменной *f(n)* постоянной ($f_1(n) = 0, f_2(n) = 1$ независимо от *n*) или «сбалансированной» ($f_3(0) = 0, f_3(1) = 1; f_4(0) = 1, f_4(1) = 0$).
- Алгоритм Саймона решает проблему чёрного ящика экспоненциально быстрее, чем любой классический алгоритм, включая вероятностные алгоритмы.

Было показано, что не для всякого алгоритма возможно «квантовое ускорение». Более того, возможность получения квантового ускорения для произвольного классического алгоритма является большой редкостью.

Пример реализации операции CNOT на зарядовых состояниях электрона в квантовых точках

Один кубит можно представить в виде электрона в двухъямном потенциале, так что $|0\rangle$ означает нахождение его в левой яме, а $|1\rangle$ — в правой. Это называется кубит на зарядовых состояниях. Общий вид квантового состояния такого электрона: $|\Psi\rangle = \lambda_0|0\rangle + \lambda_1|1\rangle$. Зависимость его от времени есть зависимость от времени амплитуд λ_0 ,λ_1; она задаётся уравнением Шредингера вида $ih\frac{\partial\Psi}{\partial t} = H\Psi$ где гамильтониан *H* имеет в силу одинакового вида ям и эрмитовости вид $\begin{pmatrix} a & -a \\ -a & a \end{pmatrix}$ для некоторой константы *a*, так что вектор $|\bar{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ есть собственный вектор этого гамильтониана с собственным значением 0 (так называемое основное состояние), а $|\bar{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ — собственный вектор со значением *2a* (первое возбуждённое состояние). Никаких других собственных состояний (с определённым значением энергии) здесь нет, так как наша задача двумерная. Поскольку каждое состояние $|\Psi\rangle$ переходит за время *t* в состояние

$\lambda_0 exp(0t)\,|0\rangle + \lambda_1 exp(-2at/h)\,|1\rangle$, то для реализации операции NOT (перехода $|0\rangle \to |1\rangle$ и наоборот достаточно просто подождать время $t=\pi h/2a$. То есть гейт NOT даётся просто естественной квантовой эволюцией нашего кубита при условии, что внешний потенциал задаёт двухъямную структуру; это делается с помощью технологии квантовых точек.

Для реализации CNOT надо расположить два кубита (то есть две пары ям) перпендикулярно друг другу, и в каждой из них расположить по отдельному электрону. Тогда константа *a для* первой (управляемой) пары ям будет зависеть от того, в каком состоянии находится электрон во второй (управляющей) паре ям: если ближе к первой, *a* будет больше, если дальше — меньше. Поэтому состояние электрона во второй паре определяет время совершения NOT в первой яме, что позволяет снова выбрать нужную длительность времени для производства операции CNOT.

Эта схема очень приблизительная и идеализирована; реальные схемы сложнее и их реализация представляет вызов экспериментальной физике.

Квантовая телепортация

Алгоритм телепортации реализует точный перенос состояния одного кубита (или системы) на другой. В простейшей схеме используются 3 кубита: телепортируемый кубит и запутанная пара, один кубит которой находится на другой стороне. Отметим, что в результате работы алгоритма первоначальное состояние источника разрушится — это пример действия общего принципа невозможности клонирования — невозможно создать точную копию квантового состояния, не разрушив оригинал. Не получится скопировать *произвольное* состояние, и телепортация — замена этой операции.

Телепортация позволяет передавать квантовое состояние системы с помощью обычных классических каналов связи. Таким образом, можно, в частности, получить связанное состояние системы, состоящей из подсистем, удалённых на большое расстояние.

Применение

Специфика применения

Основные проблемы, связанные с созданием и применением квантовых компьютеров:

- необходимо обеспечить высокую точность измерений;
- внешние воздействия могут разрушить квантовую систему или внести в неё искажения.

Приложения к криптографии

Благодаря огромной скорости разложения на простые множители, квантовый компьютер позволит расшифровывать сообщения, зашифрованные асимметричным криптографическим алгоритмом RSA. До сих пор этот алгоритм считается сравнительно надёжным, так как

эффективный способ разложения чисел на простые множители для классического компьютера в настоящее время неизвестен. Для того, например, чтобы получить доступ к кредитной карте, нужно разложить на два простых множителя число длиной в сотни цифр. Даже для самых быстрых современных компьютеров выполнение этой задачи заняло бы в сотни раз больше времени, чем возраст Вселенной. Благодаря алгоритму Шора эта задача становится вполне осуществимой, если квантовый компьютер будет построен.

Применение идей квантовой механики уже открыли новую эпоху в области криптографии, так как методы квантовой криптографии открывают новые возможности в области передачи сообщений. Прототипы систем подобного рода находятся на стадии разработки.

Физические реализации квантовых компьютеров

Построение квантового компьютера в виде реального физического прибора является фундаментальной задачей физики XXI века. По состоянию на начало 2010-х годов построены только ограниченные его варианты (самые большие сконструированные квантовые регистры имеют немногим более десятка связанных кубит). Вопрос о том, до какой степени возможно масштабирование такого устройства (так называемая «Проблема масштабирования»), является предметом новой интенсивно развивающейся области — *многочастичной квантовой механики*. Центральным здесь является вопрос о природе декогерентности (точнее, о коллапсе волновой функции), который пока остаётся открытым. Различные трактовки этого процесса можно найти в книгах.

Главные технологии для квантового компьютера:

1. Твердотельные квантовые точки на полупроводниках: в качестве логических кубитов используются либо зарядовые состояния (нахождение или отсутствие электрона в определённой точке) либо направление электронного и/или ядерного спина в данной квантовой точке. Управление через внешние потенциалы или лазерным импульсом.
2. Сверхпроводящие элементы (джозефсоновские переходы, СКВИДы и др.). В качестве логических кубитов используются

присутствие/отсутствие куперовской пары в определённой пространственной области. Управление: внешний потенциал/магнитный поток.

3. Ионы в вакуумных ловушках Пауля (или атомы в оптических ловушках). В качестве логических кубитов используются основное/возбуждённое состояния внешнего электрона в ионе. Управление: классические лазерные импульсы вдоль оси ловушки или направленные на индивидуальные ионы + колебательные моды ионного ансамбля.
4. Смешанные технологии: использование заранее приготовленных запутанных состояний фотонов для управления атомными ансамблями или как элементы управления классическими вычислительными сетями.

На рубеже XXI века во многих научных лабораториях были созданы однокубитные квантовые процессоры (по существу, управляемые двухуровневые системы, о которых можно было предполагать возможность масштабирования на много кубитов).

В конце 2001 года IBM заявила об успешном тестировании 7-кубитного квантового компьютера, реализованного с помощью ЯМР. На нём был исполнен алгоритм Шора и были найдены сомножители числа 15.

В 2005 году группой Ю. Пашкина (кандидат физ.-мат. наук, старший научный сотрудник лаборатории сверхпроводимости г. Москвы) при помощи японских специалистов был построен двухкубитный квантовый процессор на сверхпроводящих элементах.

В ноябре 2009 года физикам из Национального института стандартов и технологий в США впервые удалось собрать программируемый квантовый компьютер, состоящий из двух кубит.

В феврале 2012 года компания IBM сообщила о достижении значительного прогресса в физической реализации квантовых вычислений с использованием сверхпроводящих кубитов, которые, по мнению компании, позволят начать работы по созданию квантового компьютера.

В апреле 2012 года группе исследователей из Южно-Калифорнийского университета, Технологического университета Дельфта, университета штата Айова, и Калифорнийского университета, Санта-Барбара, удалось построить двухкубитный квантовый компьютер на кристалле алмаза с примесями. Компьютер функционирует при комнатной температуре и теоретически является масштабируемым. В качестве двух логических кубитов использовались направления спина электрона и ядра азота соответственно. Для обеспечения защиты от влияния декогерентности была разработана целая система, которая формировала импульс микроволнового излучения определённой длительности и формы. При помощи этого компьютера реализован алгоритм Гровера для четырёх вариантов перебора, что позволило получить правильный ответ с первой попытки в 95 % случаев.

Адиабатические компьютеры D-Wave

Канадская компания D-Wave Systems (англ.)русск. с 2007 года заявляла о создании различных вариантов квантового компьютера: 16 кубит — *Orion*, 28 кубит в ноябре 2007, D-Wave One (англ.)русск. с 128-кубитным чипом в мае 2011, процессор Vesuvius на 512 кубитов в конце 2012 года, более 1000 кубит в июне 2015. Компания получала инвестиции из множества источников, например 17 млн долларов США в январе 2008 года, также проводились распределённые вычисления AQUA@home (**A**diabatic **QU**antum **A**lgorithms) для тестирования алгоритмов оптимизации для адиабатических сверхпроводящих квантовых компьютеров *D-Wave*.

Компьютеры D-Wave работают на принципе квантовой релаксации (квантовый отжиг), могут решать крайне ограниченный подкласс задач оптимизации, и не подходят для реализации традиционных квантовых алгоритмов и квантовых вентилей (Quantum Annealing).

D-Wave демонстрировала решение на своих компьютерах некоторых задач, например, распознавания образов (8 декабря 2009 года на конференции NIPS (*англ.*) при участии Hartmut Neven (*англ.*), исследования трехмерной формы белка по известной последовательности аминокислот (август 2012).

Рабочая температура сверхпроводниковых чипов в аппаратах D-Wave составляет около 20 мкК, имеется тщательное экранирование от внешних электрических и магнитных полей.

С 20 мая 2011 года *D-Wave Systems* продаёт за 11 млн долларов квантовый компьютер *D-Wave One* (128 кубит), который решает только одну задачу — дискретную оптимизацию. Среди заказчиков D-Wave — Lockheed Martin (с мая 2011 года), контракт касается выполнения сложных расчетов на квантовых процессорах и включает в себя техническое обслуживание квантового компьютера *D-Wave One*.

В то же время, квантовые компьютеры *D-Wave Systems* подвергаются критике со стороны некоторых исследователей. Так, доцент (*associate professor*) Массачусетского Технологического Института Скотт Ааронсон считает, что *D-Wave* пока не смогла доказать ни того, что её компьютер решает какие-либо задачи быстрее, чем обычный компьютер, ни того, что используемые 128 кубитов удается ввести в состоянии квантовой запутанности. Если же кубиты не находятся в запутанном состоянии, то это не квантовый компьютер.

В мае 2013 года профессор *Amherst College* из канадской провинции Новая Шотландия Катерина МакГью (*Catherine McGeoch*) объявила о своих результатах сравнения компьютера *D-Wave One* на процессоре *Vesuvius* с традиционным компьютером с микропроцессором *Intel*. В первом тесте одну из задач класса *QUBO*, хорошо подходящую для структуры процессора, компьютер *D-Wave One* выполнил за 0,5 секунды, в то время как компьютеру с процессором Intel потребовалось 30 мин (выигрыш по скорости 3600 раз). Во втором тесте требовалась специальная программа для «перевода» задачи на язык компьютера *D-Wave* и скорость вычислений двух компьютеров была примерно равной. В третьем тесте, в котором также требовалась программа «перевода», компьютер *D-Wave One* за 30 минут нашёл решение 28 из 33 заданных задач, в то время как компьютер на процессоре Intel нашёл решение только для 9 задач.

В январе 2014 года учёные D-Wave опубликовали статью, в которой сообщается, что с помощью метода кубитовой туннельной спектроскопии ими было доказано наличие квантовой когерентности и запутанности между отдельными подгруппами кубитов (размером 2 и 8 элементов) в процессоре во время проведения вычислений. В декабре 2015 года специалисты компании Google подтвердили, что согласно их

исследованию компьютер D-Wave использует квантовые эффекты. При этом в «1000-кубитном» компьютере кубиты в действительности организованы в кластеры по 8 кубит каждый. Тем не менее, это позволило добиться быстродействия в 100 млн раз больше (по сравнению с обычным компьютером) в одном из алгоритмов.

Физическая реализация битовых операций

Реализация битовых операций может в принципе быть любой: механической (в том числе гидравлической и пневматической), химической, тепловой, электрической, магнитной и электромагнитной (диапазоны — ИК, видимый оптический, УФ и далее по убыванию длин волн), а также в виде комбинаций, например, электромеханической.

В первой половине XX века до изобретения транзисторов применяли электромеханические реле и электронные лампы.

В пожароопасных и взрывоопасных условиях до сих пор применяют пневматические логические устройства (пневмоника).

Наиболее распространены электронные реализации битовых операций при помощи транзисторов, например резисторно-транзисторная логика (РТЛ), диодно-транзисторная логика (ДТЛ), эмиттерно-связанная логика (ЭСЛ), транзисторно-транзисторная логика (ТТЛ), N-МОП-логика, КМОП-логика и др.

В квантовых вычислениях из перечисленных булевых операций реализуются только НЕ и искл. ИЛИ (с некоторыми оговорками). Квантовых аналогов И, ИЛИ и т. д. не существует.

Схемы аппаратной логики

Результат операции ИЛИ-НЕ или ИЛИ от всех битов двоичного регистра проверяет, равно ли значение регистра нулю; то же самое, взятое от выхода искл. ИЛИ двух регистров, проверяет равенство их значений между собой.

Битовые операции применяются в знакогенераторах и графических адаптерах; особенно велика была их роль в адаптере EGA в режимах с 16 цветами — хитроумное сочетание аппаратной логики адаптера с логическими командами центрального процессора позволяет рассматривать EGA как первый в истории графический ускоритель.

Использование в программировании

Благодаря реализации в арифметическом логическом устройстве (АЛУ) процессора многие регистровые битовые операции аппаратно доступны в языках низкого уровня. В большинстве процессоров реализованы в качестве инструкции регистровый НЕ; регистровые двухаргументные И, ИЛИ, исключающее ИЛИ; проверка равенства нулю (см. выше); три типа битовых сдвигов, а также циклические битовые сдвиги.

Регистровая операция И используется для:

- проверки бита на 0 или 1
- установки 0 в указанный бит (сброса бита)

Регистровая операция ИЛИ используется для:

- установки 1 в указанный бит

Регистровая операция исключающее ИЛИ используется для инвертирования битов регистра по маске.

Сдвиг влево/вправо используется для умножения/целочисленного деления на 2 и выделения отдельных битов.

Так, например, в сетевых интернет-технологиях операция И между значением IP-адреса и значением маски подсети используется для определения принадлежности данного адреса к подсети.

5.7. ПОЛНОТА И ЗАМКНУТОСТЬ

Ранее мы показали, что всякая булева функция с помощью операций суперпозиции может быть выражена через элементарные функции $\{x \vee y, xy, \bar{x}\}$. Поэтому для любой системы булевых функций Д возникает естественный вопрос: для всякой ли булевой функции существует равносильная ей суперпозиция функций из Д?

5.7.1. Важнейшие замкнутые классы

Система булевых функций $\mathcal{A} = \{f_1, f_2, \ldots, f_r\}$ называется **полной**, если любую булеву функцию можно представить в виде суперпозиции функций из Д.

Приведем пример полных систем.

Пример 1 Как отмечено выше, система $\mathcal{A} = \{\bar{x}, xy, x \vee y\}$ является полной.

Пример 2 Множество всех булевых функций P_2 является полной системой.

В вопросе о полноте важную роль играет следующая теорема.

Пусть $\mathcal{A}_1 = \{f_1, f_2, \ldots, f_r, \ldots\}$ *и* $\mathcal{A}_2 = \{g_1, g_2, \ldots, g_k, \ldots\}$ *системы булевых функций. Если* $Д_1$ *– полная система и каждая ее функция выражается в виде суперпозиции функций из* $Д_2$*, то система* $Д_2$ *также является полной.*

Действительно, так как $Д_1$ – полная система, то любая булева функция представима в виде суперпозиции функций из $Д_1$. А так как любая функция из $Д_1$ представима в виде суперпозиций функций из $Д_2$, то $Д_2$ – полная система.

Пример 3 Система $\{\bar{x}, x \vee y\}$ – полная.

Это следует из предыдущей теоремы и из примера 1, ибо $xy = \overline{\bar{x} \vee \bar{y}}$. Аналогичным образом получаем полноту системы $\{\bar{x}, xy\}$. Приведенные примеры говорят о том, что существуют различные полные системы булевых функций. Каждая из таких систем может быть принята в качестве набора «элементарных» функций, и любая булева функция может быть выражена в виде суперпозиции через «элементарные» функции принятого набора.

С понятием полноты тесно связано понятие замкнутого класса и замыкания.

Множество Т булевых функций называется **замкнутым классом**, если любая суперпозиция функций из Т снова принадлежит Т.

Всякая система М булевых функций порождает некоторый замкнутый класс. Этот класс состоит из всех булевых функций, которые можно получить суперпозициями из М. Такой класс называется **замыканием** М и обозначается [М]. Для замкнутого класса М следует, что [М] = М. Очевидно, что если М – полная система, что [М] = P_2.

При установлении необходимого и достаточного условия полноты важную роль играют пять замечательных классов булевых функций, которые будут рассмотрены ниже.

Первый замечательный класс – класс булевых функций, сохраняющих константу С, т.е. функций $f(x_1, x_2, \ldots, x_n)$, для которых $f(0,0,\ldots,0) = 0$.

Подсчитаем число таких булевых функций от n переменных. Поскольку на нулевом наборе значение функций из T_0 фиксировано, то в T_0 содержится ровно $\frac{1}{2}2^{2^n}$ булевых функций от n переменных.

Ясно, что функции $x, xy, x \vee y$ принадлежат классу T_0, а $\bar{x}$ не принадлежит T_0. Следовательно, $T_0 \subset P_2$.

Второй замечательный класс – класс булевых функций, сохраняющих константу 1, т.е. функций $f(x_1, x_2, \ldots, x_n)$, для которых $f(1,1,\ldots,1) = 1$.

Как и выше, легко подсчитать число таких функций от переменных. Их ровно $\frac{1}{2}2^{2^n}$.

Ясно, что функции $x, xy, x \vee y$ принадлежат классу T_1, а функция $x + y$ – нет. Следовательно, $T_1 \subset P_2$.

Третий замечательных класс – класс всех самодвойственных функций S.

Булева функция $f(x_1, x_2, \ldots, x_n)$ называется **самодвойственной**, если она совпадает со своей двойственной, т.е. $f(x_1, x_2, \ldots, x_n) = \overline{f}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$.

Пример 4 Доказать, что функция $xy \vee yz \vee xz$ является самодвойственной.

Двойственная для данной функции есть функция $\overline{\bar{x}\bar{y} \vee \bar{y}\bar{z} \vee \bar{x}\bar{z}}$. Теперь, применяя закон де Моргана, получаем:

$$\overline{\bar{x}\bar{y}} \cdot \overline{\bar{y}\bar{z}} \cdot \overline{\bar{x}\bar{z}} = (\bar{\bar{x}} \vee \bar{\bar{y}})(\bar{\bar{y}} \vee \bar{\bar{z}})(\bar{\bar{x}} \vee \bar{\bar{z}}) \cdot (x \vee y)(y \vee z)(x \vee z)$$

Используя законы дистрибутивности и идемпотентности, имеем:

$$(xy \vee xz)(x \vee z) = xy \vee yz \vee xz$$

Следовательно, искомая функция самодвойственна.

Очевидно, что x и $\bar{x}$ принадлежат классу S, а xy, $x \vee y$ не принадлежат классу S. Следовательно, $S \subset P_2$.

Подсчитаем число всех самодвойственных функций от *n* переменных.

Так как самодвойственная функция $f(x_1, x_2, \ldots, x_n)$ на наборах $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $(\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_n)$ принимает противоположные значения, то она полностью определяется своими значениями, принимаемыми ею на половине всех наборов переменных. Следовательно, число самодвойственных функций от переменных равно $2^{\frac{1}{2}2^n} = \sqrt{2^{2^n}}$.

Четвертый замечательный класс – класс всех линейных булевых функций.

Булевы функции вида $\sum_{i=1}^{n} a_i x_i + a_0$, где a_i и a_0 равны нулю или единице, называют **линейными.**

Нетрудно подсчитать число всех линейных булевых функций от переменных. Их число равно 2^{n+1}.

Очевидно, что функции $x, x + y$ принадлежат L, а функция *xy* не принадлежит L. Следовательно $L \subset P_2$.

Для того, чтобы определить, является данная булева функция линейной или нет, ее надо представить в виде полинома Жегалкина.

Пример 5 Выяснить, является ли функция $x \vee y$ линейной. Запишем данную функцию в виде полинома Жегалкина:

$$x \vee y = \overline{\bar{x}\bar{y}} = \overline{(x+1)(y+1)} = (x+1)(y+1)+1 = xy + x + y + 1 + 1 = xy + x + y$$

.

Следовательно, функция $x \vee y$ не является линейной.

Пятый замечательный класс – класс M всех монотонных булевых функций.

Два набора $\tilde{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$ называются **сравнимыми**, если $\alpha_i \le \beta_i, i = 1,2,\ldots,n$.

Запись $\tilde{\alpha} \preceq \tilde{\beta}$ означает, что набор $\tilde{\alpha}$ предшествует набору $\tilde{\beta}$. Например, $(0,1,1) \preceq (0,1,1)$, а наборы $(0,1,0)$ и $(1,0,0)$ несравнимы.

Булева функция $f(x_1, x_2, \ldots, x_n)$ называется **монотонной**, если для любых наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preceq \tilde{\beta}$, имеет место неравенство $f(\tilde{\alpha}) \le f(\tilde{\beta})$.

Очевидно, что константы 0,1 и функция *x* – монотонные функции.

Функции $\bar{x} \Rightarrow y, \bar{x}$ – монотонные функции (доказательство осуществляется проверкой).

Функции $x \Rightarrow y, \bar{x}$ не являются монотонными, так как $(0,0) \preceq (1,0)$, а $1=0 \Rightarrow 0>1 \Rightarrow 0=0, (0) \preceq (1), \bar{0} > \bar{1}$.

Следовательно, $M \subset P_2$.

Для распознавания монотонности функции полезной является следующая теорема.

Теорема 1 *Булева функция, имеющая дизъюнктивную нормальную форму, не содержащую отрицаний, является монотонной функцией, отличной от 0 и 1.*

Докажем данную теорему. Пусть булева функция $f(x_1, x_2, \ldots, x_n)$ имеет дизъюнктивную нормальную форму Д, не содержащую отрицаний, и пусть на наборе $\tilde{a} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $Д(\tilde{a}) = 1$. Тогда Д содержит конъюнкцию $X_{i_1} X_{i_2} \ldots X_{i_k}$ равную единице на наборе $\tilde{a}$. Следовательно, $\alpha_{i_1} = \alpha_{i_2} = \ldots = \alpha_{i_k} = 1$. Возьмем любой набор $\tilde{b}$ такой, что $\tilde{a} \preceq \tilde{b}$. В нем обязательно $\beta_{i_1} = \beta_{i_2} = \ldots = \beta_{i_k} = 1$. Поэтому конъюнкция $X_{i_1} X_{i_2} \ldots X_{i_k}$ при этом наборе равна 1, а значит $Д(\tilde{b}) = 1$. Итак, условие монотонности для ДНФ Д выполнено. А это значит, что функция $f(x_1, x_2, \ldots, x_n)$ монотонна.

Используя данную теорему, сразу получаем, что функции $xy \vee yz \vee xz$, $xy \vee z$ являются монотонными.

Классы булевых функций T_0, T_1, S, M, L являются замкнутыми классами.

Докажем, что T_0 – замкнутый класс. Для этого надо показать, что функция $\Phi = f(f_1, f_2, \ldots, f_k)$ принадлежит T_0, если функция f, $f_1, f_2, \ldots, f_k$ принадлежит T_0. Это следует из следующей цепочки неравенств:

$$\Phi(0,0,\ldots,0) = f(f_1(0,\ldots,0),\ldots,f_n(0,\ldots,0)) = f(0,0,\ldots,0) = 0.$$

Аналогичным образом доказывается замкнутость класса T_1.

Если $\Phi = f(f_1, f_2, \ldots, f_n), f_1, f_2, \ldots, f_k$ самодвойственные функции, то $\Phi^* = f^*(f_1^*, f_2^*, \ldots, f_n^*) = f(f_1, f_2, \ldots, f_n) = \Phi$.
Итак, S – замкнутый класс.

Пусть $\Phi = f(f_1, f_2, \ldots, f_n)$, где $f, f_1, f_2, \ldots, f_n$ – монотонные функции, и

пусть $\tilde{x} = (x_1, \ldots, x_n), \tilde{x}' = (x_{1_1}, \ldots, x_{1_n}), \ldots, \tilde{x}^k = (x_{k1}, \ldots, x_{kn})$
их наборы переменных. Здесь переменные функции Ф состоят из функций $f_1, f_2, \ldots, f_k$. Пусть $\tilde{a} = (a_1, a_2, \ldots, a_n)$ и $\tilde{b} = (b_1, b_2, \ldots, b_n)$ два таких набора, что $\tilde{a} \preceq \tilde{b}$. Каждый из наборов $\tilde{a}$ и $\tilde{b}$ однозначно определяет следующие наборы $\tilde{a}^1, \tilde{b}^1, \ldots, \tilde{a}^k, \tilde{b}^k$ значений переменных $\tilde{x}^1, \ldots, \tilde{x}^k$. Причем $\tilde{a}^1 \preceq \tilde{b}^1, \ldots, \tilde{a}^k \preceq \tilde{b}^k$. Так как $f_1, \ldots, f_n$ – монотонные функции, то $f_i(\tilde{a}^i) \preceq f_i(\tilde{b}^i)$. Следовательно, $(f_1(\tilde{a}^1), \ldots, f_k(\tilde{a}^k)) \preceq (f_1(\tilde{b}^1), \ldots, f_n(\tilde{b}^n))$. Из монотонности функции f получаем, что

$$\delta(\tilde{a}) = f(f(\tilde{a}^1),\ldots,f_k(\tilde{a}^k)) \le f(f_1(\tilde{b}^1),\ldots,f_k(\tilde{b}^k)) = \delta(\tilde{b})$$

.

Итак, M – замкнутый класс.

Замкнутость класса L следует непосредственно из определения линейных функций.

Как показано выше, функция $\bar{x}$ не принадлежит классу M. Следующая теорема показывает, что всякая немонотонная функция содержит, в некотором смысле, в своем составе функцию отрицания.

Теорема 2 *Если* $f(x_1, x_2, \ldots, x_n)$ *– немонотонная функция, то в результате ее суперпозиции с константами 0 и 1 может быть получена функция отрицания* x_i *одного из аргументов функции* $f(x_1, x_2, \ldots, x_n)$.

Докажем данную теорему. Так как функция $f(x_1, x_2, \ldots, x_n)$ немонотонная, то найдутся два набора $\tilde{a}$ и $\tilde{b}$ значений переменных таких, что $\tilde{a} \prec \tilde{b}$, и $f(\tilde{a}) > f(\tilde{b})$. Причем, в качестве этих наборов $\tilde{a}$ и $\tilde{b}$ можно выбрать соседние наборы, т.е. наборы, отличающиеся значениями только по одной из координат.

Действительно, если $\tilde{a}$ и $\tilde{b}$ не являются соседними наборами, то набор $\tilde{b}$ отличается от набора $\tilde{a}$ в t координатах, где t > 1. Причем, эти координаты в наборе $\tilde{a}$ равны 0, а в наборе $\tilde{b}$ равны 1. Из этого следует, что между наборами $\tilde{a}$ и $\tilde{b}$ можно вставить t – 1 наборов $\tilde{a}^1, \ldots, \tilde{a}^{t-1}$, таких, что

$$\tilde{a} \le \tilde{a}^1 \le \ldots \le \tilde{a}^{t-1} \le \tilde{b} \quad (5.1)$$

.

Так как $f(\tilde{a}) > f(\tilde{b})$, то обязательно найдется такая пара соседних наборов из цепочки (5.1) $\tilde{a}^i$ и $\tilde{a}^{i+1}$, что $f(\tilde{a}^i) > f(\tilde{a}^{i+1})$. Не ограничивая общности, мы можем считать, что

$\tilde{a} = (a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n)$;

$\tilde{b} = (a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n)$.

Подставим в функцию $f(x_1, x_2, \ldots, x_n)$ вместо переменной x_j константу a_i, где $j = 1, 2, \ldots, i-1, i+1, \ldots, n$. В результате мы получим функцию от одной переменной:

$$g(x_i) = f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) = f(\tilde{a}) > f(\tilde{b}) = f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n) = g(1)$$

А это значит, что $g(0) = 1, g(1) = 0$. Следовательно, $g(x_i) = \bar{x}_i$.

5.7.2.Теорема о полноте

Цель данного параграфа дать ответ на один из основных вопросов алгебры логики – вопрос о необходимом и достаточном условиях полноты системы булевых функций. Ответ на этот вопрос дает следующая теорема, доказательство которой будет вестись по методу А. В. Кузнецова и С. К. Яблонского.

Теорема 3 (о полноте) *Для того, чтобы система булевых функций Д была полной, необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов* T_0, T_1, *S, M, L.*

Доказательство. *Необходимость.* Пусть Д – полная система, целиком содержащаяся в одном из классов T_0, T_1, S, M, L. Не ограничивая

общности, будем считать, что $\mathcal{A} \subseteq T_0$. Тогда $P_2 - [\mathcal{A}] \subseteq [T_0] - T_0$. Следовательно, $T_0 - P_2$, что невозможно.

Достаточность. Пусть система булевых функций Д целиком не содержится ни в одном из классов T_0, T_1, S, M, L. Тогда в системе Д обязательно найдутся следующие функции: f_1, не сохраняющая 0; f_2, не сохраняющая 1; не самодвойственная функция f_3; нелинейная функция f_4; не монотонная функция f_5. Учитывая понятие фиктивной переменной, мы можем считать, что эти функции зависят от одних и тех же переменных.

Вначале построим из системы функций Д константы 0 и 1.

Рассмотрим функцию $g(x_1) = f(x_1, x_1, \ldots, x_1)$. Эта функция есть суперпозиция функций f_1 и x_1. Так как f_1 не принадлежит классу T_0, $g(0) = f(0,0,\ldots,0) = 1$. Если теперь g(1) = 1, то $g(x_1)$ - константа 1. Подставляя константу 1 в функцию f_2, мы получаем константу 0, ибо $f_2(1,1,\ldots,1) = 0$.

Пусть теперь $g(1) = 0$. Из равенства g(0) = 1 и g(1) = 0 заключаем, что $g(x) = \overline{x_1}$. Возьмем несамодвойственную функцию f_3. Очевидно, что в этом случае найдется такой набор переменных $(a_1, a_2, \ldots, a_n)$, что $f_3(a_1, a_2, \ldots, a_n) = f_3(\overline{a_1}, \overline{a_2}, \ldots, \overline{a_n})$.

Рассмотрим функцию $\varphi_i(x) = x^{\alpha_i}, i = 1, 2, \ldots, n$ и построим с помощью операции суперпозиции функцию $h(x) = f(\varphi_1(x), \ldots, \varphi_n(x))$. Тогда получаем:

$$h(0) = f(\varphi_1(0), \ldots, \varphi_n(0)) = f(0^{\alpha_1}, \ldots, 0^{\alpha_n}) = f(\overline{a_1}, \ldots, \overline{a_n})$$
$$= f(a_1, \ldots, a_n) = f(1^{\alpha_1}, \ldots, 1^{\alpha_n}) = h(1) \qquad (6.2)$$

Итак, h(0)=h(1). А это значит, что h(*x*) есть константа 0 или 1. Так как мы построили функцию $g(x_1) = \overline{x_1}$, то суперпозиция этой функции с одной из констант дает другую константу. Следовательно, константы 0 и 1 нами построены.

Теперь, используя предыдущую теорему, мы можем с помощью суперпозиции функции f_5 и констант 0,1 построить функцию $\overline{x_i}$, а следовательно и все функции $\overline{x_1}, \ldots, \overline{x_n}$. Ранее мы показали, что любая булева функция $f(x_1, \ldots, x_n)$ может быть представлена в виде суперпозиции уже построенных функций и функций $x_1 x_2$. Следовательно, для завершения доказательства теоремы нам осталось построить функцию $x_1 x_2$. Для этого возьмем функцию f_4 и построим для этой функции полином Жегалкина. Так как эта функция нелинейная, то в этом полиноме найдется слагаемое, содержащее не менее двух множителей. Без ограничения общности можно считать, что этими множителями являются x_1 и x_2. Тогда мы можем записать полином Жегалкина для функции f_4 в следующем виде:

$$f_4(x_1, x_2, \ldots, x_n) = x_1 x_2 h(x_3, \ldots, x_n) + x_1 h_2(x_3, \ldots, x_n) + x_2 h_3(x_3, \ldots, x_n) + h_4(x_3, \ldots, x_n)$$

В силу единственности полинома, функция $h_1(x_3, \ldots, x_n)$ не равна тождественно нулю. Выберем такие значения переменных $a_3, a_4, \ldots, a_n$, что $h(a_3, \ldots, a_n) = 1$. Ввиду этого, мы приходим к функции

$$\varphi(x_1, x_2) = x_1 x_2 + \alpha x_1 + \beta x_2 - \gamma,$$

где α, β, j константы 0 или 1. Построим функцию $\varphi(x_1, x_2)$ следующим образом:

$$\varphi(x_1, x_2) = \varphi(x_1 + \beta, x_2 + \alpha) + \alpha\beta + j =$$
$$= (x_1 + \beta)(x_2 - \beta) + \alpha(x_1 + \beta) + \beta(x_2 + \alpha) + j + \alpha\beta + j = x_1 x_2.$$

Итак, теорема о полноте полностью доказана.

В тех задачах, где требуется выяснить, является ли данная система булевых функций $\mathcal{A} = \{f_1, f_2, \ldots, f_n\}$ полной, мы будем составлять таблицы, которые называются таблицами Поста. Данные таблицы имеют следующий вид.

	T_0	T_1	S	L	M
f_1					
f_2					
…	…	…	…	…	…
f_{n-1}					
f_n					

В клетках данной таблицы мы будем писать плюс или минус, в зависимости от того, входит функция, стоящая в данной строке в класс, стоящий в данном столбце, или не входит. Используя теорему о полноте, мы получаем, что для полноты данной системы булевых функций необходимо и достаточно, чтобы в каждом столбце стоял хотя бы один минус.

Пример 6 Выяснить, являются ли следующие системы булевых функций полными.

$\mathcal{A}_1 = \{x \mid y\}$, $\mathcal{A}_2 = \{x + y + z, xy, 0, 1\}$;

$\mathcal{A}_3 = \{x \Rightarrow y, \bar{x}y\}$, $\mathcal{A}_4 = \{\overline{xy} \vee \overline{\bar{x}z} \vee \overline{yz}\}$.

Составим таблицу Поста для системы $Д_1$:

	T_0	T_1	S	L	M
$x \mid y$	-	-	-	-	-

Нетрудно заметить, что $x \mid y = \bar{x} \vee \bar{y}$.

Ясно, что $\bar{x} \vee \bar{y}$ не принадлежит T_0 и T_1. Двойственная функция к функции $\bar{x} \vee \bar{y}$ имеет вид: $\overline{\bar{x} \vee \bar{y}} = \bar{x} \cdot \bar{y}$.

Следовательно, $\bar{x} \vee \bar{y}$ не принадлежит классу S. Найдем полином Жегалкина для функции $\bar{x} \vee \bar{y}$: $\bar{x} \vee \bar{y} = \overline{xy} = xy + 1$.

Следовательно, функция $\bar{x} \vee \bar{y}$ не принадлежит классу L. Так как $(0,0) \leq (1,1)$, а $\bar{0} \vee \bar{0} > 1 \vee \bar{1}$, то $\bar{x} \vee \bar{y}$ не принадлежит классу M.

Итак, система $\mathcal{A}_1 = \{x \mid y\}$ является полной.

Составим таблицу Поста для системы $Д_2$:

	T_0	T_1	S	L	M
$x + y + z$	+	+	+	+	–
xy	+	+	–	–	+
	+	–	–	+	+
	–	+	–	+	+

Исследуем функцию $x+y+z$. Легко проверить, что она принадлежит классам T_1, T_0, L. Покажем, что она является самодвойственной.

$$\overline{\overline{x}+\overline{y}+\overline{z}}=x+1+y+1+z+1+1=x+y+z.$$

Функция $x+y+z$ не монотонна, так как $(1,0,0)\le(1,1,0)$, а $1+0+0>1+1+0$. В каждом столбце таблица Поста для системы $Д_2$ стоит минус. Следовательно, система $Д_2$ является полной.

Составим таблицу Поста для системы $Д_3$:

	T_0	T_1	S	L	M
$x\Rightarrow y$	–	+	–	–	–
$\overline{x}$	–	–	+	+	–

Из таблицы видно, что система $Д_3$ является полной.

Составим таблицу Поста для системы $Д_4$:

	T_0	T_1	S	L	M
$\overline{xy}\vee\overline{xz}\vee\overline{yz}$	-	-	+	-	-

Исследуем функцию $\overline{xy}\vee\overline{xz}\vee\overline{yz}$.

Легко проверить, что данная функция не принадлежит ни T_0, ни T_1. Так как $(0,0,0)\le(1,1,1)$, а значение функции при наборе (0,0,0) больше, чем при наборе (1,1,1), то функция $\overline{xy}\vee\overline{xz}\vee\overline{yz}$ не монотонна. Очевидно, что все переменные данной функции являются существенными. А так как данная функция не может совпадать ни с

$x+y+z$, ни с $x+y+z+1$, то она нелинейная. Как и в примере 6, можно показать, что функция $\overline{xy}\vee\overline{xz}\vee\overline{yz}$ является самодвойственной. Следовательно, система $Д_4$ не является полной.

Назовем систему булевых функций **несократимой**, если из нее нельзя исключить ни одной функции так, чтобы оставшаяся после исключения система снова была полной.

Очевидно, что любую полную систему булевых функций можно свести к несократимой. Как следует из теоремы о полноте, в любой несократимой полной системе содержится не более 5 функций. Следующая теорема показывает, что в действительности их число всегда может быть сокращено до 4.

Теорема 4 *Максимальное возможное число функций в несократимой полной системе булевых функций равно 4.*

Действительно, при доказательстве теоремы о полноте мы видели, что из любой полной системы булевых функций можно выделить полную подсистему, содержащую не более пяти функций. Причем, функция f, не сохраняющая 0, либо не сохраняет 1, либо если $f(1,1,\dots,1)=1$ является самодвойственной. Следовательно, кроме этой функции достаточно оставить в системе лишь три функции: нелинейную, немонотонную либо функцию, не сохраняющую 1, либо несамодвойственную функцию.

Следующий пример показывает, что константа 4 не может быть понижена.

Пример 7 Рассмотрим систему функций $Д=\{x_1x_2,0,1,x_1+x_2+x_3\}$.

Составим таблицу Поста для данной системы:

	T_0	T_1	S	L	M

x_1x_2	+	+	–	–	+	
	+	–	–	+	+	
	–	+	–	+	+	
$x_1+x_2+x_3$	+	+	+	+	–	

Из таблицы видно, что данная система является полной и несократимой, ибо

$$\{0,1,x_1+x_2+x_3\}\subset L;\ \{x_1x_2,1,x_1+x_2+x\}\subset T_1;$$
$$\{x_1x_2,0,x_1+x_2+x_3\}\subset T_0;$$

$$\{x_1x_2,0,1\}\subset M$$

6. КОНТАКТНЫЕ СХЕМЫ

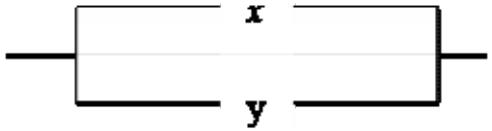
6.1. Анализ и синтез контактных схем

В начале XX века известный физик П. Эренфест впервые указал на возможность применения аппарата алгебры логики в технике. Эта идея нашла свое воплощение в работах советского физика В. И. Шестакова, американского математика К. Шеннона и японского инженера А. Какасима. Первыми объектами применения алгебры логики для решения технических задач были контактные схемы. Под контактными схемами мы будем понимать электрические цепи, содержащие только контакты. Каждый контакт может находиться в двух состояниях – разомкнут (0) и замкнут (1). Такие цепи мы будем изображать диаграммой, на которой возле контактов пишется x_i или $\overline{x_i}$. Причем значение 1 этих переменных соответствует прохождению через данный контакт, а значения 0 нет.

Если контакты *x* и y соединены последовательно, то цепь замкнута, когда оба контакта замкнуты и разомкнута, когда хотя бы один из контактов разомкнут. Ясно, что такой схеме

соответствует булева функция xy.

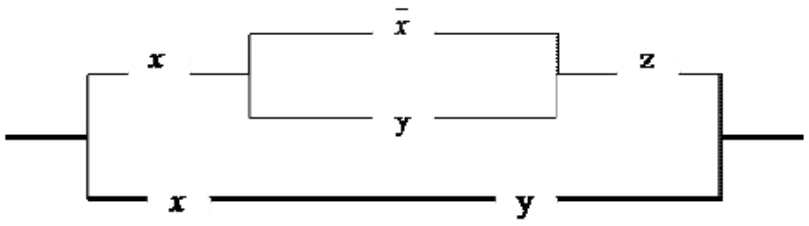
Если контакты *x* и *y* соединены параллельно, то цепь замкнута, когда хотя бы один контакт замкнут и разомкнута, когда оба контакта разомкнуты. Ясно, что такой схеме



соответствует булева функция $x \vee y$.

Указанное соответствие позволяет любую булеву функцию представить в виде контактной схемы. С другой стороны, любая контактная схема с последовательно или параллельно соединенными контактами реализуется булевой функцией. Задача анализа контактной схемы и состоит в построении соответствующей ей булевой функции.

Например, контактная схема

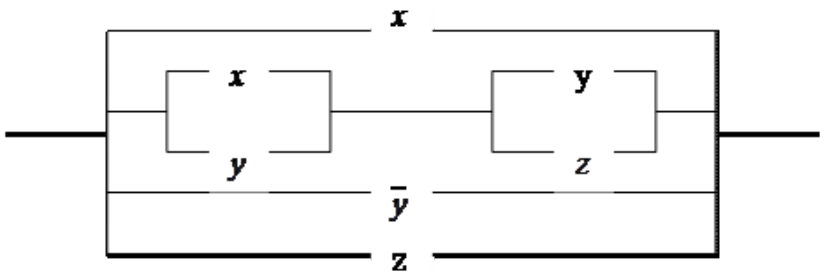


реализуется булевой функцией $x(\bar{x} \vee y)z \vee xy$.

Однако, поскольку одна и та же булева функция может быть выражена различными формулами, то ее реализация контактными схемами неоднозначна. Всегда можно построить много различных контактных схем, соответствующих данной функции. Такие схемы называют эквивалентными.

Задача синтеза контактной схемы состоит в построении контактной схемы по заданной булевой функции, которая может быть задана как формулой, так и таблицей. В обоих случаях необходимо выразить функцию через операции конъюнкции, дизъюнкции и отрицания. Каждая операция конъюнкции соответствует последовательному соединению контактов. В результате параллельного соединения получаем контактную схему. Из множества эквивалентных схем, путем упрощения формул выделяют наиболее простую схему. Центральной проблемой синтеза контактных схем является построение для данной булевой функции более простой схемы. Часто эта проблема сводится к минимизации булевых функций, т.е. к такому их представлению, в котором соответствующие формулы содержат минимальное количество вхождений переменных.

Рассмотрим схему



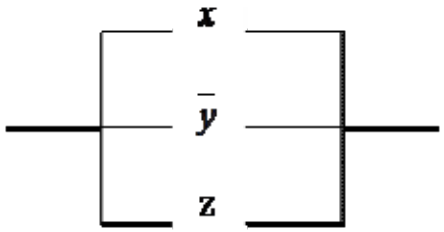
Данная схема реализуется следующей формулой:

$$x \vee (x \vee \bar{y})(y \vee \bar{z}) \vee \bar{y} \vee z$$

Упростим данную формулу. Используя закон дистрибутивности, получаем:

$$x \vee xy \vee x\bar{z} \vee y\bar{y} \vee \overline{yz} \vee \bar{y} \vee z =$$
$$x(1 \vee y \vee \bar{z}) \vee \bar{y}(y \vee \bar{y} \vee 1) \vee z = x \vee \bar{y} \vee z$$

Следовательно, данную схему можно упростить, заменив ее следующей эквивалентной схемой:



Решим теперь следующую задачу: из контактов x, y, z составить по возможности более простую схему так, чтобы она замкнулась тогда и только тогда, когда замкнуты не менее двух контактов.

Составим таблицу истинности для булевой функции, соответствующей требуемой контактной схеме

$x\ y\ z\ \ f(x,y,z)$

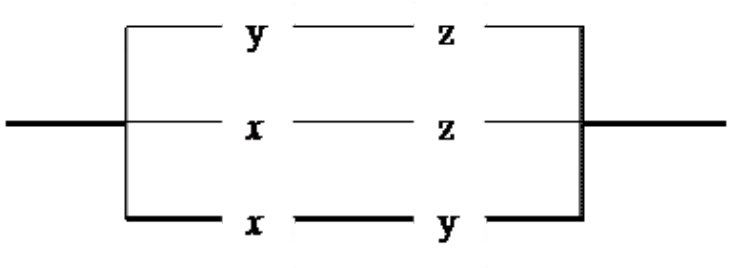
Найдем для данной булевой функции совершенную ДНФ:

$$f(x,y,z) = \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz$$

Упростим данную формулу

$$\bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz =$$
$$yz(\bar{x} \vee x) \vee xz(\bar{y} \vee y) \vee xy(\bar{z} \vee z) = yz \vee xz \vee xy$$

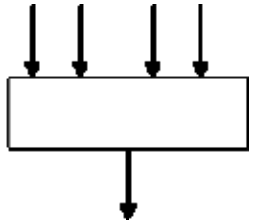
Данной формуле соответствует следующая контактная схема:



Контактные схемы исторически были первыми техническими средствами реализации булевых функций. В дальнейшем появилось много различных устройств, реализующих булевы функции одной и нескольких переменных.

Пусть имеется некоторое устройство, имеющее n упорядоченных «входов» и один «выход», причем внутренняя структура этого устройства нас не интересует. На каждый из входов могут подаваться два сигнала, которые мы будем обозначать символами 0 и 1. При

каждом наборе сигналов на входах и выходе возникает один из сигналов 0 или 1. Причем набор сигналов на входах однозначно определяет сигнал на выходе. Очевидно, что каждое такое устройство реализует булеву функцию.



6.2. Схемы со многими выходами

Если необходимо реализовать несколько булепых функций, то каждая из них может быть представлена соответствующей контактной схемой. Однако такой путь неэкономичен. Более целесообразно построить единую схему с несколькими выходами (рис. 1), соответствующими данной системе функций:

$$y_1 = f_1(x_1, \ldots, x_n);\ y_2 = f_2(x_1, \ldots, x_n);$$
$$\ldots;\ y_m = f_m(x_1, \ldots, x_n).$$

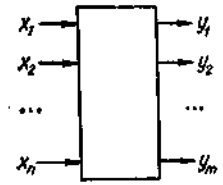


Рис. 1. Контактная схема с *n* входами и *m* выходами

Примером *многовыходной схемы* может служить *полное релейное дерево,* в котором каждая конституента единицы представлена одним выходным полюсом, а всего имеется 2^n выходов (на рис.2, *а* изображено полное релейное дерево для $n = 3$).

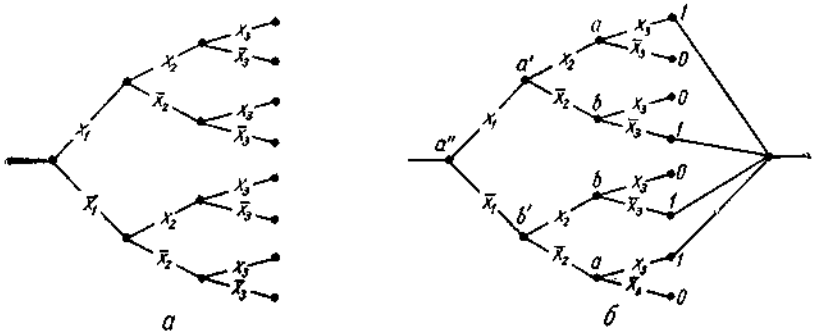


Рис. 2. Полное релейное дерево для трех переменных (*а*) и его преобразование для конкретной функции *(б).*

Любую функцию от *n* переменных можно реализовать объединением выходов полного релейного дерева, которые соответствуют тем наборам переменных, на которых функция принимает значения 1. Контакты, которые не подсоединены к требуемым выходам, удаляются из схемы. Например, для функции, заданной таблицей

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y	0	1	1	0	1	0	0	1

,

построение приведено на рис. 2, *б.*

Более простые схемы можно получить объединением участков релейного дерева, общих для путей, которые соответствуют различным конституентам. Для этого обозначаем одинаковыми буквами или цифрами те узлы, из которых выходят пары x_n и $\bar{x}_n$ с совпадающими значениями функции. Дальше аналогично обозначаем одинаковыми буквами узлы, из которых выходят пары x_{n-1} и $\bar{x}_{n-1}$ с совпадающими предыдущими обозначениями (порядок букв также учитывается) и т. д. до последней пары x_x и $\bar{x}_1$. После этого одинаково обозначенные узлы объединяются и проводятся упрощения в соответствии с рис.3.

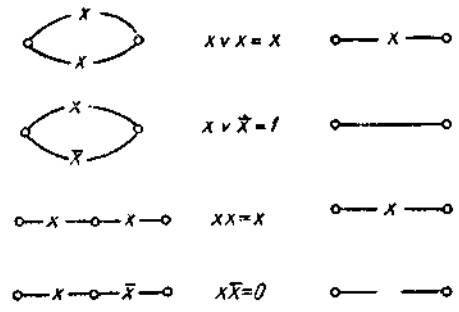


Рис.3. Упрощение контактных схем для одной переменной.

Так, в схеме рис.2, б для пар $(x_3, \bar{x}_3)$ имеется две комбинации значений (1, 0) и (0, 1). Узлы, из которых выходят пары с комбинациями (1, 0), обозначаем буквой *a,* а узлы, из которых выходят пары с комбинациями (0, 1) — буквой *b.* Для пар $(x_2, \bar{x}_2)$ также встречаются две комбинации в предыдущих обозначениях: *(a, b)* и *(b, a).* Узлы, из которых выходят эти пары, обозначаем соответственно через *a′* и *b′.* Наконец, для пары $(x_1, \bar{x}_1)$ имеется единственная комбинация (*a*', *b'),* и узел, из которого выходит эта пара, обозначаем через *a″.* Объединяя узлы с одинаковыми обозначениями *(a* и *b),* приходим к схеме, показанной на рис.4, которая после замены параллельных контактов $x_я$ и x_3 на x_3, а также $\bar{x}_3$ и $\bar{x}_3$ на x_3, совпадает с мостиковой схемой.

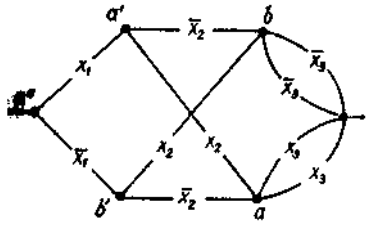


Рис. 4. Преобразование контактной схемы (рис. 2, *б)*

Объединяя выходы полного релейного дерева, можно построить контактные схемы и для нескольких функций при условии, что множества наборов значений переменных, на которых эти функции принимают значения 1, не пересекаются. Пусть, например, требуется построить контактную схему с двумя выходами, реализующую

функции $y_1 = x_1x_2 \vee \bar{x}_1\bar{x}_2x_3$ и $y_2 = x_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3$. Из таблицы соответствия для этих функций

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y_1	0	1	0	0	0	0	1	1
y_2	1	0	1	0	1	1	0	0

видим, что ни на одном наборе значений переменных функции не принимают одновременно значений, равных 1. Следовательно, для построения требуемой контактной схемы можно воспользоваться полным релейным деревом (рис. 5, *а),* в результате преобразования которого получаем схему с двумя выходами (рис. 5, *б).*

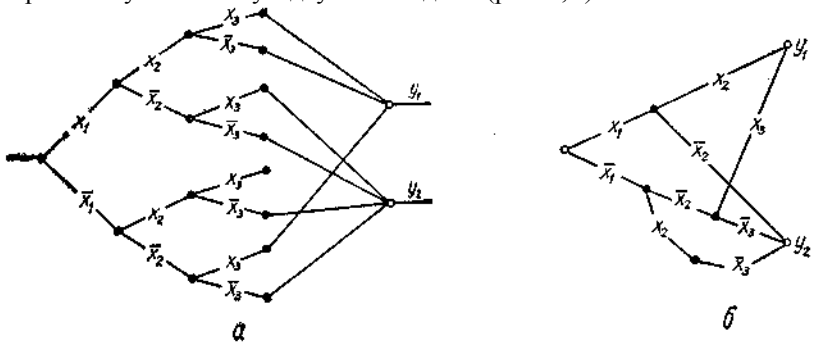


Рис. 5. Построение схемы с двумя выходами:

а— преобразование полного релейного дерева; б —контактная схема

6.3. Булевы матрицы

Для описания контактных схем произвольной структуры с любым числом выходов используются различные типы *булевых матриц,* элементами которых являются константы 0 и 1, переменные $x_1, x_2, \ldots, x_n$, и функции этих переменных.

Пусть контактная схема имеет *k* узлов. *Матрица непосредственных связей (примитивная матрица соединений) P* — это квадратная таблица $k \times k$, элементы главной диагонали которой равны 1, а элементы $p_{ij} = p_{ji}$ представляют собой булеву функцию прямого соединения между узлами *i* и *j*. *Матрица полных связей (полная матрица*

соединений) Q отличается тем, что ее элементы $q_{ij} = q_{ji}$ представляют собой булеву функцию с учетом всевозможных путей без циклов между узлами *i* и *j*. Так, для схемы рис. 6 имеем:

$$P = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & x_4 & x_2 \\ 0 & x_4 & 1 & x_3 \\ x_1 & x_2 & x_3 & 1 \end{bmatrix};$$

$$Q = \begin{bmatrix} 1 & x_1(x_2 \vee x_3x_4) & x_1(x_3 \vee x_2x_4) & x_1 \\ x_1(x_2 \vee x_3x_4) & 1 & x_4 \vee x_2x_3 & x_2 \vee x_3x_4 \\ x_1(x_3 \vee x_2x_4) & x_4 \vee x_2x_3 & 1 & x_3 \vee x_2x_4 \\ x_1 & x_2 \vee x_3x_4 & x_3 \vee x_2x_4 & 1 \end{bmatrix}.$$

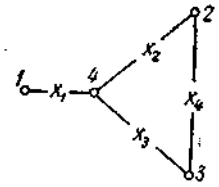


Рис. 6. К определению булевых матриц контактной схемы

Произведение булевых матриц определяется, как и для обычных матриц, правилом «строка на столбец», но операциям сложения и умножения действительных чисел соответствуют дизъюнкция и конъюнкция логических переменных и функций. Элементы матрицы $C = AB$, где *A* и B — булевы матрицы, выражаются соотношением $c_{ij} = a_{i1}b_{1j} \vee a_{i2}b_{2j} \vee \ldots \vee a_{in}b_{nj}$. Произведения матрицы самой на себя выражаются как ее степени $AA = A^2$, $A^2A = A^3, \ldots, A^{n-1}A = A^n$.

Можно показать, что для любой контактной схемы с *k* узлами существует такое $r \le k - 1$, что $P^r = P^{r+s} = Q$, где s — произвольное целое положительное число. Это значит, что матрицу полных связей можно получить умножением матрицы непосредственных связей *P* на саму себя до тех пор, пока результат не начнет повторяться, причем число таких умножений не превышает $k - 1$. Так, для рассматриваемого примера имеем:

$$P^2 = \begin{bmatrix} 1 & x_1x_2 & x_1x_3 & x_1 \\ x_1x_2 & 1 & x_4 \vee x_2x_3 & x_2 \vee x_3x_4 \\ x_1x_3 & x_4 \vee x_2x_3 & 1 & x_3 \vee x_2x_4 \\ x_1 & x_2 \vee x_3x_4 & x_3 \vee x_2x_4 & 1 \end{bmatrix}; \quad P^3 = Q.$$

Следует отметить, что элементы матрицы P^i представляют собой функции всех связей между узлами посредством не более чем $i - 1$

узлов. В частности, каждый элемент матрицы P^2 учитывает непосредственные связи между парой узлов и связи между ними посредством еще одного узла. Например, $p_{23} = p_{32} = x_4 \vee x_2x_3$

соответствует непосредственной связи между узлами 2 и 3 через контакту, а также связи посредством узла 4 (член x_2x_3).

6.4. О разложении определителей булевых матриц

В разделе указаны необходимые и достаточные условия разложимости булевых определителей по строке или столбцу квадратной матрицы с элементами из произвольной булевой алгебры. Вводится естественное разложение произвольной булевой матрицы на внутреннюю, внешнюю и детерминированную части. Введённые понятия позволяют сформулировать основной результат: разложения по строке (столбцу) детерминантов произвольной квадратной булевой матрицы выполняются тогда и только тогда, когда формулы разложения выполняются по соответствующей строке (столбцу) для детерминантов её внутренней части.

6.4.1. Введение

Определение 1.1. *Булевой алгеброй* называют систему $\langle B, \cup, \cap, ', 0, I\rangle$, где B — множество *(носитель),* $\cup\colon B \times B \to B$, $\cap\colon B \times B \to B$— две бинарные операции, $'\colon B \to B$ — унарная операция, а 0 и I — различные $(0 \neq I)$элементы из B, такие что следующие тождества выполняются для любых $x, y, z \in B$.

1.1. $x \cup y = y \cup x$.
1.2. $x \cap y = y \cap x$.
2.1. $(x \cup y) \cup z = x \cup (y \cup z)$.

2.2. $(x \cap y) \cap z = x \cap (y \cap z)$.
3.1. $x \cup (x \cap y) = x$.
3.2. $x \cap (x \cup y) = x$.
4.1. $x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$.
4.2. $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$.
5.1. $x \cap 0 = 0$.
5.2. $x \cup I = I$.
6.1. $x \cup x' = I$.
6.2. $x \cap x' = 0$.

Назовём $\cup$ *объединением,* $\cap$ *пересечением,* а $'$ *дополнением.* Таким образом, объединение и пересечение — *коммутативные* (1.1 и 1.2), *ассоциативные* (2.1 и 2.2), удовлетворяющие законам *поглощения* (3.1 и 3.2) операции. Это означает, что $\langle B, \cup, \cap\rangle$ —*решётка.* Эта решётка *дистрибутивна* (4.1 и 4.2), обладает *нулём* (5.1) и *единицей* (5.2). Более того, эта решётка является *решёткой с дополнением* (6.1 и 6.2). Очевидно, что множество (n × n)-матриц с элементами из булевой алгебры $\langle B, \cup, \cap, ', 0, I\rangle$ также образуют булеву алгебру, в которой объединение, пересечение и дополнение определяются для любых (n×n)-матриц $A = (a^i_j)$ и $B = (b^i_j)$ как

$$A \cup B = (a^i_j) \cup (b^i_j) = (a^i_j \cup b^i_j),$$
$$A \cap B = (a^i_j) \cap (b^i_j) = (a^i_j \cap b^i_j),$$
$$A' = (a^i_j)' = (a'^i_j).$$

Определение 1.2. *Ориентированные полу перманенты* $\overset{+}{\nabla}A$, $\overset{-}{\nabla}A$ (или компоненты *бидетерминанта) (n* × n)-матрицы A (n≥ 2) с элементами из произвольнойбулевой алгебры $\langle B, \cup, \cap, ', 0, I\rangle$ определяются как

$$\overset{+}{\nabla}A = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in\overset{+}{P}} (a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n}), \quad \overset{-}{\nabla}A = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in\overset{-}{P}} (a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n}),$$

где a^i_j — элементы матрицы A, а через $\overset{+}{P}$ и $\overset{-}{P}$ обозначаются чётные и нечётные n-перестановки верхних индексов соответственно. Ориентированные полуперманенты позволяют ввести *перманент*

$$\operatorname{Per} A = \overset{+}{\nabla}A \cup \overset{-}{\nabla}A$$

и *общую часть* ориентированных полуперманентов

$$\triangle A = \overset{+}{\nabla}A \cap \overset{-}{\nabla}A.$$

Правый и левый (ориентированные) определители вводятся соответственно как

$$\mathrm{RDet}\, A = \overset{+}{\nabla} A \setminus \overline{\nabla} A = \overset{+}{\nabla} A \cap \left(\overline{\nabla} A\right)' \quad \text{и} \quad \mathbf{LDet}\, A = \overline{\nabla} A \setminus \overset{+}{\nabla} A = \overline{\nabla} A \cap \left(\overset{+}{\nabla} A\right)'.$$

Определителем булевых матриц называют булеву сумму правого и левого определителя: $\mathbf{Det}\, A = \mathbf{RDet}\, A \cup \mathbf{LDet}\, A$.

Известно, что для ориентированных полуперманентов $\overset{+}{\nabla} A$, $\overline{\nabla} A$ и перманента Per A справедливы формулы Лапласа, дающие разложения по элементам строк или столбцов любой квадратной матрицы A над произвольным коммутативным полукольцом. Для правых $\mathrm{RDet}\, A$, левых $\mathbf{LDet}\, A$ ориентированных определителей и определителя $\mathbf{Det}\, A$ формулы Лапласа не выполняются в общем случае. Попытки определения условий, при которых разложения Лапласа таких определителей возможны, предпринимались и ранее. Авторы настоящего раздела указывают некоторые условия разложения определителя по строке.

В этом разделе будет показано, что формулы разложения определителей по элементам строк или столбцов булевой матрицы верны, тем не менее, для достаточно широкого класса матриц над произвольной булевой алгеброй. Сначала будет определено естественное разложение произвольной булевой матрицы на внутреннюю, внешнюю и детерминированную части, опуская исследования их свойств. Введённые понятия позволят сформулировать основной результат этого раздела о разложении определителей по строке или столбцу (теорема 6.8.1): разложения по строке (столбцу) ориентированных детерминантов для произвольной квадратной булевой матрицы выполняются тогда и только тогда, когда формулы разложения выполняются по соответствующей строке (столбцу) для ориентированных детерминантов её внутренней части.

В заключение приводятся некоторые приложения. В частности, показана связь проблем обратимости булевых матриц и разложимости по строкам или столбцам их определителей.

6.4.2. Некоторые свойства определителей

Отображение упорядоченных пар элементов некоторого конечного множества в произвольную булеву алгебру называется *булевым бинарным отношением* на этом конечном множестве. Такое булево бинарное отношение определяет некоторую квадратную булеву

матрицу с точностью до одновременных и одинаковых перестановок строк и столбцов этой матрицы. Верно и обратное, булевы квадратные матрицы, отличающиеся друг от друга указанными перестановками строк и столбцов, образуют классы эквивалентности, определяющие некоторое булево бинарное отношение.

Очевидно, чётные перестановки строк или столбцов квадратной булевой матрицы не меняют ориентированные полуперманенты. Нечётные перестановки строк или столбцов данной матрицы переводят полуперманенты $\overset{+}{\nabla} A$ и $\overline{\nabla} A$ друг в друга. Получается, что ориентированные полуперманенты и определители дают примеры булевозначных инвариантов для булевых бинарных отношений на конечном множестве.

Для дальнейших рассуждений понадобятся следующие свойства ориентированных полуперманентов. Пусть λ — элемент булевой алгебры, а A — произвольная квадратная булева матрица. Здесь и далее пересечение (объединение) элементов λ с матрицей A естественно понимается как поэлементное. Тогда

$$\overset{\pm}{\nabla}(\lambda \cap A) = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in \overset{\pm}{P}} \left((\lambda \cap a_1^{\alpha_1}) \cap \ldots \cap (\lambda \cap a_k^{\alpha_k}) \cap \ldots \cap (\lambda \cap a_n^{\alpha_n})\right) = \lambda \cap \overset{\pm}{\nabla} A$$

и

$$\overset{\pm}{\nabla}(\lambda \cup A) = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in \overset{\pm}{P}} \left((\lambda \cup a_1^{\alpha_1}) \cap \ldots \cap (\lambda \cup a_k^{\alpha_k}) \cap \ldots \cap (\lambda \cup a_n^{\alpha_n})\right) =$$

$$= \lambda \cup \left(\bigcup_{(\alpha_1,\ldots,\alpha_n)\in \overset{\pm}{P}} (a_1^{\alpha_1} \cap \ldots \cap a_k^{\alpha_k} \cap \ldots \cap a_n^{\alpha_n}) \right) = \lambda \cup \overset{\pm}{\nabla} A.$$

Эти соотношения позволяют получить тождества, указанные в следующей теореме.

Теорема 2.1. *Для произвольного элемента λ булевой алгебры и произвольной квадратной булевой матрицы A справедливы формулы*

$$\mathrm{Per}(\lambda \cap A) = \lambda \cap \mathrm{Per}\, A, \qquad \triangle(\lambda \cap A) = \lambda \cap \triangle A,$$

$$\mathrm{RDet}(\lambda \cap A) = \lambda \cap \mathrm{RDet}\, A, \quad \mathbf{LDet}(\lambda \cap A) = \lambda \cap \mathbf{LDet}\, A,$$

$$\mathrm{Per}(\lambda \cup A) = \lambda \cup \mathrm{Per}\, A, \qquad \triangle(\lambda \cup A) = \lambda \cup \triangle A,$$

$$\mathrm{RDet}(\lambda \cup A) = \lambda' \cap \mathrm{RDet}\, A = \mathrm{RDet}\, A \setminus \lambda = \mathrm{RDet}(\lambda' \cap A),$$

$$\mathbf{LDet}(\lambda \cup A) = \lambda' \cap \mathbf{LDet}\, A = \mathbf{LDet}\, A \setminus \lambda = \mathbf{LDet}(\lambda' \cap A).$$

Доказательство. Справедливость формул для перманентов и общих частей очевидна. Докажем лишь формулы для левого детерминанта:

$$\mathbf{LDet}(\lambda \cap A) = \bar{\nabla}(\lambda \cap A) \setminus \overset{+}{\nabla}(\lambda \cap A) = \bar{\nabla}(\lambda \cap A) \cap \big(\overset{+}{\nabla}(\lambda \cap A)\big)' =$$
$$= (\lambda \cap \bar{\nabla} A) \cap (\lambda \cap \overset{+}{\nabla} A)' = (\lambda \cap \bar{\nabla} A) \cap (\lambda' \cup (\overset{+}{\nabla} A)') =$$
$$= \lambda \cap \bar{\nabla} A \cap (\overset{+}{\nabla} A)' = \lambda \cap \mathbf{LDet}\, A,$$
$$\mathbf{LDet}(\lambda \cup A) = \bar{\nabla}(\lambda \cup A) \setminus \overset{+}{\nabla}(\lambda \cup A) = \bar{\nabla}(\lambda \cup A) \cap \big(\overset{+}{\nabla}(\lambda \cup A)\big)' =$$
$$= (\lambda \cup \bar{\nabla} A) \cap (\lambda \cup \overset{+}{\nabla} A)' = (\lambda \cup \bar{\nabla} A) \cap (\lambda' \cap (\overset{+}{\nabla} A)') =$$
$$= \bar{\nabla} A \cap (\overset{+}{\nabla} A)' \cap \lambda' = \mathbf{LDet}\, A \setminus \lambda.$$

Аналогично проверяются формулы для правого определителя.

6.4.3. Разложения булевой матрицы на внутреннюю, детерминированную и внешнюю части

Так как

$$(\mathbf{Per}\, A)' \cup \triangle A \cup \mathbf{RDet}\, A \cup \mathbf{LDet}\, A = I,$$

то любую булеву матрицу A можно представить в виде

$$A = [(\mathbf{Per}\, A)' \cap A] \cup [\triangle A \cap A] \cup [\mathbf{RDet}\, A \cap A] \cup [\mathbf{LDet}\, A \cap A].$$

Вводя обозначения

$$\check{A} = (\mathbf{Per}\, A)' \cap A, \quad \hat{A} = \triangle A \cap A, \quad \overset{+}{A} = \mathbf{RDet}\, A \cap A, \quad \bar{A} = \mathbf{LDet}\, A \cap A,$$

запишем такую линейную комбинацию как

$$A = \check{A} \cup \hat{A} \cup \overset{+}{A} \cup \bar{A}.$$

Как нетрудно заметить, из попарной непересекаемости $(\mathbf{Per}\, A)'$, $\triangle A$, $\mathbf{RDet}\, A$, $\mathbf{LDet}\, A$ получаются равенства

$$\mathbf{Per}\, A = \mathbf{Per}\, \check{A} \cup \mathbf{Per}\, \hat{A} \cup \mathbf{Per}\, \overset{+}{A} \cup \mathbf{Per}\, \bar{A},$$
$$\triangle A = \triangle \check{A} \cup \triangle \hat{A} \cup \triangle \overset{+}{A} \cup \triangle \bar{A},$$
$$\mathbf{RDet}\, A = \mathbf{RDet}\, \check{A} \cup \mathbf{RDet}\, \hat{A} \cup \mathbf{RDet}\, \overset{+}{A} \cup \mathbf{RDet}\, \bar{A},$$
$$\mathbf{LDet}\, A = \mathbf{LDet}\, \check{A} \cup \mathbf{LDet}\, \hat{A} \cup \mathbf{LDet}\, \overset{+}{A} \cup \mathbf{LDet}\, \bar{A},$$

дающие дизъюнктные разложения перманента, общей части ориентированных полуперманентов и определителей матрицы A, соответствующие матрицам $\check{A}$, $\hat{A}$, $\overset{+}{A}$, $\bar{A}$.

Кроме этого, справедливо следующее утверждение.

Теорема 3.1. *Перманент, общая часть полуперманентов и определители матриц* $\check{A}$, $\hat{A}$, $\overset{+}{A}$, $\bar{A}$ *обладают следующими свойствами:*

$$\mathbf{Per}\, \check{A} = 0, \quad \mathbf{Per}\, \hat{A} = \triangle \hat{A} = \triangle A,$$
$$\mathbf{Per}\, \overset{+}{A} = \mathbf{RDet}\, \overset{+}{A} = \mathbf{RDet}\, A, \quad \mathbf{Per}\, \bar{A} = \mathbf{LDet}\, \bar{A} = \mathbf{LDet}\, A,$$
$$\triangle \check{A} = \triangle \overset{+}{A} = \triangle \bar{A} = \mathbf{RDet}\, \hat{A} = \mathbf{LDet}\, \hat{A} = \mathbf{RDet}\, \check{A} = \mathbf{LDet}\, \check{A} = \mathbf{RDet}\, \bar{A} = \mathbf{LDet}\, \overset{+}{A} = 0.$$

Доказательство. Используя теорему 2.1 и определение матриц $\check{A}$, $\hat{A}$, $\overset{+}{A}$, $\bar{A}$, получаем равенства

$$\mathbf{Per}\, \check{A} = \mathbf{Per}\big((\mathbf{Per}\, A)' \cap A\big) = (\mathbf{Per}\, A)' \cap \mathbf{Per}\, A = 0;$$
$$\mathbf{Per}\, \hat{A} = \mathbf{Per}(\triangle A \cap A) = \triangle A \cap \mathbf{Per}\, A = \triangle A;$$
$$\mathbf{Per}\, \overset{+}{A} = \mathbf{Per}(\mathbf{RDet}\, A \cap A) = \mathbf{RDet}\, A \cap \mathbf{Per}\, A = \mathbf{RDet}\, A;$$
$$\mathbf{Per}\, \bar{A} = \mathbf{Per}(\mathbf{LDet}\, A \cap A) = \mathbf{LDet}\, A \cap \mathbf{Per}\, A = \mathbf{LDet}\, A.$$

Аналогично проверяются равенства

$$\triangle \check{A} = \triangle \overset{+}{A} = \triangle \bar{A} = \mathbf{RDet}\, \hat{A} = \mathbf{LDet}\, \hat{A} = \mathbf{RDet}\, \check{A} = \mathbf{LDet}\, \check{A} = \mathbf{RDet}\, \bar{A} = \mathbf{LDet}\, \overset{+}{A} = 0.$$

Тогда

$$\triangle A = \triangle \check{A} \cup \triangle \hat{A} \cup \triangle \overset{+}{A} \cup \triangle \bar{A} = \mathbf{0} \cup \triangle \hat{A} \cup 0 \cup \mathbf{0} = \triangle \hat{A};$$

$$\mathbf{RDet}\, A = \mathbf{RDet}\, \check{A} \cup \mathbf{RDet}\, \hat{A} \cup \mathbf{RDet}\, \overset{+}{A} \cup \mathbf{RDet}\, \bar{A} =$$
$$= 0 \cup 0 \cup \mathbf{RDet}\, \overset{+}{A} \cup 0 = \mathbf{RDet}\, \overset{+}{A};$$
$$\mathbf{LDet}\, A = \mathbf{LDet}\, \check{A} \cup \mathbf{LDet}\, \hat{A} \cup \mathbf{LDet}\, \overset{+}{A} \cup \mathbf{LDet}\, \bar{A} =$$
$$= 0 \cup 0 \cup 0 \cup \mathbf{LDet}\, \bar{A} = \mathbf{LDet}\, \bar{A}.$$

Определение 3.1. Назовём матрицу $\hat{A} \cup \check{A}$ *вырожденной частью* матрицы A, состоящей из *внутренности* $\hat{A}$ и *внешности* $\check{A}$ матрицы A. Матрицу $\overset{+}{A} \cup \bar{A}$ назовём *невырожденной* (или *детерминированной) частью,* состоящей из *положительной* $\overset{+}{A}$ и *отрицательной* $\bar{A}$ частей. Ненулевую матрицу назовём *внешней, внутренней, положительной, отрицательной,* если она совпадает со своей внешней, внутренней, положительной, отрицательной частью соответственно.
Для нулевой матрицы Θ приемлемо любое название: она является одновременно и внешней, и внутренней, и положительной, и отрицательной.

Различные примеры проявления введённой таким образом ориентируемости и вырожденности булевых бинарных отношений (или булевых матриц) можно найти в литературе.
Следующая теорема показывает однозначность разложения булевой матрицы на внутреннюю, невырожденную и внешнюю части.

Теорема 3.2. *Предположим, что ненулевую квадратную булеву матрицу A представили линейной комбинацией матриц с какими-то попарно непересекающимися коэффициентами* $\alpha^i \ (i = 1, \ldots, 4)$ *вида*

$$A = (\alpha^1 \cap A) \cup (\alpha^2 \cap A) \cup (\alpha^3 \cap A) \cup (\alpha^4 \cap A) = A_1 \cup A_2 \cup A_3 \cup A_4,$$

причём $A_1 = \alpha^1 \cap A = \check{A}_1$ *есть внешняя булева матрица, матрица* $A_2 = \alpha^2 \cap A = \hat{A}_2$ *является внутренней,* $A_3 = \alpha^3 \cap A = \overset{+}{A}_3$ *— положительной и* $A_4 = \alpha^4 \cap A = \bar{A}_4$ *— отрицательной матрицами соответственно. Тогда* A_1, A_2, A_3, A_4 *являются внешней, внутренней, положительной и отрицательной частями матрицы A соответственно.*

Доказательство. Из теоремы 3.1 и того, что любой элемент матрицы A содержится в объединении $\bigcup_{i=1}^{4} \alpha^i$ попарно непересекающихся коэффициентов $\alpha^i \ (i = 1, \ldots, 4)$, следует, что

$$\triangle A = \triangle A_1 \cup \triangle A_2 \cup \triangle A_3 \cup \triangle A_4 = 0 \cup \triangle A_2 \cup 0 \cup 0 = \triangle A_2;$$

$$\begin{aligned}\mathrm{RDet}\, A &= \mathrm{RDet}\, A_1 \cup \mathrm{RDet}\, A_2 \cup \mathrm{RDet}\, A_3 \cup \mathrm{RDet}\, A_4 = \\ &= 0 \cup 0 \cup \mathrm{RDet}\, A_3 \cup 0 = \mathrm{RDet}\, A_3;\end{aligned}$$

$$\begin{aligned}\mathrm{LDet}\, A &= \mathrm{LDet}\, A_1 \cup \mathrm{LDet}\, A_2 \cup \mathrm{LDet}\, A_3 \cup \mathrm{LDet}\, A_4 = \\ &= 0 \cup 0 \cup 0 \cup \mathrm{LDet}\, A_4 = \mathrm{LDet}\, A_4.\end{aligned}$$

То, что

$$A_2 = \alpha_2 \cap A = \alpha_2 \cap \alpha_2 \cap A = \alpha_2 \cap A_2,$$

позволяет записать

$$\triangle A = \triangle A_2 = \triangle(\alpha_2 \cap A_2) = \alpha_2 \cap \triangle A_2 \subseteq \alpha_2.$$

Аналогично из

$$A_3 = \alpha_3 \cap A_3$$

получим

$$\mathrm{RDet}\, A = \mathrm{RDet}\, A_3 \subset \alpha_3,$$

а из

$$A_4 = \alpha_4 \cap A_4$$

получим

$$\mathrm{LDet}\, A = \mathrm{LDet}\, A_4 \subset \alpha_4.$$

Тогда

$$\alpha_1 \cap \mathrm{Per}\, A = \alpha_1 \cap (\triangle A \cup \mathrm{RDet}\, A \cup \mathrm{LDet}\, A) \subseteq \alpha_1 \cap (\alpha_2 \cup \alpha_3 \cup \alpha_4) = 0,$$

так как $\alpha^i \ (i = 1, \ldots, 4)$ попарно не пересекаются. Это означает, что

$$(\mathrm{Per}\, A)' \cap \alpha_1 = \alpha_1,$$

и тогда

$$(\mathrm{Per}\, A)' \cap A_1 = (\mathrm{Per}\, A)' \cap \alpha_1 \cap A = \alpha_1 \cap A = A_1.$$

Покажем, что

$$(\mathrm{Per}\, A)' \cap A_2 = (\mathrm{Per}\, A)' \cap A_3 = (\mathrm{Per}\, A)' \cap A_4 = \Theta.$$

Действительно,

$$\begin{aligned}(\mathrm{Per}\, A)' \cap A_2 &= (\triangle A \cup \mathrm{RDet}\, A \cup \mathrm{LDet}\, A)' \cap A_2 = \\ &= (\triangle A)' \cap (\mathrm{RDet}\, A)' \cap (\mathrm{LDet}\, A)' \cap A_2 \subseteq (\triangle A)' \cap A_2 = (\triangle A)' \cap \hat{A}_2 = \\ &= (\triangle A)' \cap (\triangle A_2 \cap A_2) = (\triangle A)' \cap \triangle A \cap A_2 = \Theta;\end{aligned}$$

$$\begin{aligned}(\mathrm{Per}\, A)' \cap A_3 &= (\triangle A \cup \mathrm{RDet}\, A \cup \mathrm{LDet}\, A)' \cap A_3 = \\ &= (\triangle A)' \cap (\mathrm{RDet}\, A)' \cap (\mathrm{LDet}\, A)' \cap A_3 \subseteq (\mathrm{RDet}\, A)' \cap A_3 = (\mathrm{RDet}\, A)' \cap \overset{+}{A}_3 = \\ &= (\mathrm{RDet}\, A)' \cap (\mathrm{RDet}\, A_3 \cap A_3) = (\mathrm{RDet}\, A)' \cap \mathrm{RDet}\, A \cap A_3 = \Theta;\end{aligned}$$

$$\begin{aligned}(\mathrm{Per}\, A)' \cap A_4 &= (\triangle A \cup \mathrm{RDet}\, A \cup \mathrm{LDet}\, A)' \cap A_4 = \\ &= (\triangle A)' \cap (\mathrm{RDet}\, A)' \cap (\mathrm{LDet}\, A)' \cap A_4 \subseteq (\mathrm{LDet}\, A)' \cap A_4 = (\mathrm{LDet}\, A)' \cap \bar{A}_4 = \\ &= (\mathrm{LDet}\, A)' \cap (\mathrm{LDet}\, A_4 \cap A_4) = (\mathrm{LDet}\, A)' \cap \mathrm{LDet}\, A \cap A_4 = \Theta.\end{aligned}$$

Тогда получаем

$$\check{A} = (\mathrm{Per}\, A)' \cap A = (\mathrm{Per}\, A)' \cap (A_1 \cup A_2 \cup A_3 \cup A_4) = (\mathrm{Per}\, A)' \cap A_1 = A_1,$$

а также

$$\begin{aligned}\hat{A} &= \triangle A \cap A = \triangle A \cap (A_1 \cup A_2 \cup A_3 \cup A_4) = \triangle A \cap A_2 = A_2; \\ \overset{+}{A} &= \mathrm{RDet}\, A \cap A = \mathrm{RDet}\, A \cap (A_1 \cup A_2 \cup A_3 \cup A_4) = \mathrm{RDet}\, A \cap A_3 = A_3; \\ \bar{A} &= \mathrm{LDet}\, A \cap A = \mathrm{LDet}\, A \cap (A_1 \cup A_2 \cup A_3 \cup A_4) = \mathrm{LDet}\, A \cap A_4 = A_4.\end{aligned}$$

Заметим, что непустые бинарные отношения на конечном множестве, представляемые квадратными *(0,1)*-матрицами над двухэлементной булевой алгеброй, принадлежат одному из четырёх типов: они бывают или внешние, или внутренние, или положительные, или трицательные.

6.4.4. Формулы Лапласа для перманентов булевых матриц

Определение 4.1. Обозначим $((n-1)\times(n-1))$-матрицу, получаемую из $(n\times n)$-матрицы A удалением из неё i-й строки и k-го столбца с сохранением порядка следования остающихся строк и столбцов, через $\partial_k^i A$ $(i,k=1,2,\ldots,n)$.

Ясно, что

$$\partial_k^i(\lambda\cup A)=\lambda\cup\partial_k^i A,\qquad \partial_k^i(\lambda\cap A)=\lambda\cap\partial_k^i A,$$
$$\partial_k^i(A\cup B)=\partial_k^i A\cup\partial_k^i B,\qquad \partial_k^i(A\cap B)=\partial_k^i A\cap\partial_k^i B$$

для любых квадратных булевых матриц A, B и любого элемента λ булевой алгебры.

Запишем определяющее перманент равенство

$$\operatorname{Per} A=\bigcup_{(\alpha_1,\ldots,\alpha_n)\in\overset{+}{P}\cup\overset{-}{P}}(a_1^{\alpha_1}\cap\ldots\cap a_k^{\alpha_k}\cap\ldots\cap a_n^{\alpha_n})$$

для произвольного номера $k=1,\ldots,n$ в виде

$$\operatorname{Per} A=\bigcup_{\alpha_k=1}^{n}\left(a_k^{\alpha_k}\cap\bigcup_{(\alpha_1,\ldots,\alpha_{k-1},\alpha_{k+1},\ldots,\alpha_n)\in P_{n-1}}(a_1^{\alpha_1}\cap\ldots\cap a_{k-1}^{\alpha_{k-1}}\cap a_{k+1}^{\alpha_{k+1}}\cap\ldots\cap a_n^{\alpha_n})\right).$$

Здесь каждая перестановка $(\alpha_1,\ldots,\alpha_{k-1},\alpha_{k+1},\ldots,\alpha_n)$ есть (n —1)-перестановка номеров, отличных от числа α_k. С другой стороны,

$$\bigcup_{(\alpha_1,\ldots,\alpha_{k-1},\alpha_{k+1},\ldots,\alpha_n)\in P_{n-1}}(a_1^{\alpha_1}\cap\ldots\cap a_{k-1}^{\alpha_{k-1}}\cap a_{k+1}^{\alpha_{k+1}}\cap\ldots\cap a_n^{\alpha_n})=\operatorname{Per}\partial_k^{\alpha_k}A.$$

Если теперь положить $\alpha_k=i$, то получим *формулу разложения перманента* по k-му столбцу:

$$\operatorname{Per} A=\bigcup_{i=1}^{n}(a_k^i\cap\operatorname{Per}\partial_k^i A).$$

Похожая формула верна и для строк.

Определение 4.2. Пусть $m\in\mathbb{N}$. Определим *функции знака* $\overset{m}{\sigma}$ на ориентированных полуперманентах квадратной булевой матрицы A следующим образом:

$\overset{m}{\sigma}(\overset{+}{\nabla})A=\overset{+}{\nabla}A$, $\overset{m}{\sigma}(\overset{-}{\nabla})A=\overset{-}{\nabla}A$, если m чётное, и $\overset{m}{\sigma}(\overset{+}{\nabla})A=\overset{-}{\nabla}A$, $\overset{m}{\sigma}(\overset{-}{\nabla})A=\overset{+}{\nabla}A$, если m нечётное.

Функции знака помогут записать теперь *разложения полуперманентов* по i-й строке (или аналогичные формулы разложения по столбцу):

$$\overset{+}{\nabla}A=\bigcup_{i=1}^{n}(a_k^i\cap\overset{i+k}{\sigma}(\overset{+}{\nabla})\partial_k^i A),\quad \overset{-}{\nabla}A=\bigcup_{i=1}^{n}(a_k^i\cap\overset{i+k}{\sigma}(\overset{-}{\nabla})\partial_k^i A).$$

Доказательство последних формул можно получить похожими рассуждениями так, как это было сделано при выводе формулы разложения перманента.

6.4.5. Комбинаторные свойства внешних и детерминированных булевых матриц

Определение 5.1. Последовательность $(a_1^{\alpha_1},a_2^{\alpha_2},\ldots,a_n^{\alpha_n})$ элементов матрицы A назовём *нечётной (чётной) диагональю* данной матрицы A *с ненулевым пересечением,* если $a_1^{\alpha_1}\cap a_2^{\alpha_2}\cap\ldots\cap a_n^{\alpha_n}\neq 0$ для нечётных (соответственно чётных) перестановок верхних индексов $(\alpha_1,\alpha_2,\ldots,\alpha_n)$. Диагональ $(a_1^{\alpha_1},a_2^{\alpha_2},\ldots,a_n^{\alpha_n})$ назовём с нулевым пересечением, если $a_1^{\alpha_1}\cap a_2^{\alpha_2}\cap\ldots\cap a_n^{\alpha_n}=0$.

Будем также говорить, что диагональ *проходит через элементы,* если они являются её составляющими.

Следующее очевидное утверждение даёт признак внешности.

Теорема 5.1. *Ненулевая матрица A является внешней, т. е. для матрицы выполнено условие* $\operatorname{Per} A=\overset{+}{\nabla}A=\overset{-}{\nabla}A=0$, *тогда и только тогда, когда все диагонали в матрице A являются диагоналями с нулевым пересечением.*

Доказательство. Действительно, равенство

$$\operatorname{Per} A=\bigcup_{(\alpha_1,\ldots,\alpha_n)\in\overset{+}{P}\cup\overset{-}{P}}(a_1^{\alpha_1}\cap a_2^{\alpha_2}\cap\ldots\cap a_n^{\alpha_n})=0$$

эквивалентно равенству $a_1^{\alpha_1}\cap a_2^{\alpha_2}\cap\ldots\cap a_n^{\alpha_n}=0$ для всех перестановок верхних индексов $(\alpha_1,\alpha_2,\ldots,\alpha_n)$.

Ниже приводится ещё один признак внешних ненулевых булевых матриц.

Теорема 5.2. *Булева матрица A является внешней тогда и только тогда, когда для всех значений индексов i и k матрица* $a_k^i\cap\partial_k^i A$ *является внешней.*

Доказательство. Матрица A является внешней тогда и только тогда, когда выполняется равенство

$$\operatorname{Per} A=\bigcup_{i=1}^{n}(a_k^i\cap\operatorname{Per}\partial_k^i A)=0,$$

и следовательно, тогда и только тогда, когда

$$a_k^i\cap\operatorname{Per}\partial_k^i A=\operatorname{Per}(a_k^i\cap\partial_k^i A)=0$$

для всех значений индексов i и k.
Равенство $\mathbf{Per}(a_k^i \cap \partial_k^i A) = 0$ означает, что
для всех значений индексов i и k матрица $a_k^i \cap \partial_k^i A$ является внешней.
Укажем теперь признак таких булевых матриц, для которых выполнено условие $\mathbf{Per}\, A = \mathbf{RDet}\, A \neq 0$ (или $\mathbf{Per}\, A = \mathbf{LDet}\, A \neq 0$).Такое условие эквивалентно тому, что матрица A может быть представлена как

$$A = [(\mathbf{Per}\, A)' \cap A] \cup [\mathbf{Per}\, A \cap A] = [(\mathbf{Per}\, A)' \cap A] \cup [\mathbf{RDet}\, A \cap A] = \check{A} \cup \overset{+}{A}$$

(или $A = \check{A} \cup \overset{-}{A}$ соответственно), где $\overset{+}{A}$, $\overset{-}{A}$— ненулевые положительные и отрицательные части.

Теорема 5.3. *Условие* $\mathbf{Per}\, A = \mathbf{RDet}\, A \neq 0$ (*или* $\mathbf{Per}\, A = \mathbf{LDet}\, A \neq 0$) *верно для булевой матрицы A тогда и только тогда, когда существует чётная (нечётная) диагональ с ненулевым пересечением и все нечётные (соответственно чётные) диагонали этой матрицы A являются диагоналями с нулевым пересечением.*

Доказательство. Действительно, для квадратной булевой матрицы A соотношение $\mathbf{Per}\, A = \mathbf{RDet}\, A \neq 0$выполняется тогда и только тогда, когда

$$\mathbf{RDet}\, A = \overset{+}{\nabla} A = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in \overset{+}{P}} (a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n}) \neq 0$$

и

$$\triangle A = \mathbf{LDet}\, A = \overset{-}{\nabla} A = \bigcup_{(\alpha_1,\ldots,\alpha_n)\in \overset{-}{P}} (a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n}) = 0.$$

Но это равносильно тому, что найдутся такие наборы индексов $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \overset{+}{P}$, что $a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n} \neq 0$, и будет выполнено $a_1^{\beta_1} \cap a_2^{\beta_2} \cap \ldots \cap a_n^{\beta_n} = 0$ для любых нечётных перестановок индексов $(\beta_1, \beta_2, \ldots, \beta_n) \in \overset{-}{P}$.
Случай $\mathbf{Per}\, A = \mathbf{LDet}\, A \neq 0$ проверяется аналогично рассмотренному.

Определение 5.2. Пусть $m \in \mathbb{N}$. Определим *функции знака* $\overset{m}{\sigma}$ *на ориентированных определителях* квадратной булевой матрицы A следующим образом:

$$\overset{m}{\sigma}(\mathbf{RDet})A = \mathbf{RDet}\, A, \quad \overset{m}{\sigma}(\mathbf{LDet})A = \mathbf{LDet}\, A,$$

если m чётное, и

$$\overset{m}{\sigma}(\mathbf{RDet})A = \mathbf{LDet}\, A, \quad \overset{m}{\sigma}(\mathbf{LDet})A = \mathbf{RDet}\, A,$$

если m нечётное.

Введём также следующие обозначения для определителей. Посредством $(\mathbf{R})(\mathbf{L})\mathbf{Det}\, A$ будем обозначать или правый определитель $\mathbf{RDet}\, A$, или левый определитель $\mathbf{LDet}\, A$, или определитель $\mathbf{Det}\, A$ матрицы A, если тип определителя не имеет значения.

Теорема 5.4. *Для булевой матрицы A условие* $\mathbf{Per}\, A = \mathbf{RDet}\, A \neq 0$ *(или* $\mathbf{Per}\, A = \mathbf{LDet}\, A \neq 0$) *выполнено тогда и только тогда, когда для любой пары индексов i и k выполняется одно из следующих двух условий:*

1) $\mathbf{Per}(a_k^i \cap \partial_k^i A) = \overset{i+k}{\sigma}(\mathbf{RDet})(a_k^i \cap \partial_k^i A) \neq 0$ *соответственно* $\mathbf{Per}(a_k^i \cap \partial_k^i A) = \overset{i+k}{\sigma}(\mathbf{LDet})(a_k^i \cap \partial_k^i A) \neq 0$),
2) $\mathbf{Per}(a_k^i \cap \partial_k^i A) = 0$,

причём условие 1) *выполняется по крайней мере для одной пары индексов.*

Доказательство. Если $\mathbf{Per}\, A = \mathbf{RDet}\, A \neq 0$, то найдётся такой набор индексов $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \overset{+}{P}$, что

$$a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \ldots \cap a_n^{\alpha_n} \neq 0$$

и выполняется

$$a_1^{\beta_1} \cap a_2^{\beta_2} \cap \ldots \cap a_n^{\beta_n} = 0$$

для любых нечётных перестановок индексов
$(\beta_1, \beta_2, \ldots, \beta_n) \in \overset{-}{P}$ (β1,β2,...,βn) (теорема 5.3).
Предположим, что $\alpha_k = \beta_k = i$, тогда

$$a_1^{\alpha_1} \cap \ldots \cap a_k^i \cap \ldots \cap a_n^{\alpha_n} \neq 0,$$

если $(\alpha_1, \ldots, i, \ldots, \alpha_n) \in \overset{+}{P}$, и

$$a_1^{\beta_1} \cap \ldots \cap a_k^i \cap \ldots \cap a_n^{\beta_n} = 0,$$

если $(\beta_1, \ldots, i, \ldots, \beta_n) \in \overset{-}{P}$.
Это означает, что диагонали $(a_1^{\alpha_1}, \ldots, a_k^i, \ldots, a_n^{\alpha_n})$ и $(a_1^{\beta_1}, \ldots, a_k^i, \ldots, a_n^{\beta_n})$ проходят через элемент a_k^i. Очевидно, что это возможно всегда при $n \geqslant 3$. Таким образом, условие

$$a_1^{\alpha_1} \cap \ldots \cap a_k^i \cap \ldots \cap a_n^{\alpha_n} \neq 0$$

влечёт существование диагонали или нескольких диагоналей $(a_k^i \cap a_1^{\alpha_1}, a_k^i \cap a_2^{\alpha_2}, \ldots, a_k^i \cap a_{k-1}^{\alpha_{k-1}}, a_k^i \cap a_{k+1}^{\alpha_{k+1}}, \ldots, a_k^i \cap a_n^{\alpha_n})$ в матрице $a_k^i \cap \partial_k^i A$ с ненулевым пересечением, которые являются чётными, если сумма $i + k$ чётная, или нечётными, если $i + k$ — нечётное число. Все остальные диагонали в матрице $a_k^i \cap \partial_k^i A$ являются диагоналями с нулевым пересечением.

Это следует из того, что нужно совершить $i + k - 2$ транспозиций соседних строк и столбцов матрицы A, чтобы вывести элемент a_k^i на первую строчку и в первый столбец. Получили, что для матрицы $a_k^i \cap \partial_k^i A$ выполнено

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = {}^{i+k}\hat{\sigma}(\mathrm{RDet})(a_k^i \cap \partial_k^i A) \neq 0.$$

Предположим теперь, что ни одна из чётных диагоналей $(a_1^{\alpha_1}, a_2^{\alpha_2}, \ldots, a_n^{\alpha_n})$ матрицы A с ненулевым пересечением не содержит элемента a_k^i. Это будет означать, что $a_1^{\alpha_1} \cap \ldots \cap a_k^i \cap \ldots \cap a_n^{\alpha_n} = 0$ для любых чётных (и, конечно же, для нечётных) перестановок $(\alpha_1, \ldots, i, \ldots, \alpha_n)$. Тогда все диагонали в матрице $a_k^i \cap \partial_k^i A$ являются диагоналями с нулевым пересечением. Следовательно, матрица $a_k^i \cap \partial_k^i A$ является внешностью, т. е. $\mathrm{Per}(a_k^i \cap \partial_k^i A) = 0$, либо она является нулевой матрицей.

Докажем теперь достаточность условия теоремы 5.4. Пусть по крайней мере для одной пары индексов i и k выполнено условие

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = \mathrm{RDet}(a_k^i \cap \partial_k^i A) \neq 0,$$

если $i + k$ чётно, или выполнено условие

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = \mathrm{LDet}(a_k^i \cap \partial_k^i A) \neq 0,$$

если $i + k$ нечётно, а для всех остальных значений индексов i и k выполнено $\mathrm{Per}(a_k^i \cap \partial_k^i A) = 0$, т. е. матрица $a_k^i \cap \partial_k^i A$ является внешней, возможно нулевой. Следует показать, что тогда для булевой матрицы A выполняется условие

$$\mathrm{Per}\, A = \mathrm{RDet}\, A \neq 0.$$

Пусть Ω — непустое множество элементов a_k^i матрицы A, для которых выполняется

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = \mathrm{RDet}(a_k^i \cap \partial_k^i A) \neq 0,$$

если $i + k$ чётно, или

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = \mathrm{LDet}(a_k^i \cap \partial_k^i A) \neq 0,$$

если $i+k$ нечётно. Пусть для определённости множество Ω содержит элемент a_1^1. Тогда по теореме 5.3 существует чётная диагональ с ненулевым пересечением, и все нечётные диагонали матрицы $a_1^1 \cap \partial_1^1 A$ имеют нулевое пересечение. Последнее будет означать, что в матрице A можно найти чётную диагональ с ненулевым пересечением, содержащую a_1^1, а все остальные диагонали матрицы A, содержащие a_1^1, являются диагоналями с нулевым пересечением. То же самое дают

и другие случаи выбора $a_k^i \in \Omega$: в матрице A можно найти чётную диагональ с ненулевым пересечением, содержащую a_k^i, а все остальные диагонали матрицы A, содержащие a_k^i, являются диагоналями с нулевым пересечением.

Все чётные и нечётные диагонали матрицы A, не содержащие элементов матрицы A, входящих в Ω, являются диагоналями с нулевым пересечением. Это следует из того, что все матрицы $a_k^i \cap \partial_k^i A$ являются внешними, если a_k^i не принадлежит Ω. Тогда, учитывая теорему 5.1, мы не можем найти диагонали с ненулевым пересечением. Таким образом, мы доказали, что $\mathrm{Per}\, A = \mathrm{RDet}\, A \neq 0$.

Похожими рассуждениями доказывается теорема 5.4 в случае $\mathrm{Per}\, A = \mathrm{LDet}\, A \neq 0$.

6.4.6. Формулы Лапласа для булевых матриц с нулевой внутренностью

Определение 6.1. Ненулевая булева матрица называется *матрицей с нулевой внутренностью (без внутренности),* если её внутренняя часть есть нулевая матрица ($\hat{A} = \Theta$ или, что то же самое, $\triangle A = 0$).

Теорема 6.1. *Пусть A — квадратная матрица над произвольной булевой алгеброй с нулевой внутренностью. Тогда справедливы разложения детерминантов по любой i-й строке этой матрицы:*

$$\mathrm{RDet}\, A = \bigcup_{k=1}^{n} (a_k^i \cap {}^{i+k}\hat{\sigma}(\mathrm{RDet})\partial_k^i A), \quad (1)$$

$$\mathrm{LDet}\, A = \bigcup_{k=1}^{n} (a_k^i \cap {}^{i+k}\hat{\sigma}(\mathrm{LDet})\partial_k^i A). \quad (2)$$

Из (1) *и* (2) *получается разложение определителя по i-й строке:*

$$\mathrm{Det}\, A = \bigcup_{k=1}^{n} (a_k^i \cap \mathrm{Det}\, \partial_k^i A). \quad (3)$$

Аналогичное утверждение справедливо и для столбцов.

Доказательство. Как уже отмечалось в теореме 5.2, для любой внешней матрицы, определяемой условием

$$\mathrm{Per}\, A = \overset{+}{\nabla} A = \overset{-}{\nabla} A = (\mathrm{R})(\mathrm{L})\mathrm{Det}\, A = 0,$$

выполнено

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = a_k^i \cap \mathrm{Per}\, \partial_k^i A = 0$$

для всех значений i, k. Следовательно,

$$a_k^i \cap (\mathrm{R})(\mathbf{L})\mathrm{Det}\,\partial_k^i A \subseteq a_k^i \cap \mathrm{Per}\,\partial_k^i A = 0$$

и

$$a_k^i \cap (\mathrm{R})(\mathbf{L})\mathrm{Det}\,\partial_k^i A = 0$$

для всех значений i, *k*. Тогда указанные в теореме формулы (1) и (2) выполняются для внешних матриц, так как в правых и левых частях стоят нули.

Докажем формулы (1) и (2) для детерминированных матриц. Проведём рассуждения, полагая для определённости, что A есть положительная ненулевая матрица, т. е. удовлетворяет условию $A = \overset{+}{A} = \mathrm{RDet}\,A \cap A$. Следовательно, $\mathrm{RDet}\,A = \mathrm{Per}\,A \neq 0$, и разложение перманента даёт равенство

$$\mathrm{RDet}\,A = \mathrm{Per}\,A = \bigcup_{k=1}^{n}(a_k^i \cap \mathrm{Per}\,\partial_k^i A) = \bigcup_{k=1}^{n}\mathrm{Per}(a_k^i \cap \partial_k^i A).$$

Из теоремы 5.4 получаем

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = {}^{i+k}\sigma(\mathrm{RDet})(a_k^i \cap \partial_k^i A) \neq 0$$

по крайней мере для одной пары индексов i и k. Но возможен случай $\mathrm{Per}(a_k^i \cap \partial_k^i A) = 0$, в котором снова получается равенство

$$\mathrm{Per}(a_k^i \cap \partial_k^i A) = {}^{i+k}\sigma(\mathrm{RDet})(a_k^i \cap \partial_k^i A) = 0,$$

так как

$${}^{i+k}\sigma(\mathrm{RDet})(a_k^i \cap \partial_k^i A) \subseteq \mathrm{Per}(a_k^i \cap \partial_k^i A).$$

Следовательно, формулы (1) и (2) выполняются для детерминированных матриц. Докажем теперь теорему для общего случая, т. е. для матрицы с нулевой внутренностью $\hat{A} = \Theta$. Тогда

$$A = \check{A} \cup \hat{A} \cup \overset{+}{A} \cup \bar{A} = \check{A} \cup \overset{+}{A} \cup \bar{A}$$

и

$$\bigcup_{k=1}^{n}(a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i A) = \bigcup_{k=1}^{n}\left(a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i(\check{A} \cup \overset{+}{A} \cup \bar{A})\right) =$$

$$= \bigcup_{k=1}^{n}\{a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i[((\mathrm{Per}\,A)' \cap A) \cup (\mathrm{RDet}\,A \cap A) \cup (\mathbf{L}\mathrm{Det}\,A \cap A)]\} =$$

$$= \bigcup_{k=1}^{n}\{a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})[((\mathrm{Per}\,A)' \cap \partial_k^i A) \cup (\mathrm{RDet}\,A \cap \partial_k^i A) \cup (\mathbf{L}\mathrm{Det}\,A \cap \partial_k^i A)]\}.$$

Последнее выражение с учётом указанных в теореме 2.1 формул для определителей, а также того, что булевы коэффициенты $(\mathrm{Per}\,A)'$, $\mathrm{RDet}\,A$, $\mathbf{L}\mathrm{Det}\,A$ попарно не пересекаются, даёт

$$\bigcup_{k=1}^{n}\{[((\mathrm{Per}\,A)' \cap a_k^i) \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i((\mathrm{Per}\,A)' \cap A)] \cup$$

$$\cup [(\mathrm{RDet} \cap a_k^i) \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i(\mathrm{RDet} \cap A)] \cup$$

$$\cup [(\mathbf{L}\mathrm{Det} \cap a_k^i) \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i(\mathbf{L}\mathrm{Det} \cap A)]\} =$$

$$= \bigcup_{k=1}^{n}(\check{a}_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i \check{A}) \cup \bigcup_{k=1}^{n}(\overset{+}{a}_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i \overset{+}{A}) \cup$$

$$\cup \bigcup_{k=1}^{n}(\bar{a}_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i \bar{A}) = \mathrm{RDet}\,\check{A} \cup \mathrm{RDet}\,\overset{+}{A} \cup \mathrm{RDet}\,\bar{A} = \mathrm{RDet}\,A.$$

Здесь через $\check{a}_k^i$, $\overset{+}{a}_k^i$, $\bar{a}_k^i$ обозначены соответственно элементы матриц $\check{A}$, $\overset{+}{A}$, $\bar{A}$, стоящие в *i*-й строке и *k*-м столбце.

Аналогично проверяются формулы (2).

Пример 6.1. Для матрицы

$$D = \begin{pmatrix} (2{,}5;5) & [0;2{,}5] & [0;3] \\ (2{,}5;4) & (2{,}5;3] & [2;3] \\ [0;8] & (2;5) & [2{,}5;5] \end{pmatrix}$$

с элементами, являющимися подмножествами множества действительных чисел, выполняется

$\overset{+}{\nabla}D = [2;3]$, $\bar{\nabla}D = (2{,}5;3]$, $\triangle D = (2{,}5;3]$, $\mathrm{RDet}\,D = [2;2{,}5]$, $\mathbf{L}\mathrm{Det}\,D = \varnothing$.

Тогда

$$A = D \setminus (2{,}5;3] = \begin{pmatrix} (3;5) & [0;2{,}5] & [0;2{,}5] \\ (3;4) & \varnothing & [2;2{,}5] \\ [0;2{,}5] \cup (3;8] & (2;2{,}5] \cup (3;5) & \{2{,}5\} \cup (3;5] \end{pmatrix}$$

есть матрица без внутренности, причём

$$\mathrm{Det}\,A = \mathrm{RDet}\,A = \overset{+}{\nabla}A = [2;2{,}5], \quad \mathbf{L}\mathrm{Det}\,A = \triangle A = \bar{\nabla}A = \varnothing.$$

Можно осуществить разложение по формуле (3) по 3-й строке определителя Det*A:*

$$\left\{\{[0;2{,}5]\cup(3;8]\}\cap\mathbf{Det}\begin{pmatrix}[0;2{,}5] & [0;2{,}5]\\ \varnothing & [2;2{,}5]\end{pmatrix}\right\}\cup$$
$$\cup\left\{\{(2;2{,}5]\cup(3;5)\}\cap\mathbf{Det}\begin{pmatrix}(3;5) & [0;2{,}5]\\ (3;4) & [2;2{,}5]\end{pmatrix}\right\}\cup$$
$$\cup\left\{\{\{2{,}5\}\cup(3;5]\}\cap\mathbf{Det}\begin{pmatrix}(3;5) & [0;2{,}5]\\ (3;4) & \varnothing\end{pmatrix}\right\}=$$
$$=\{\{[0;2{,}5]\cup(3;8]\}\cap[2;2{,}5]\}\cup\{\{(2;2{,}5]\cup(3;5)\}\cap[2;2{,}5]\}\cup$$
$$\cup\left\{\{\{2{,}5\}\cup(3;5]\}\cap\varnothing\right\}=[2;2{,}5].$$

6.4.7. Разложения Лапласа и вырожденные матрицы

Следующие примеры показывают, что разложимость по формулам (1) и (2) свойственна не только матрицам с нулевой внутренностью.

Пример 7.1. Формулы (1) и (2) для внутренней матрицы

$$\begin{pmatrix}I & I & 0\\ I & I & I\\ 0 & I & I\end{pmatrix}$$

не выполняются при разложении, например, по первой строке. С одной стороны,

$$(\mathrm{R})(\mathbf{L})\mathbf{Det}\begin{pmatrix}I & I & 0\\ I & I & I\\ 0 & I & I\end{pmatrix}=\mathbf{0}.$$

С другой стороны, правые части формул (1) и (2) разложений по первой строке этой матрицы дают соответственно

$$\left(I\cap\mathrm{RDet}\begin{pmatrix}I & I\\ I & I\end{pmatrix}\right)\cup\left(I\cap\mathrm{LDet}\begin{pmatrix}I & I\\ 0 & I\end{pmatrix}\right)\cup\left(0\cap\mathrm{RDet}\begin{pmatrix}I & I\\ 0 & I\end{pmatrix}\right)=0\cup0\cup0=0$$

и

$$\left(I\cap\mathrm{LDet}\begin{pmatrix}I & I\\ I & I\end{pmatrix}\right)\cup\left(I\cap\mathrm{RDet}\begin{pmatrix}I & I\\ 0 & I\end{pmatrix}\right)\cup\left(0\cap\mathrm{LDet}\begin{pmatrix}I & I\\ 0 & I\end{pmatrix}\right)=0\cup I\cup0=I.$$

Однако для внутренней матрицы

$$J=\begin{pmatrix}I & \cdots & I\\ \vdots & & \vdots\\ I & \cdots & I\end{pmatrix},$$

все элементы которой единицы, формулы (1) и (2) выполняются для любых её строк и столбцов.

Пусть A — вырожденная матрица, т. е. $A=\check{A}\cup\hat{A}$. Необходимым и достаточным признаком вырожденной матрицы является равенство ориентированных полуперманентов $\overset{+}{\nabla}A=\overset{-}{\nabla}A$. Действительно, равенство $\overset{+}{\nabla}A=\overset{-}{\nabla}A$ эквивалентно равенству $\mathbf{Det}\,A=\mathbf{0}$, т. е. $\overset{+}{A}=\overset{-}{A}=\Theta$ (теорема 3.1). Учитывая это и формулы разложения полуперманентов по строке (столбцу), получаем, что необходимым и достаточным признаком вырожденной матрицы является равенство

$$\overset{+}{\nabla}A=\bigcup_{k=1}^{n}(a_k^i\cap\overset{i+k}{\sigma}(\overset{+}{\nabla})\partial_k^iA)=\bigcup_{k=1}^{n}(a_k^i\cap\overset{i+k}{\sigma}(\overset{-}{\nabla})\partial_k^iA)=\overset{-}{\nabla}A. \quad (4)$$

Лемма 7.1. *Формулы* (1) *и* (2) *разложения ориентированных определителей вырожденной матрицы A по строке (столбцу) выполняются тогда и только тогда, когда выполняется формула* (3) *для разложения определителя матрицы A по этой строке (столбцу).*

Доказательство. Очевидно, что из формул (1) и (2) всегда следует (3). Из формулы (3)

$$\mathbf{Det}\,A=0=\bigcup_{k=1}^{n}(a_k^i\cap\mathbf{Det}\,\partial_k^iA)$$

для вырожденной матрицы A получаем $\mathrm{RDet}\,A=\mathbf{LDet}\,A=\mathbf{0}$. С другой стороны,

$$a_k^i\cap\mathbf{Det}\,\partial_k^iA=\mathbf{Det}(a_k^i\cap\partial_k^iA)=\mathrm{RDet}(a_k^i\cap\partial_k^iA)\cup\mathbf{LDet}(a_k^i\cap\partial_k^iA)=\mathbf{0},$$

что также даёт равенства

$$\mathrm{RDet}(a_k^i\partial_k^iA)=\mathbf{LDet}(a_k^i\cap\partial_k^iA)=0$$

для всех k и, следовательно, равенства (1) и (2).

Теорема 7.1. *Следующие условия эквивалентны.*

1. *Матрица A вырожденна, и для неё выполняются формулы разложения детерминантов* (1) *и* (2) *по i-й строке.*
2. *В формулах разложения по i-й строке ориентированных полуперманентов матрицы A имеет место почленное равенство булевых слагаемых*

$$a_k^i\cap\overset{+}{\nabla}\partial_k^iA=a_k^i\cap\overset{-}{\nabla}\partial_k^iA$$

для каждого $k=1,\ldots,n$.

3. *Каждая матрица* $a_k^i\cap\partial_k^iA$ $(k=1,\ldots,n)$ *для данного i является вырожденной, т. е.* $\mathbf{Det}(a_k^i\cap\partial_k^iA)=\mathbf{0}$.

Аналогичное утверждение справедливо и для столбца.

Доказательство. Предположим, что выполняется первое условие теоремы, т. е. верны разложения по *i*-й строке (1) и (2) для некоторой вырожденной матрицы *A*. Тогда из

$$\mathrm{RDet}\, A = 0 = \bigcup_{k=1}^{n} \left(a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i A\right)$$

получаем, что

$${}^{i+k}\sigma(\mathrm{RDet})(a_k^i \cap \partial_k^i A) = 0$$

для всех значений *k*. Аналогично из (2) получаем, что

$${}^{i+k}\sigma(\mathbf{LDet})(a_k^i \cap \partial_k^i A) = 0$$

для всех *k*. Следовательно,

$$\mathbf{Det}(a_k^i \cap \partial_k^i A) = 0,$$

или

$$\overset{+}{\nabla}(a_k^i \cap \partial_k^i A) = \overset{-}{\nabla}(a_k^i \cap \partial_k^i A),$$

или

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A,$$

и в формулах разложений ориентированных полуперманентов (4) наблюдаются равенства соответствующих булевых слагаемых.

Пусть каждая матрица $a_k^i \cap \partial_k^i A$ $(k = 1, \ldots, n)$ для данного *i* является вырожденной, т. е. $\mathbf{Det}(a_k^i \cap \partial_k^i A) = 0$.

Тогда

$$\overset{+}{\nabla}(a_k^i \cap \partial_k^i A) = \overset{-}{\nabla}(a_k^i \cap \partial_k^i A),$$

и для некоторой *i*-й строки в формулах разложений полуперманентов матрицы имеют место равенства всех соответствующих булевых слагаемых:

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A.$$

Тогда, с одной стороны, выполняется равенство (4) и, следовательно, матрица *A* вырожденная и $(\mathrm{R})(\mathbf{L})\mathbf{Det}\, A = 0$. С другой стороны, при условии

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A$$

получаем

$$a_k^i \cap {}^{i+k}\sigma(\mathrm{RDet})\partial_k^i A = a_k^i \cap {}^{i+k}\sigma(\mathbf{LDet})\partial_k^i A = a_k^i \cap \mathrm{RDet}\,\partial_k^i A = a_k^i \cap \mathbf{LDet}\,\partial_k^i A = 0$$

для всех номеров *k*. Следовательно, вырожденная матрица *A* удовлетворяет формулам разложения детерминантов (1) и (2) по *i*-й строке.

Как было ранее отмечено, квадратные вырожденные (0,*I*)-матрицы бывают двух типов: внешние и внутренние.

Следующая теорема о разложимости определителей внутренних (0, *I*)-матриц придаёт проблеме комбинаторный характер.

Теорема 7.2. *Матрица с элементами из булевой алгебры* $B_2 = \{0, I\}$ *является ненулевой внутренней и формулы разложения её определителя по строке (или столбцу) верны тогда и только тогда, когда в этой строке (или столбце) существует по крайней мере один ненулевой элемент, через который проходят по крайней мере две диагонали с ненулевыми пересечениями, причём одна чётная и одна нечётная, а все остальные диагонали этой матрицы являются диагоналями с нулевыми пересечениями.*

Доказательство. Как было доказано, если формула разложения по *i*-й строке (3) (или, по лемме 7.1, формулы (1) и (2)) для некоторой внутренней матрицы *A* выполняется, то

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A,$$

где $k = 1, \ldots, n$. Следовательно, может быть два варианта: либо

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A = 0,$$

либо

$$a_k^i \cap \overset{+}{\nabla}\partial_k^i A = a_k^i \cap \overset{-}{\nabla}\partial_k^i A = I.$$

Первое условие даёт, что все диагонали

$$(a_k^i \cap a_1^{\alpha_1}, a_k^i \cap a_2^{\alpha_2}, \ldots, a_k^i \cap a_{k-1}^{\alpha_{k-1}}, a_k^i \cap a_{k+1}^{\alpha_{k+1}}, \ldots, a_k^i \cap a_n^{\alpha_n})$$

в матрице $a_k^i \cap \partial_k^i A$ являются диагоналями с нулевым пересечением. Это значит, что

$$a_1^{\alpha_1} \cap \ldots \cap a_k^i \cap \ldots \cap a_n^{\alpha_n} = 0$$

и все проходящие через элемент a_k^i диагонали в матрице *A* являются диагоналями с нулевыми пересечениями.

Такие же рассуждения позволяют заключить во втором случае, что через элемент $a_k^i = I$ матрицы *A* проходят по крайней мере две диагонали, причём одна чётная, а другая нечётная, с ненулевыми пересечениями.

То, что в строке (или столбце) внутренней матрицы *A* обязательно существует ненулевой элемент, через который проходят две диагонали с ненулевыми пересечениями и противоположной чётности, следует из определяющего внутреннюю матрицу равенства

$$\triangle A = \overset{+}{\nabla} A = \overset{-}{\nabla} A = I.$$

Докажем утверждение в обратную сторону. Пусть в *i*-й строке (или столбце) матрицы *A* существуют элементы $a_k^i = I$, $k \in K \subseteq \{1, \ldots, n\}$, через каждый из которых проходят по крайней мере две диагонали вида $(a_1^{\alpha_1}, \ldots, a_k^i, \ldots, a_n^{\alpha_n})$ с ненулевыми

пересечениями, причём одна чётная и одна нечётная. Следовательно, $\triangle A = \overset{+}{\nabla} A = \overset{-}{\nabla} A = I$, и матрица A является внутренней. С другой стороны, это влечёт существование двух диагоналей, причём одной чётной и другой нечётной, с ненулевым пересечением вида

$$(a_k^i \cap a_1^{\alpha_1}, a_k^i \cap a_2^{\alpha_2}, \ldots, a_k^i \cap a_{k-1}^{\alpha_{k-1}}, a_k^i \cap a_{k+1}^{\alpha_{k+1}}, \ldots, a_k^i \cap a_n^{\alpha_n})$$

в матрице $a_k^i \cap \partial_k^i A$. Следовательно,

$$a_k^i \cap \overset{+}{\nabla} \partial_k^i A = a_k^i \cap \overset{-}{\nabla} \partial_k^i A = I$$

для всех $k \in K \subseteq \{1, \ldots, n\}$.

Ясно, что если все диагонали в матрице A, проходящие через каждый элемент $a_k^i = I$ при $k \in \{1, \ldots, n\} \setminus K$ (тем более через элементы $a_k^i = 0$), являются диагоналями с нулевыми пересечениями, то снова получаем равенство

$$a_k^i \cap \overset{+}{\nabla} \partial_k^i A = a_k^i \cap \overset{-}{\nabla} \partial_k^i A = 0.$$

Выполнено условие 2 теоремы 7.1, следовательно, формула (3) для i-й строки (или столбца) верна.

Следствие. *Для внутренней ненулевой (0,I)-матрицы выполняются формулы разложения* по любой строке (или столбцу) *тогда и только тогда, когда в каждой строке (или столбце) существуют ненулевые элементы, через которые проходят по крайней мере две диагонали с ненулевыми пересечениями, причём противоположной чётности, а все остальные диагонали являются диагоналями с нулевыми пересечениями.*

Условия этого следствия выполняются для матриц из следующего примера.

Пример 7.2. Любая внутренняя (3 × 3)-матрица над $B_2 = \{0, I\}$ может быть представлена одним из следующих типов:

$$A_1 = \begin{pmatrix} I & I & a_3^1 \\ I & I & a_3^2 \\ a_1^3 & a_2^3 & I \end{pmatrix}, \quad A_2 = \begin{pmatrix} I & a_2^1 & a_3^1 \\ a_1^2 & I & I \\ a_1^3 & I & I \end{pmatrix},$$

$$A_3 = \begin{pmatrix} I & a_2^1 & I \\ a_1^2 & I & a_3^2 \\ I & a_2^3 & I \end{pmatrix}, \quad A_4 = \begin{pmatrix} a_1^1 & I & I \\ a_1^2 & I & I \\ I & a_2^3 & a_3^3 \end{pmatrix}, \quad A_5 = \begin{pmatrix} a_1^1 & a_2^1 & I \\ I & I & a_3^2 \\ I & I & a_3^3 \end{pmatrix}.$$

Рассмотрим матрицу A_1. В ней через элемент $a_1^1 = I$ уже проходит одна чётная диагональ (a_1^1, a_2^2, a_3^3), $a_1^1 \cap a_2^2 \cap a_3^3 = I$. Следовательно, для выполнения формул разложения (1)—(3) для первой строки нечётная диагональ (a_1^1, a_2^3, a_3^2), проходящая через этот же элемент

$a_1^1 = I$, должна быть с ненулевым пересечением, т. е. $a_1^1 \cap a_2^3 \cap a_3^2 = I$. Поэтому $a_2^3 = a_3^2 = I$.

Рассмотрим теперь элемент $a_2^2 = I$ матрицы A_1. Через него уже проходит одна чётная диагональ (a_1^1, a_2^2, a_3^3), $a_1^1 \cap a_2^2 \cap a_3^3 = I$. Чтобы формулы разложения (1)—(3) выполнялись для второй строки, нужно, чтобы $a_1^3 \cap a_2^2 \cap a_3^1 = I$ для нечётной диагонали (a_1^3, a_2^2, a_3^1), проходящей через элемент $a_2^2 = I$. Поэтому

$a_1^3 = a_3^1 = I$.

Таким образом, для внутренних (3 × 3)-матриц над $B_2 = \{0, I\}$ формулы разложения (1)—(3) выполняются для двух строк (и, следовательно, для любой строки или столбца) только для матрицы

$$A_1 = \begin{pmatrix} I & I & I \\ I & I & I \\ I & I & I \end{pmatrix} = J.$$

Можно привести пример внутренних ($n \times n$)-матриц ($n \geq 4$), отличных от матрицы J, для которых формулы разложения определителей по любой строке выполняются. Например, матрица

$$\begin{pmatrix} I & I & 0 & 0 \\ I & I & 0 & 0 \\ 0 & 0 & I & I \\ 0 & 0 & I & I \end{pmatrix}$$

внутренняя, так как $\overset{\pm}{\nabla} A = I$. Разложения (1) и (2) по любой строке или столбцу выполняются. Так,

$$\left(I \cap (\mathrm{R})(\mathrm{L})\mathrm{Det} \begin{pmatrix} I & 0 & 0 \\ 0 & I & I \\ 0 & I & I \end{pmatrix}\right) \cup \left(I \cap (\mathrm{L})(\mathrm{R})\mathrm{Det} \begin{pmatrix} I & 0 & 0 \\ 0 & I & I \\ 0 & I & I \end{pmatrix}\right) \cup 0 \cup 0 = 0$$

есть разложение по первым двум строкам (или столбцам).

6.4.8. Разложения Лапласа для произвольных квадратных булевых матриц

Теорема 8.1. *Пусть A — квадратная матрица с элементами из произвольной булевой алгебры. Формулы* (1) *и* (2) *разложения по i-й строке (столбцу) ориентированных детерминантов матрицы A выполняются тогда и только тогда, когда формулы разложения по i-й строке (столбцу)* (1) *и* (2) *выполняются для ориентированных детерминантов её внутренней части A.*

Доказательство. Правая часть формулы (1) и разложение $A = \check{A} \cup \hat{A} \cup \overset{+}{A} \cup \overline{A}$ дают

$$\bigcup_{k=1}^{n} (a_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i A) = \bigcup_{k=1}^{n} \{a_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i [\check{A} \cup \hat{A} \cup \overset{+}{A} \cup \overline{A}]\} =$$

$$= \bigcup_{k=1}^{n} \{a_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i [((\mathrm{Per}\, A)' \cap A) \cup (\triangle A \cap A) \cup$$

$$\cup (\mathrm{RDet}\, A \cap A) \cup (\mathbf{LDet}\, A \cap A)]\} =$$

$$= \bigcup_{k=1}^{n} \{a_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})[((\mathrm{Per}\, A)' \cap \partial_k^i A) \cup (\triangle A \cap \partial_k^i A) \cup$$

$$\cup (\mathrm{RDet}\, A \cap \partial_k^i A) \cup (\mathbf{LDet}\, A \cap \partial_k^i A)]\}.$$

Учитывая теорему 2.1 и то, что коэффициенты $(\mathrm{Per}\, A)'$, $\triangle A$, $\mathrm{RDet}\, A$, $\mathbf{LDet}\, A$ попарно не пересекаются, получаем равенство

$$\bigcup_{k=1}^{n} (a_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i A) =$$

$$= \bigcup_{k=1}^{n} (\check{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \check{A}) \cup \bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \hat{A}) \cup$$

$$\cup \bigcup_{k=1}^{n} (\overset{+}{a}{}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overset{+}{A}) \cup \bigcup_{k=1}^{n} (\overline{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overline{A}). \qquad (5)$$

Здесь посредством $\check{a}_k^i$, $\hat{a}_k^i$, $\overset{+}{a}{}_k^i$, $\overline{a}_k^i$ обозначены элементы матриц $\check{A}$, $\hat{A}$, $\overset{+}{A}$, $\overline{A}$ соответственно. Следовательно,

$$\check{a}_k^i = (\mathrm{Per}\, A)' \cap a_k^i \subseteq (\mathrm{Per}\, A)', \qquad \hat{a}_k^i = \triangle A \cap a_k^i \subseteq \triangle A,$$

$$\overset{+}{a}{}_k^i = \mathrm{RDet}\, A \cap a_k^i \subseteq \mathrm{RDet}\, A, \qquad \overline{a}_k^i = \mathbf{LDet}\, A \cap a_k^i \subseteq \mathbf{LDet}\, A.$$

Предположим, что формулы (1) и (2) выполняются для матрицы A. Заметим, что в равенстве (5) стоят слагаемые, удовлетворяющие включениям

$$\bigcup_{k=1}^{n} (\check{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \check{A}) \subseteq (\mathrm{Per}\, A)', \qquad \bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \hat{A}) \subseteq \triangle A,$$

$$\bigcup_{k=1}^{n} (\overset{+}{a}{}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overset{+}{A}) \subseteq \mathrm{RDet}\, A, \qquad \bigcup_{k=1}^{n} (\overline{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overline{A}) \subseteq \mathbf{LDet}\, A,$$

и $(\mathrm{Per}\, A)'$, $\triangle A$, $\mathrm{RDet}\, A$, $\mathbf{LDet}\, A$ попарно не пересекаются. Если выполняется (1), то все эти слагаемые должны содержаться в $\mathrm{RDet}\, A$. Это даёт равенства

$$\bigcup_{k=1}^{n} (\check{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \check{A}) = \bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \hat{A}) = \bigcup_{k=1}^{n} (\overline{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overline{A}) = 0.$$

Подобным образом из (2) получаем равенства

$$\bigcup_{k=1}^{n} (\check{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathbf{LDet})\partial_k^i \check{A}) = \bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathbf{LDet})\partial_k^i \hat{A}) = \bigcup_{k=1}^{n} (\overset{+}{a}{}_k^i \cap {}^{i}\sigma^{i+k}(\mathbf{LDet})\partial_k^i \overset{+}{A}) = 0.$$

Тогда формулы (1) и (2) выполняются для внутренней части $\hat{A}$ матрицы A. Их можно записать в виде

$$\bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \hat{A}) = 0 = \mathrm{RDet}\, \hat{A}, \qquad (1')$$

$$\bigcup_{k=1}^{n} (\hat{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathbf{LDet})\partial_k^i \hat{A}) = 0 = \mathbf{LDet}\, \hat{A}. \qquad (2')$$

Предположим теперь, что формулы разложения по i-й строке (1'), (2') выполняются для детерминантов внутренней части $\hat{A}$ матрицы A.

Учитывая теорему 6.1 и то, что $\check{A}$, $\overset{+}{A}$, $\overline{A}$ являются матрицами с нулевой внутренностью, получаем равенства

$$\bigcup_{k=1}^{n} (\check{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \check{A}) = \mathrm{RDet}\, \check{A} = 0,$$

$$\bigcup_{k=1}^{n} (\overset{+}{a}{}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overset{+}{A}) = \mathrm{RDet}\, \overset{+}{A} = \mathrm{RDet}\, A,$$

$$\bigcup_{k=1}^{n} (\overline{a}_k^i \cap {}^{i}\sigma^{i+k}(\mathrm{RDet})\partial_k^i \overline{A}) = \mathrm{RDet}\, \overline{A} = 0.$$

Тогда (5) даёт равенство (1) для матрицы A. Аналогичными рассуждениями устанавливается формула (2).

Очевидно, таким же образом можно доказать, что формула разложения детерминанта (3) матрицы A выполняются тогда и только тогда, когда она выполняются для детерминанта внутренней части A матрицы A.

6.4.9. Обратимые булевы матрицы и разложения детерминантов

Определение 9.1. Определим *произведение* $C = A \sqcap B$ квадратных $(n \times n)$-матриц A и B как матрицу того же размера, элементы которой c_j^i вычисляются по формуле

$$c_j^i = \bigcup_{t=1}^{n} (a_t^i \cap b_j^t).$$

Определение 9.2. Матрица A *обратима,* если существует матрица A^{-1} и выполнены равенства $A \sqcap A^{-1} = A^{-1} \sqcap A = E$. Матрицу A^{-1} называют *обратной* матрицей для A, а $E = (\delta_j^i)$, где δ_j^i есть I, если $i = j$, и 0, если $i \neq j$, — *единичной* матрицей.

Общий вид обратимой булевой матрицы и различные условия обратимости хорошо известны.

Определение 9.3. *Положительной* и *отрицательной присоединённой* для $(n \times n)$-матрицы A назовём $(n \times n)$- матрицы $\overset{+}{\operatorname{adj}} A$ и $\overset{-}{\operatorname{adj}} A$, элементы которых

$$(\overset{+}{\operatorname{adj}} A)_j^i \text{ и } (\overset{-}{\operatorname{adj}} A)_j^i$$

определяются равенствами

$$(\overset{+}{\operatorname{adj}} A)_j^i = \overset{i+j}{\sigma}(\overset{+}{\nabla})\partial_i^j A, \quad (\overset{-}{\operatorname{adj}} A)_j^i = \overset{i+j}{\sigma}(\overset{-}{\nabla})\partial_i^j A$$

соответственно. Таким образом,

$$(\overset{+}{\operatorname{adj}} A)_j^i = \overset{+}{\nabla}\partial_i^j A, \quad (\overset{-}{\operatorname{adj}} A)_j^i = \overset{-}{\nabla}\partial_i^j A,$$

если $i + j$ чётно, и

$$(\overset{+}{\operatorname{adj}} A)_j^i = \overset{-}{\nabla}\partial_i^j A, \quad (\overset{-}{\operatorname{adj}} A)_j^i = \overset{+}{\nabla}\partial_i^j A,$$

если $i + j$ нечётно.

В литературе установлена связь обратных и присоединённых булевых матриц. Главный результат установленной связи сформулирован ниже и приводится без доказательства.

Лемма 9.1. *Для булевой квадратной матрицы A обратная матрица A^{-1} существует тогда и только тогда, когда*

$\operatorname{Det} A = I$ *и* $(A \sqcap \overset{+}{\operatorname{adj}} A) \cap (A \sqcap \overset{-}{\operatorname{adj}} A) = \Theta$ *(или* $(\overset{+}{\operatorname{adj}} A \sqcap A) \cap (\overset{-}{\operatorname{adj}} A \sqcap A) = \Theta$*). Кроме того, обратная матрица равна булевой сумме её присоединённых матриц, т. е.*

$A^{-1} = \overset{+}{\operatorname{adj}} A \cup \overset{-}{\operatorname{adj}} A$, *причём* $\overset{+}{\operatorname{adj}} A \cap \overset{-}{\operatorname{adj}} A = \Theta$.

Определение 9.4. Пусть $i \neq j$. Тогда равенства

$$\bigcup_{k=1}^{n} (a_k^i \cap \overset{j+k}{\sigma}(\operatorname{RDet})\partial_k^j A) = 0, \quad \bigcup_{k=1}^{n} (a_k^i \cap \overset{j+k}{\sigma}(\operatorname{LDet})\partial_k^j A) = 0$$

и эквивалентное им равенство

$$\bigcup_{k=1}^{n} (a_k^i \cap \operatorname{Det} \partial_k^j A) = 0 \qquad (6)$$

назовём формулами *ложного разложения* определителей для строк i и j. Формулы ложного разложения определителей для пары столбцов можно выписать аналогичным образом.

Заметим, что формулы ложных разложений определителей квадратных булевых матриц в общем случае не выполняются.

Следующая теорема показывает связь проблемы обратимости булевых матриц и возможности разложения их определителей.

Теорема 9.1. *Для квадратной булевой матрицы A обратная матрица существует тогда и только тогда, когда* $\operatorname{Det} A = I$ *и для её определителя выполняются формулы разложений и формулы ложных разложений для любых её строк (или столбцов).*

Доказательство. Очевидно, что из формул (3), (6) и равенства $\operatorname{Det} A = I$ следует существование обратной матрицы A^{-1}. Элементами такой матрицы являются $(A^{-1})_j^i = \operatorname{Det} \partial_i^j A$ $(i, j = 1, \ldots, n)$.

Покажем теперь, что существование обратной матрицы A^{-1} влечёт выполнение (3), (6) и равенства $\operatorname{Det} A = I$.

Из леммы 9.1 сразу получаем $\operatorname{Det} A = I$.

Обозначим через $A^{\langle [i] \Downarrow [j] \rangle}$ и $A_{\langle [i] \Rightarrow [j] \rangle}$ матрицы, полученные из матрицы A заменой j-й строки (или j-го столбца) строкой (соответственно столбцом) этой же матрицы с номером i. Понятно, что

$$A^{\langle [j] \Downarrow [j] \rangle} = A \text{ и } A_{\langle [j] \Rightarrow [j] \rangle} = A.$$

Равенство

$$(A \sqcap \overset{+}{\operatorname{adj}} A) \cap (A \sqcap \overset{-}{\operatorname{adj}} A) = \Theta$$

из леммы 9.1 даёт

$$\left[\bigcup_{k=1}^{n} (a_k^i \cap (\overset{+}{\operatorname{adj}} A)_j^k)\right] \cap \left[\bigcup_{k=1}^{n} (a_k^i \cap (\overset{-}{\operatorname{adj}} A)_j^k)\right] = 0$$

для всех $i,j = 1,...,n$. Тогда, учитывая определение 9.3 для присоединённых матриц, получаем

$$\left[\bigcup_{k=1}^{n} (a_k^i \cap \overset{i+j}{\sigma}(\overset{+}{\nabla})\partial_k^j A)\right] \cap \left[\bigcup_{k=1}^{n} (a_k^i \cap \overset{i+j}{\sigma}(\overset{-}{\nabla})\partial_k^j A)\right] = 0$$

для всех $i,j = 1,\ldots,n$. Формулы разложения Лапласа полуперманентов любых квадратных булевых матриц всегда выполняются. Тогда последние равенства перепишутся как

$$\left[\overset{+}{\nabla} A^{\langle [i]\Downarrow[j]\rangle}\right] \cap \left[\overset{-}{\nabla} A^{\langle [i]\Downarrow[j]\rangle}\right] = \triangle A^{\langle [i]\Downarrow[j]\rangle} = 0$$

(или $\triangle A_{\langle [i]\Rightarrow[j]\rangle} = 0$) для всех $i, j = 1,\ldots,n$.

Таким образом, из существования обратной матрицы A^{-1} следует, что все матрицы $A^{\langle [i]\Downarrow[j]\rangle}$ (или $A_{\langle [i]\Rightarrow[j]\rangle}$), $i,j = 1,\ldots,n$, являются матрицами с нулевой внутренностью. Следовательно, для них выполняются формулы разложения определителей (теорема 6.1) по любой строке (или столбцу). Остаётся заметить, что в случае $i = j$ разложения определителя $\mathrm{Det}\, A^{\langle [i]\Downarrow[j]\rangle}$ по *j-й* строке есть формулы (3), а в случае $i \neq j$ они становятся формулами ложных разложений (6).

Замечание. Условия $A^{-1} = \overset{+}{\mathrm{adj}}\, A \cup \overset{-}{\mathrm{adj}}\, A$, $\overset{+}{\mathrm{adj}}\, A \cap \overset{-}{\mathrm{adj}}\, A = \Theta$ из леммы 9.1 можно записать в виде

$$(A^{-1})_i^j = \overset{+}{\nabla}\partial_j^i A \cup \overset{-}{\nabla}\partial_j^i A = \left[\overset{+}{\nabla}\partial_j^i A \setminus \overset{-}{\nabla}\partial_j^i A\right] \cup \left[\overset{-}{\nabla}\partial_j^i A \setminus \overset{+}{\nabla}\partial_j^i A\right].$$

Мы получаем некий аналог известных выражений для элементов обратных матриц:

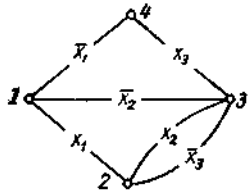
$$(A^{-1})_i^j = \mathrm{Per}\,\partial_j^i A = \mathrm{Det}\,\partial_j^i A \quad (i,j = 1,\ldots,n).$$

Очевидно, обратимые матрицы не являются единственными, для определителей которых выполняются формулы разложений Лапласа и формулы ложных разложений для любых её строк или столбцов. В частности, они выполняются для матрицы $\alpha \cap A$, где A — обратимая булева матрица, а $\alpha \in B$, причём $\alpha \neq I$.

6.5. Исключение (анализ) и введение (синтез) узлов

При анализе контактной схемы с помощью булевых матриц сначала записывается матрица непосредственных связей *P,* а затем путем возведения ее в соответствующую степень получается матрица полных связей *Q.* Элементы матрицы Q и представляют собой булевы функции данной контактной схемы между парами узлов с номерами *i* и *j*. Однако такой способ в большинстве случаев не является рациональным, так как обычно представляют интерес только некоторые из функций q_{ij} между внешними узлами (полюсами) схемы. Поэтому имеет смысл предварительно исключить внутренние узлы и таким образом уменьшить порядок матрицы *P,* прежде чем возводить ее в

требуемую степень. При исключении s-го узла в матрице непосредственных связей вычеркиваются *s*-я строка и s-й столбец и каждый ее элемент p_{ij} заменяется элементом $p_{ij} \vee p_{is}p_{sj}$. Член $p_{is}p_{sj}$ учитывает путь между узлами *i* и *j* через узел s, который действует параллельно с непосредственной связью p_{ij}. В результате исключения узла матрица *P* преобразуется к матрице P_s на единицу меньшего порядка, которая представляет собой матрицу непосредственных связей относительно неисключенной совокупности узлов. Пусть, например, в схеме рис. 7 требуется определить булевы функции между узлами 1, *2* и *3*.



Ркс. 7. Контактная схема к примеру

Матрицы *P* и P_4 имеют вид:

$$P = \begin{bmatrix} 1 & x_1 & \bar{x}_2 & \bar{x}_1 \\ x_1 & 1 & x_2 \vee x_3 & 0 \\ \bar{x}_2 & x_2 \vee \bar{x}_3 & 1 & x_3 \\ \bar{x}_1 & 0 & x_3 & 1 \end{bmatrix}; \quad P_4 = \begin{bmatrix} 1 & x_1 & \bar{x}_2 \vee \bar{x}_1 x_3 \\ x_1 & 1 & x_2 \vee \bar{x}_3 \\ \bar{x}_2 \vee x_1 x_3 & x_2 \vee \bar{x}_3 & 1 \end{bmatrix}.$$

Определив P^2_4 после преобразований, получим матрицу полных связей относительно узлов 1, *2* и *3,* называемую *матрицей выходов:*

$$F = \begin{bmatrix} 1 & x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_2) & \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3) \\ x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_2) & 1 & x_1\bar{x}_2 \vee x_2 \vee x_3 \\ \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3) & x_1\bar{x}_2 \vee x_2 \vee x_3 & 1 \end{bmatrix}.$$

Элементы этой матрицы являются функциями выходов: $f_{12} = x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_2)$; $f_{13} = \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3)$; $f_{23} = x_1\bar{x}_2 \vee x_2 \vee x_3$.

При синтезе контактных схем задаются функции для внешних узлов (полюсов), которые определяют матрицу выходов. Необходимое и достаточное условие непротиворечивости этих функций состоит в том,

что матрица выходов должна быть *устойчивой,* т. е. удовлетворять равенству $F = F^2$.

Структуру контактной схемы, реализующей заданную непротиворечивую совокупность функций, можно получить из матрицы F путем ее последовательного расширения, соответствующего *операции введения узла.* Эта операция обратна исключению узла и приводит к матрице F_s, порядок которой на единицу выше, а элементы таковы, что при исключении узла s снова получим матрицу F. Последовательным применением операции введения узла исходная матрица расширяется и преобразуется к виду, при котором элементы представляют собой константы 0 или 1, переменные, их отрицания или элементарные конъюнкции переменных. Тогда полученную матрицу можно рассматривать как матрицу непосредственных связей, на основе которой легко построить соответствующую контактную схему. При этом элементарные конъюнкции реализуются последовательными соединениями соответствующих контактов.

Операция введения неоднозначна, поэтому можно получать различные схемы, удовлетворяющие заданным функциям. Выбор наилучшего пути преобразования матрицы F к матрице непосредственных связей P, определяющей вид контактной схемы, в значительной степени зависит от искусства инженера.

Пусть требуется построить контактную схему со следующими функциями; $f_{12} = \bar{x}_1\bar{x}_2 \vee x_1x_3$; $f_{13} = \bar{x}_3(x_2 \vee x_1x_4)$; $f_{23} = 0$. Матрица выходов имеет вид:

$$F = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 \vee x_1x_3 & \bar{x}_3(x_2 \vee x_1x_4) \\ \bar{x}_1\bar{x}_2 \vee x_1x_3 & 1 & 0 \\ \bar{x}_3(x_2 \vee x_1x_4) & 0 & 1 \end{bmatrix}.$$

Элементы этой матрицы можно рассматривать как результат исключения узла 4, который мы должны ввести, т. е. $\bar{x}_1\bar{x}_2 \vee x_1x_3 = f'_{12} \vee f'_{14}f'_{42}$, $\bar{x}_3(x_2 \vee x_1x_4) = f'_{13} \vee f'_{14}f'_{43}$ и $0 = f'_{23} \vee f'_{24}f'_{43}$. Полагая $f'_{14} = x_2 \vee x_1x_4$ и $f'_{43} = \bar{x}_3$ (возможны и другие варианты), имеем $f'_{13} = f'_{42} = f'_{23} = f'_{24} = 0$ и $f'_{12} = \bar{x}_1\bar{x}_2 \vee x_1x_3$.

Таким образом, в результате введения узла 4 имеем матрицу

$$F_{(4)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 \vee x_1x_3 & 0 & x_2 \vee x_1x_4 \\ \bar{x}_1\bar{x}_2 \vee x_1x_3 & 1 & 0 & 0 \\ 0 & 0 & 1 & \bar{x}_3 \\ x_2 \vee x_1x_4 & 0 & \bar{x}_3 & 1 \end{bmatrix}.$$

Продолжая аналогично, можно записать соотношения для элементов матрицы $F_{(4,5)}$, соответствующей введению узла 5:

$$\bar{x}_1\bar{x}_2 \vee x_1x_3 = f''_{12} \vee f''_{15}f''_{52};\ 0 = f''_{13} \vee f''_{15}f''_{53};\ x_2 \vee x_1x_4 = f''_{14} \vee f''_{15}f''_{54};$$
$$0 = f''_{23} \vee f''_{25}f''_{53};\ 0 = f''_{24} \vee f''_{25}f''_{54};\ \bar{x}_3 = f''_{31} \vee f''_{35}f''_{51}.$$

Если принять

$f''_{15} = x_1$,

То необходимо продолжить

$f''_{12} = \bar{x}_1\bar{x}_2$; $f''_{52} = x_3$; $f''_{13} = f''_{53} = 0$; $f''_{14} = x_2$;

$f''_{51} = x_4$; $f''_{23} = f''_{25} = f''_{24} = 0$.

В результате приходим к матрице, которую можно рассматривать как матрицу непосредственных связей P синтезируемой схемы:

$$P = F_{(4,5)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 & 0 & x_2 & x_1 \\ \bar{x}_1\bar{x}_2 & 1 & 0 & 0 & x_3 \\ 0 & 0 & 1 & \bar{x}_3 & 0 \\ x_2 & 0 & \bar{x}_3 & 1 & x_4 \\ x_1 & x_3 & 0 & x_4 & 1 \end{bmatrix}.$$

Схема, соответствущая этой матрице, показана на рис. 8

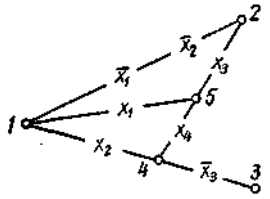


Рис.8. Схема, построенная по матрице непосредственных связей.

6.6. Вентильные схемы.

До сих пор предполагалось, что контакты обладают двусторонней проводимостью, т. е. в открытом состоянии они пропускают сигналы как в прямом, так и в обратном направлениях. Таковы, например, контакты электромагнитных реле. Однако при использовании электронных ключей, например управляемых диодов, проводимость в прямом направлении настолько превышает проводимость в обратном направлении, что практически можно считать контакты односторонними, т. е. пропускающими сигналы только в прямом направлении. Схемы с односторонними контактами называют *вентильными схемами.*

На вентильных схемах, как и ранее, изображаются только соединения контактов, а управляющие цепи обычно опускаются. При этом предполагается, что управление осуществляется как сигналами, соответствующим переменными $x_1, x_2, \ldots, x_n$,так и их отрицаниям $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$, что отмечается на схеме одним из символов x_i или $\bar{x}_i$ - для каждого контакта. Кроме того, в вентильных схемах обычно имеет место естественное разделение сигналов: если к узлу схемы одновременно поступают несколько сигналов, то результирующий сигнал в этом узле действует как их дизъюнкция. Направления прохождении сигналов обозначаются на схемах стрелками, относящимися к соответствующим контактам. Пример вентильной схемы показан на рис.9.

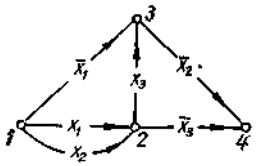


Рис. 9. Вентильная схема.

Булевы матрицы вентильных схем в общем случае несимметричны. Так, для приведенной схемы имеем:

$$P = \begin{bmatrix} 1 & x_1 \vee x_2 & \bar{x}_1 & 0 \\ 0 & 1 & x_3 & \bar{x}_3 \\ 0 & 0 & 1 & \bar{x}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Q = \begin{bmatrix} 1 & x_1 \vee x_2 & \bar{x}_1 \vee x_3 & \bar{x}_2 \vee \bar{x}_3 \\ 0 & 1 & x_3 & \bar{x}_2 \vee x_3 \\ 0 & 0 & 1 & \bar{x}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

При этом в соответствии $Q = P^3$. Матрицу Q можно также записать непосредственно из вентильной схемы, учитывая для ее элементов q_{ij} все пути от i-го узла к j-му узлу по направлению стрелок.

Так, $q_{12} = x_1 \vee x_2$; $q_{13} = \bar{x}_1 \vee (x_1 \vee x_2)x_3 = \bar{x}_1 \vee x_1x_3 \vee x_2x_3 = \bar{x}_1 \vee x_3 \vee x_2x_3 = \bar{x}_1 \vee x_3$; $q_{14} = \bar{x}_1\bar{x}_2 \vee (x_1 \vee x_2)(\bar{x}_3 \vee x_3\bar{x}_2) = \bar{x}_2 \vee \bar{x}_3$ и т. д. Булева функция для любого выхода может быть определена также последовательным исключением узлов, кроме входного и выходного.

Синтез вентильных схем осуществляется аналогично изложенному в (6.5), причем в исходной матрице выходов все функции, кроме заданных, обычно полагаются тождественно равными нулю. Пусть, например, $f_{12} = x_1x_2 \vee \bar{x}_1\bar{x}_3$ и $f_{13} = x_1\bar{x}_3 \vee \bar{x}_1x_2$. Матрица выходов ее расширения имеют вид:

$$F = \begin{bmatrix} 1 & x_1x_2 \vee \bar{x}_1\bar{x}_3 & x_1\bar{x}_3 \vee \bar{x}_1x_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad F_{(4)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_3 & \bar{x}_1x_2 & x_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & x_2 & \bar{x}_3 & 1 \end{bmatrix};$$

$$F_{(4,5)} = \begin{bmatrix} 1 & 0 & 0 & x_1 & \bar{x}_1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & x_2 & \bar{x}_3 & 1 & 0 \\ 0 & \bar{x}_3 & x_2 & 0 & 1 \end{bmatrix}.$$

Схемы, соответствующие $F_{(4)}$ и $F_{(4,5)}$ показаны на рис. 204. Как видно, вторая схема (рис. 204, б) содержит на один контакт меньше, чем первая (рис. 204, *а).*

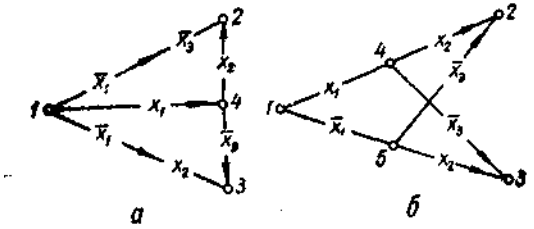


Рис. 10. Схемы, реализующие функции $f_{12} = x_1x_3 \vee \bar{x}_1x_3$ и $f_{13} = x_1\bar{x}_3 \vee x_1x_2$:

а- с четырьмя узлами; *б* - с пятью узлами.

Любой, самый примитивный компьютер – сложнейшее техническое устройство. Но даже такое сложное устройство, как и все в природе и в

технике, состоит из простейших элементов. Любой компьютер, точнее, любой его электронный логический блок состоит из десятков и сотен тысяч так называемых вентилей (логических устройств, базовых логических схем), объединяемых по правилам и законам (аксиомам) алгебры вентилей в схемы, модули.

Логический вентиль (далее – просто вентиль) – это своего рода атом, из которого состоят электронные узлы ЭВМ. Он работает по принципу крана (отсюда и название), открывая или закрывая путь сигналам.

Логические схемы предназначены для реализации различных функций алгебры логики и реализуются с помощью трех базовых логических элементов (вентилей, логических схем или так называемых переключательных схем). Они воспроизводят функции полупроводниковых схем.

Работу вентильных, логических схем мы, как и принято, будем рассматривать в двоичной системе и на математическом, логическом уровне, не затрагивая технические аспекты (аспекты микроэлектроники, системотехники, хотя они и очень важны в технической информатике).

Логические функции отрицания, дизъюнкции и конъюнкции реализуют, соответственно, логические схемы, называемые инвертором, дизъюнктором и конъюнктором.

Логическая функция "инверсия", или отрицание, реализуется логической схемой (вентилем), называемой инвертор.

Принцип его работы можно условно описать следующим образом: если, например, "0" или "ложь" отождествить с тем, что на вход этого устройства скачкообразно поступило напряжение в 0 вольт, то на выходе получается 1 или "истина", которую можно также отождествить с тем, что на выходе снимается напряжение в 1 вольт.

Аналогично, если предположить, что на входе инвертора будет напряжение в 1 вольт ("истина"), то на выходе инвертора будет сниматься 0 вольт, то есть "ложь" (схемы на рисунках 11 а, б).

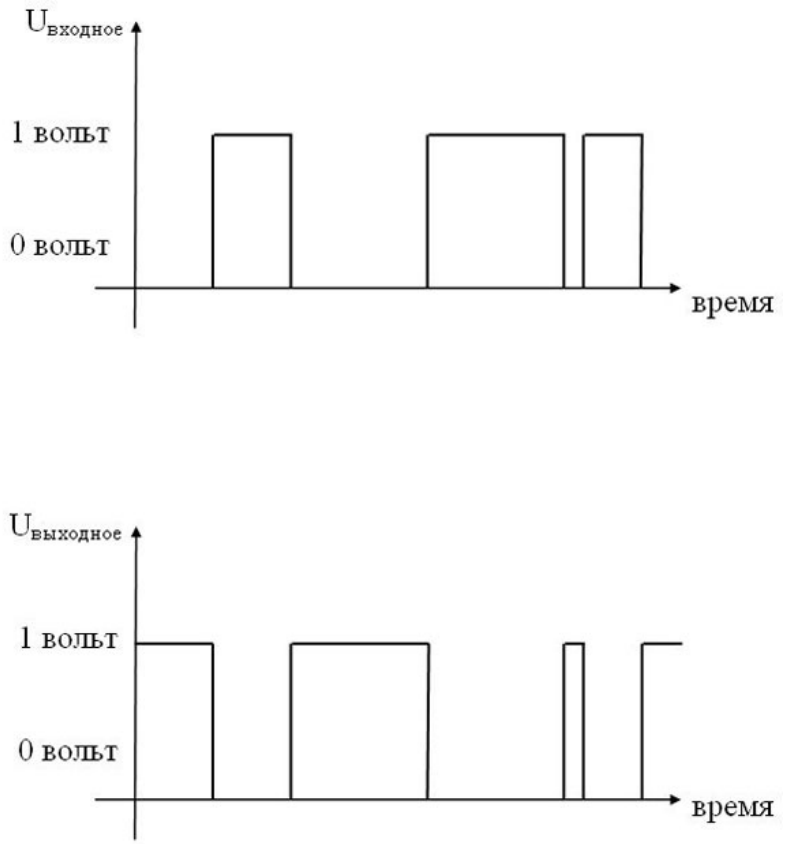


Рис. 11. Принцип работы инвертора

Функцию отрицания можно условно отождествить с электрической схемой соединения в цепи с лампочкой (рис. 12), в которой замкнутая цепь соответствует 1 ("истина") или x = 1, а размыкание цепи соответствует 0 ("ложь") или x = 0.

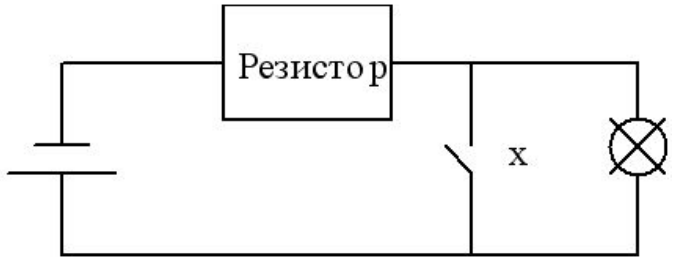


Рис. 12. Электрический аналог схемы инвертора

Дизъюнкцию $x \vee y$ реализует логическое устройство (вентиль) называемое **дизъюнктор** (рис. 13 а, б, в):

Рис. 13a.

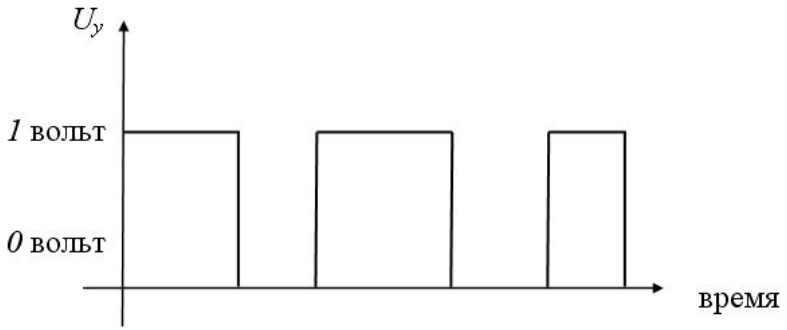


Рис. 1.3b.

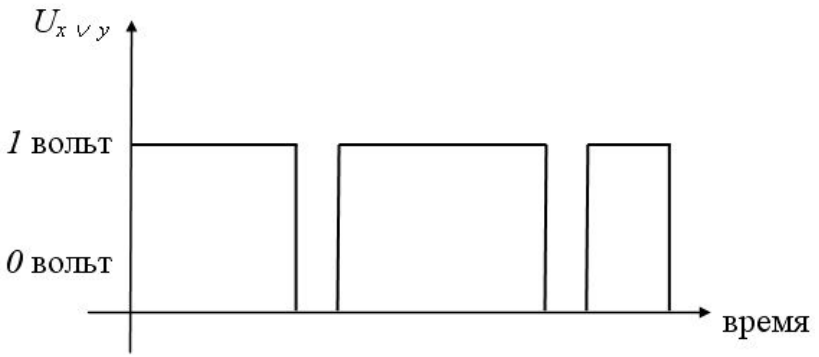


Рис. 13c. Принцип работы дизъюнктора

Дизъюнктор условно изображается схематически электрической цепью вида (рис. 14)

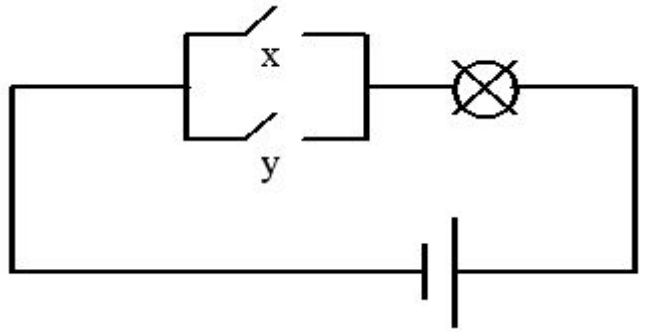


Рис. 14. Электрический аналог схемы дизъюнктора

Конъюнкцию $x \wedge y$ реализует логическая схема (вентиль), называемая конъюнктором (рис. 15 а, б, в):

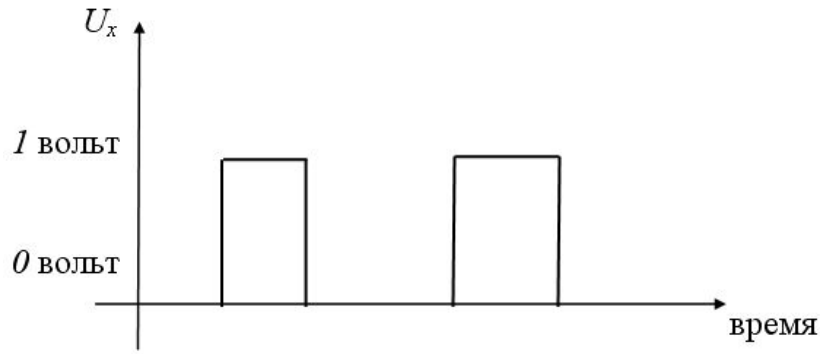


Рис. 15a.

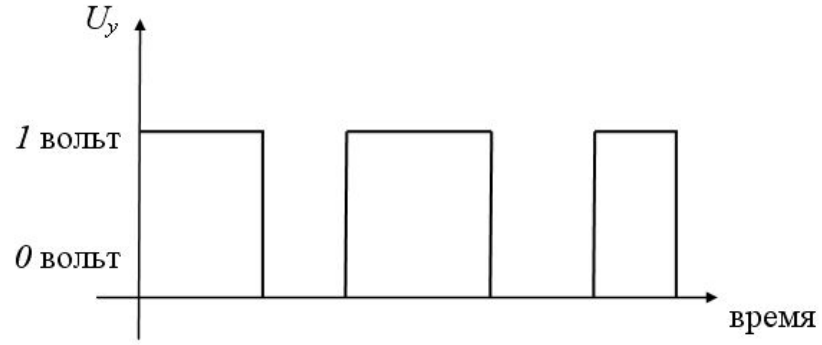


Рис. 15b.

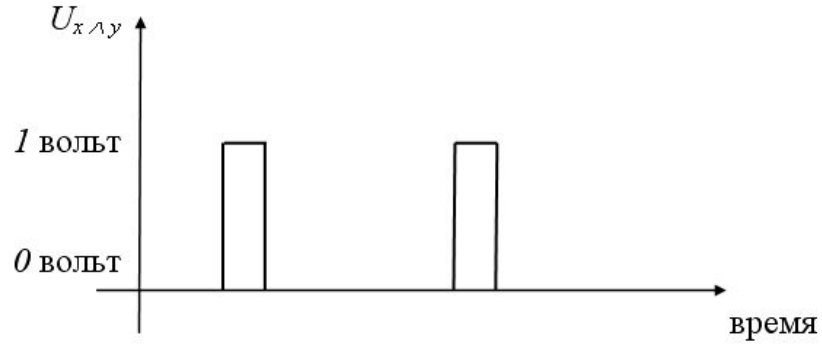


Рис. 15c. Принцип работы конъюнктора

Конъюнктор можно условно изобразить схематически электрической цепью вида (рис. 16)

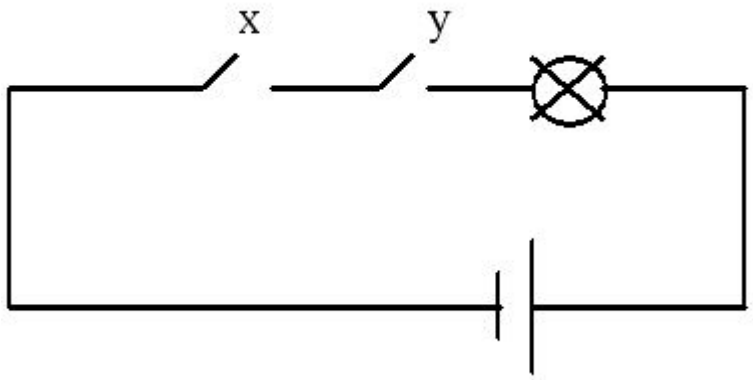


Рис. 16. Электрический аналог схемы конъюнктора

Схематически инвертор, дизъюнктор и конъюнктор на логических схемах различных устройств можно изображать условно следующим образом (рис. 17 а, б, в). Есть и другие общепринятые формы условных обозначений.

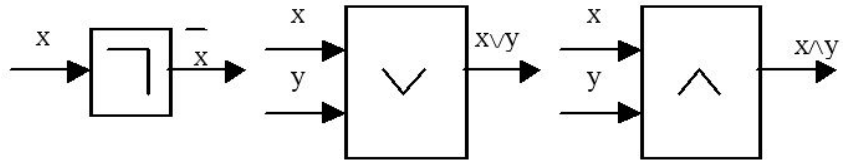


Рис. 17. а, б, в. Условные обозначения вентилей (вариант)

Пример. Транзисторные схемы, соответствующие логическим схемам $\neg$ (инвертор), $\vee$ (дизъюнктор), $\wedge$ (конъюнктор) имеют, например, следующий вид (рис. 18 а, б, в):

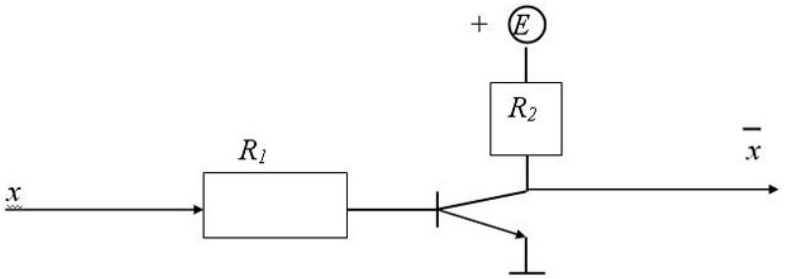


Рис. 18a. Инвертор

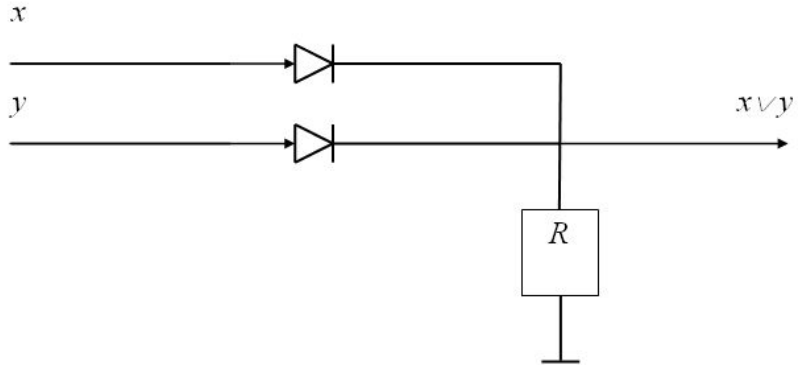


Рис. 18b. Дизъюнктор

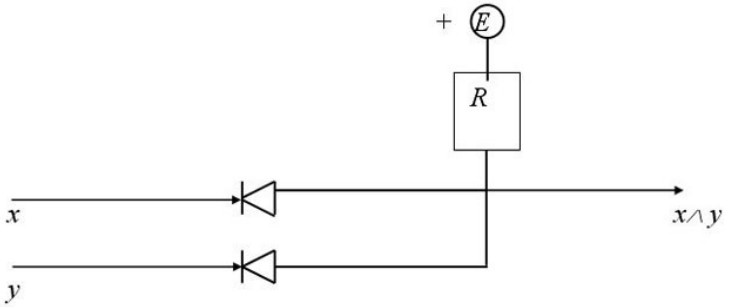


Рис. 18c. Конъюнктор

Из указанных простейших базовых логических элементов собирают, конструируют сложные логические схемы ЭВМ, например, сумматоры, шифраторы, дешифраторы и др. Большие (БИС) и сверхбольшие

(СБИС) интегральные схемы содержат в своем составе (на кристалле кремния площадью в несколько квадратных сантиметров) десятки тысяч вентилей. Это возможно еще и потому, что базовый набор логических схем (инвертор, конъюнктор, дизъюнктор) является функционально полным (любую логическую функцию можно представить через эти базовые вентили), представление логических констант в них одинаково (одинаковы электрические сигналы, представляющие 1 и 0) и различные схемы можно "соединять" и "вкладывать" друг в друга (осуществлять композицию и суперпозицию схем).

Таким способом конструируются более сложные узлы ЭВМ – ячейки памяти, регистры, шифраторы, дешифраторы, а также сложнейшие интегральные схемы.

Пример. В двоичной системе таблицу суммирования цифры x и цифры y и получения цифры z с учетом переноса p в некотором разряде чисел x и y можно изобразить таблицей вида

x	y	z	p
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Эту таблицу можно интерпретировать как совместно изображаемую таблицу логических функций (предикатов) вида

$$z = \bar{x} \wedge y \vee x \wedge \bar{y}$$
$$p = x \wedge y$$

Логический элемент, соответствующий этим функциям, называется одноразрядным сумматором и имеет следующую схему (обозначим ее как $\sum$ или $\sum_i$ – если мы хотим акцентировать именно выбранный, текущий i-й разряд) (рис. 19):

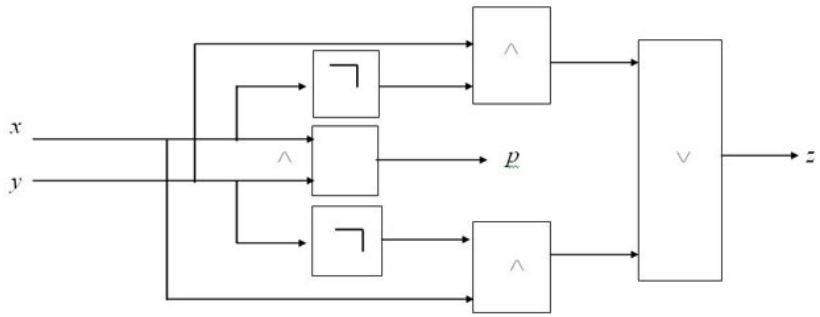


Рис. 19. Схема одноразрядного сумматора

Пример. "Черным ящиком" называется некоторое закрытое устройство (логическая, электрическая или иная схема), содержимое которого неизвестно и может быть определено (идентифицировано) только по отдельным проявлениям входа/выхода ящика (значениям входных и выходных сигналов). В "черном ящике" находится некоторая логическая схема, которая в ответ на некоторую последовательность входных (для ящика) логических констант выдает последовательность логических констант, получаемых после выполнения логической схемы внутри "черного ящика". Определим логическую функцию внутри "черного ящика" (рис. 20), если операции выполняются с логическими константами для входных последовательностей (поразрядно). Например, x = 00011101 соответствует последовательности поступающих значений: "ложь", "ложь", "ложь", "истина", "истина", "истина", "ложь", "истина".

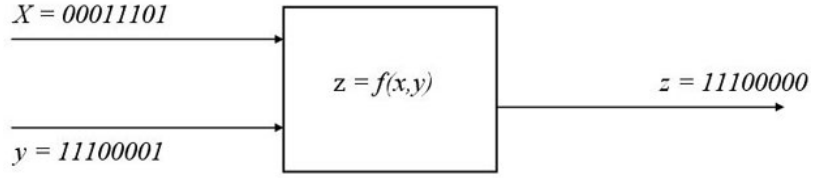


Рис.20. Схема "черного ящика 1"

Из анализа входных значений (входных сигналов) *x, y* и поразрядного сравнения логических констант в этих сообщениях с константами в значении z – результате выполнения функции в "черном ящике", видно, что подходит, например, функция вида

$$z = (x \vee y) \wedge \overline{x}$$

Действительно, в результате "поразрядного" сравнения сигналов (последовательностей значений "истина", "ложь") получаем следующие выражения (последовательности логических констант):

$$x \vee y = 00011101 \vee 11100001 = 11111101,$$
$$z = (x \vee y) \wedge \overline{x} = 11111101 \wedge 11100010 = 11100000.$$

Пример. Попробуйте самостоятельно выписать функцию для "черного ящика"? указанного на рис. 21:

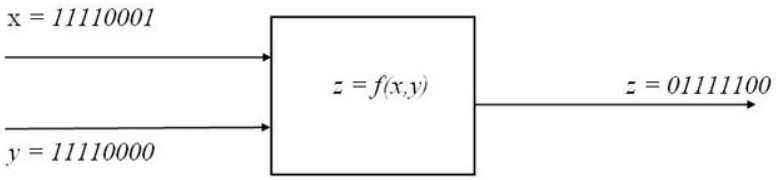


Рис. 21. Схема "черного ящика 2"

Важной задачей (технической информатики) является минимизация числа вентилей для реализации той или иной схемы (устройства), что необходимо для более рационального, эффективного воплощения этих схем, для большей производительности и меньшей стоимости ЭВМ.

Эту задачу решают с помощью методов теоретической информатики (методов булевой алгебры).

Пример. Построим схему для логической функции

$$u = \overline{x \wedge y \wedge z \vee t}$$

Схема, построенная для этой логической функции, приведена на рис. 22.

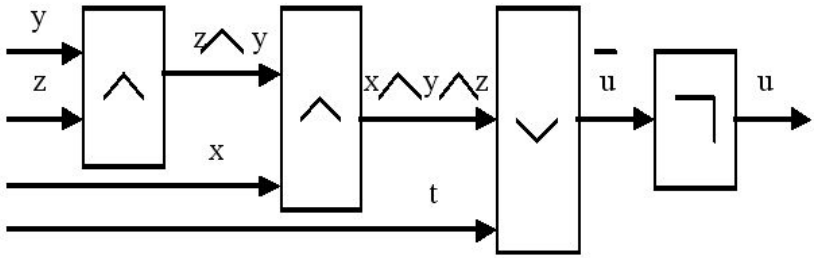


Рис. 22. Схема для функции 1

Пример. Определим логическую функцию $u = f(x, y, z, t)$, реализуемую логической схемой вида (рис. 23)

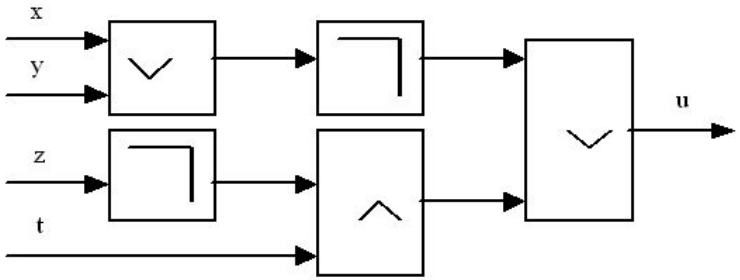


Рис. 23. Схема для функции 2

Искомая логическая функция, если выписать ее последовательно, заполняя "верх" каждой стрелки, будет иметь следующий вид:

$$u = \overline{x \vee y} \vee (z \wedge t)$$

6.7. Криотронные схемы.

Перспективным ключевым элементом является пленочный криотрон, действие которого основано на явлении сверхпроводимости при низких температурах. Условное изображение криотрона показано на рис. 24.

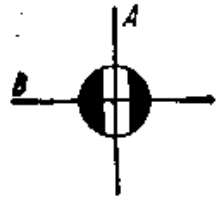


Рис. 24. Условное изображение криотрона

При отсутствии тока в управляющей шине материал (например, олово) обладает сверхпроводимостью, а при прохождении по шине *A* тока достаточной величины этот материал имеет конечное сопротивление. В результате цепь *B* действует как двусторонний управляемый контакт, причем для управления используются сигналы, соответствующие переменным x_i, и их отрицаниям $\bar{x}_i$.

Для анализа и синтеза криотроннмх схем применяют все рассмотренные методы с учетом специфических особенностей криотронов. Например, на рис. 25, *а* показана *криотронная схема с инверсными выходами,* реализующая функции $y = x_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2$ и $\bar{y} = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_2x_3$, а на рис. 25, *б* — соответствующая ей последовательно-параллельная контактная схема.

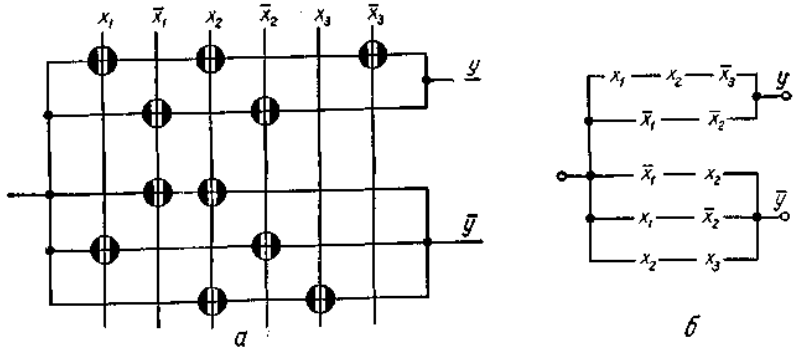


Рис. 25. Криотронная схема с инверсными выходами (а) и соответствующая ей контактная схема (б).

Аналогично используются полное криотронное дерево, булевы матрицы и т, п.

6.7.1. Справка из общего курса физики

Напомним, что при очень низких температурах, ниже т.н. "критической температуры" T_K (для каждого сверхпроводящего материала она своя), ряд металлов и сплавов (в частности ниобий и его аналоги, используемые чаще всего), а также и ряд керамических материалов становятся сверхпроводящими, т.е. не оказывают совсем никакого сопротивления прохождению электрического тока. Кроме того, из толщи сверхпроводника магнитное поле вытесняется: индукция магнитного поля внутри сверхпроводника равна нулю. Однако, когда внешнее магнитное поле становится достаточно сильным, выше "критического" значения B_K, то состояние сверхпроводимости "разрушается". Разрушается оно и в случае, если плотность электрического тока в сверхпроводнике превышает некоторое "критическое" значение j_K.

Поскольку электрическое сопротивление сверхпроводника равно нулю, то в замкнутом сверхпроводящем контуре электрический ток может бесконечно долго циркулировать без затухания при отсутствии любой посторонней ЭДС.

Сверхпроводимость объясняется тем, что при определенных условиях двум электронам с противоположно направленными спинами становится энергетически выгоднее объединиться в неразрывную пару. Такие связанные между собой электроны называют "куперовскими парами" (в честь ученого Л. Купера, который впервые показал, что электроны в сверхпроводниках объединяются в пары). Каждая пара связанных электронов ведет себя как квазичастица с нулевым спином и с электрическим зарядом, равным двум зарядам электрона. Куперовские пары, как и всякая квантовая система, могут находиться лишь в разрешенных энергетических состояниях. Для перехода в другое разрешенное энергетическое состояние нужна значительная энергия. Поэтому куперовские пары не могут рассеиваться на фононах, примесных атомах, ионах, дефектах кристаллической решетки, из-за чего и исчезает электрическое сопротивление.

На квазичастицы с нулевым спином не распространяется квантово-механический принцип Паули, и все они находятся в одном и том же квантовом состоянии, описываются общей волновой функцией, другими словами, являются когерентными. Именно благодаря этому в сверхпроводниках для куперовских пар электронов имеет место т.н. "макроскопическая квантовая интерференция". Она приводит к тому, что магнитный поток, пронизывающий отверстие контура, "квантуется", т.е. должен быть кратным характерной величине

$$\Phi_0 = \frac{h}{2e} = 2.07 * 10^{-15} [Вб], \qquad (1.1)$$

где h и e – известные физические константы (постоянная Планка и заряд электрона). Эту величину называют "квантом магнитного потока" или "флюксоном", а замкнутые контуры из сверхпроводников с включенными в них т.н. переходами Джозефсона – сверхпроводящими квантовыми интерферометрами или сокращенно "сквидами" (от англ. "SQUID" – "Superconducting Quantum Interference Device").

6.7.2. Сверхпроводящая элементная база на криотронах

Существование "критического" значения напряженности магнитного поля B_K, при превышении которого состояние сверхпроводимости исчезает, позволило создать довольно эффективные криоэлектронные вентили, которые были названы криотронами. Структура простейшего криотрона показана на рис. 1.1.а.

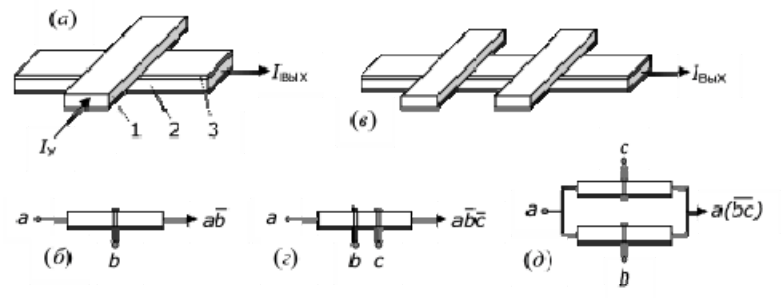


Рис. 1.1. (а) Структура простейшего криотрона; (б) его логическая схема; (в) криотрон с двумя управляющими шинами; (г) его логическая схема; (д) пример логической схемы на криотронах; 1 – сверхпроводник с большим значением B_K; 2 – сверхпроводник с меньшим значением B_K; 3 – изолирующий слой

Он состоит из сверхпроводящей шины 1 из материала с большим значением критического магнитного поля B_K (например, из ниобия) и ортогональной к ней сверхпроводящей шины 2 из материала со значительно меньшим значением B_K (например, из тантала), которые гальванически разделены тонким слоем изолятора 3. Первая сверхпроводящая шина (1) считается управляющей, вторая (2) – управляемой. Через управляющую шину пропускают электрический ток I_y такой величины, чтобы создаваемое им магнитное поле превышало критическое значение B_K для шины 2. В этом случае состояние сверхпроводимости в ней разрушается, и электрический ток сквозь нее резко уменьшается. Геометрические размеры и материал сверхпроводящих шин можно подобрать так, чтобы выходной ток был значительно больше управляющего тока и был достаточным для управления одновременно несколькими криотронами последующей логической цепи.

Если считать наличие сверхпроводящего состояния и протекание сквозь управляемую шину значительного тока логической "1", а разрушенное состояние сверхпроводимости и протекание незначительного тока –логическим "0", то соответствующая логическая схема простейшего криотрона будет выглядеть так, как показано на рис. 1.1.б. Управляемая шина изображена здесь прямоугольником, а управляющая – двойной чертой. Значительный ток сквозь криотрон течет лишь в том случае, когда на вход a криотрона

подается значительный ток ($a = 1$) и одновременно в управляющей шине ток незначителен ($b = 0$). Реализуется логическая операция $a \wedge \bar{b}$. На рис. 1.1.в показан криотрон с двумя управляющими шинами, а на рис. 1.1.г – соответствующая ему логическая схема. Еще один вариант логической схемы на криотронах показан на рис. 1.1.д.

Другой, более удобный для практики вариант логики на криотронах показан на рис. 1.2.

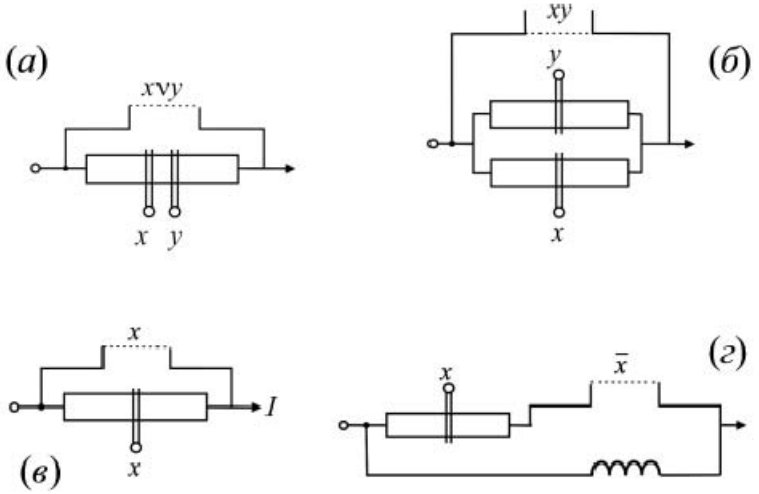


Рис. 1.2. Логические схемы на криотронах с альтернативной выходной ветвью: (а) элемент, реализующий дизъюнкцию; (б) элемент, реализующий конъюнкцию; (в) элемент, реализующий тождество (повторитель); (г) элемент, реализующий отрицание

Здесь параллельно каждой управляемой шине включена альтернативная ветвь. Когда управляемая шина находится в сверхпроводящем состоянии, через альтернативную ветвь ток не течет (или течет очень незначительный ток). Это состояние считают логическим "0". Когда в управляемой шине состояние сверхпроводимости разрушается, то электрический ток течет через альтернативную ветвь. Это состояние считают логической "1". В схеме

на рис. 1.2.а ток в выходной ветви ($x \vee y$) течет в тех случаях, когда хотя бы через одну из управляющих шин (x, y) течет сверхпроводящий ток, разрушающий сверхпроводимость в управляемой ветви. На рис. 1.2.б показана схема, реализующая конъюнкцию. Ток в выходной ветви ($x \wedge y$) течет лишь в том случае, когда через каждую из управляющих шин (x, y) течет ток, разрушающий сверхпроводимость в соответствующей управляемой шине. На рис. 1.2.в показана схема, реализующая логическую операцию тождества, а на рис. 1.2.г – схема, реализующая логическую операцию отрицания. Электрический ток, который течет через выходную ветвь, используется для управления другими криотронами или для считывания результата логических преобразований.

В целом перечисленные элементы образуют технически полную систему логических элементов, т.е. из них можно построить любую логическую схему, а также элементы памяти, триггеры, процессоры.

Принцип действия криотронов допускает их масштабирование до нанометровых размеров. Чем меньше размеры криотронов, тем меньшие токи они потребляют и, соответственно, тем меньше рассеивают тепла. Собственное время их переключения можно оценить по формуле

$$\tau \approx \frac{\mu_0}{\rho} d_y d_u \qquad (1.2)$$

где $\mu_0 = 1{,}257 \cdot 10^{-6}$ Гн/м – магнитная постоянная; ρ – удельное сопротивление материала управляемой шины в нормальном (не сверхпроводящем) состоянии; d_y – толщина управляющей шины; d_u – толщина слоя изолятора. При d_y и $d_u < 100$ нм собственное время переключения может быть меньше 10 пс. Реальное быстродействие криотронных логических схем зависит от электрической емкости межсоединений и даже при нанометровых размерах криотронов не превышает нескольких гигагерц. Для современных требований это маловато.

При всех своих положительных характеристиках – малая потребляемая мощность, высокие потенциально возможные плотность компоновки и уровень интеграции, – наноразмерные пленочные криотроны, кроме далеко не рекордного быстродействия, имеют такой недостаток, как работоспособность лишь при низких температурах.

6.7.3. Переходы и эффекты Джозефсона.

В 1962 г. Б. Джозефсон, позднее ставший лауреатом Нобелевской премии, показал, что определенный сверхпроводящий ток может протекать, не встречая сопротивления, также и через тонкий туннельный барьер между двумя сверхпроводниками (рис. 1.3.а). Такие барьеры стали называть "переходами Джозефсона" (далее мы употребляем сокращение "ПД").

На практике применяют обычно один из нескольких типов ПД. Чаще всего это очень тонкий (1-2 нм) слой окисла между двумя металлическими сверхпроводниками, – структура SIS (сверхпроводник – изолятор – сверхпроводник). Другой тип имеет структуру SNS (сверхпроводник – нормальный металл – сверхпроводник). Слой нормального металла может иметь толщину уже порядка 10 нм. Третий тип это SFS (сверхпроводник – ферромагнетик – сверхпроводник). Имеются и другие.

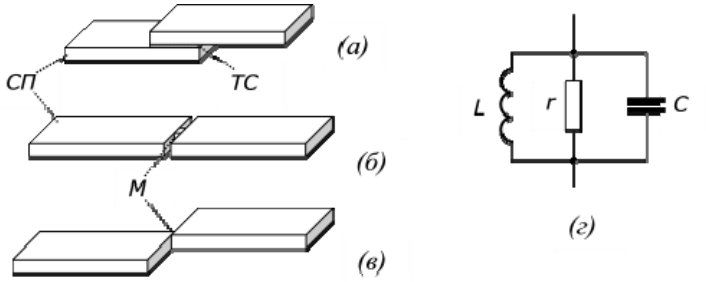


Рис. 1.3. Возможные варианты структуры перехода Джозефсона (а, б, в) и его эквивалентная электрическая схема (г): СП – сверхпроводник; ТС – туннельный слой; М – "мостик" между сверхпроводниками; С – электроемкость; L – нелинейная индуктивность; r – активное сопротивление (становится существенным, когда ток сквозь переход превышает I_k)

Позднее выяснилось, что ПД может возникать при наличии любого "слабого" контакта между сверхпроводниками. Из них чаще всего используют т.н. "мостик" – тонкую перемычку между двумя планарными сверхпроводниками, размер которой меньше "длины когерентности" (среднее расстояние между связанными электронами куперовской пары, рис. 1.2.б,в).

Энергетическая диаграмма ПД для куперовских пар электронов показана на рис. 1.4: слева – для случая, когда напряжение на переходе равно нулю, справа – для случая, когда оно отличается от нуля.

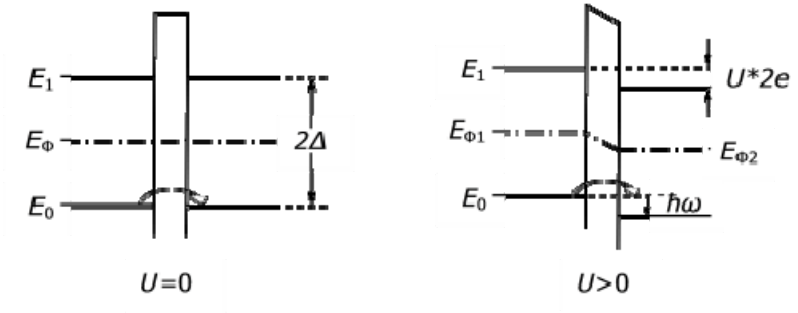


Рис. 1.4. Энергетические диаграммы перехода Джозефсона для куперовских пар: слева – когда напряжение на ПД U = 0, справа – когда U>0

Здесь вдоль вертикали отложена энергия, вдоль горизонтали – координата. E_0 – это наиболее низкий разрешенный уровень энергии для куперовской пары, E_1 – следующий разрешенный уровень, 2Δ – это ширина "запрещенной зоны" ("энергетической щели") между этими уровнями. Через $E_Ф$ обозначен уровень Ферми. При отсутствии напряжения на ПД уровни Ферми слева и справа от ПД совпадают, и, если переход достаточно тонок, то куперовские пары благодаря туннельному эффекту могут проникать сквозь него.

Критический ток I_K сквозь ПД как правило значительно меньше, чем критический ток в соответствующем сверхпроводнике, и равен

$$I_K = \frac{\Delta}{2eR} \qquad (1.3)$$

где R – электрическое сопротивление ПД в нормальном (не сверхпроводящем) состоянии.

Между волновыми функциями куперовских пар с обеих сторон барьера в результате туннельного обмена устанавливается некоторая разность фаз φ, которая и определяет величину электрического сверхпроводящего тока сквозь ПД:

$$I = I_K \sin\varphi \qquad (1.4)$$

Когда сквозь ПД электрический ток не течет, тогда разность фаз $\varphi = 0$, а когда течет максимально допустимый сверхпроводящий ток $I = I_K$, то $\varphi = \pi/2$. Пока $I \le I_K$, напряжение на ПД равно нулю. Все это вместе называют стационарным эффектом Джозефсона.

Если сквозь ПД пропускать электрический ток $I > I_K$, то часть его $I - I_K$ будет переноситься не куперовскими парами, а обычными (попарно не связанными) электронами проводимости. Поскольку для этой части тока (не сверхпроводящей) действует закон Ома, то на ПД появляется падение напряжения

$$U = R(I - I_K) \qquad (1.5)$$

Б. Джозефсон показал, что, когда на переходе имеется электрическое напряжение U, то разность фаз φ по законам квантовой механики начинает изменяться. Скорость изменения описывается формулой Джозефсона

$$\frac{d\varphi}{dt} = \frac{2eU}{\hbar} \qquad (1.6)$$

Интегрируя выражение (1.6) по времени, получаем

$$\varphi(t) = \varphi(0) + \frac{2eU}{\hbar}t = \varphi_0 + \omega t \qquad (1.7)$$

где

$$\omega = \frac{2eU}{\hbar} \approx 483.6(\text{МГц/мкВ}) * U(\text{мкВ}) \qquad (1.8)$$

- т.н. "джозефсоновская частота". Подставляя (1.7) в формулу (1.4), находим, что при наличии напряжения на ПД сверхпроводящая составляющая тока через переход

$$I_{СП} = I_K \sin(\varphi_0 + \omega T) \qquad (1.9)$$

т.е. гармонически изменяется со временем. Это означает, что через ПД, кроме постоянного не сверхпроводящего тока $I = U/R$, течет сверхпроводящий переменный электрический ток с частотой (1.8), пропорциональной приложенному напряжению. Этот эффект называют нестационарным эффектом Джозефсона.

На энергетической диаграмме, изображенной на рис. 1.4 справа, этому эффекту можно поставить в соответствие процесс, в котором куперовская пара прошла сквозь ПД. На другой стороне перехода оказывается, что она имеет "избыточную" (относительно разрешенного здесь уровня) энергию $2eR$, которую она сразу же излучает в виде кванта электромагнитной волны с такой же энергией. Суммарное когерентное действие многих куперовских пар, которые проходят сквозь ПД, и предопределяет "джозефсоновскую генерацию".

Переход Джозефсона лучше характеризовать не вольтамперной характеристикой, как это обычно делают, а ампер-вольтной (рис. 1.5), где вдоль оси абсцисс отложен электрический ток сквозь переход, а вдоль оси ординат – падение напряжения на переходе.

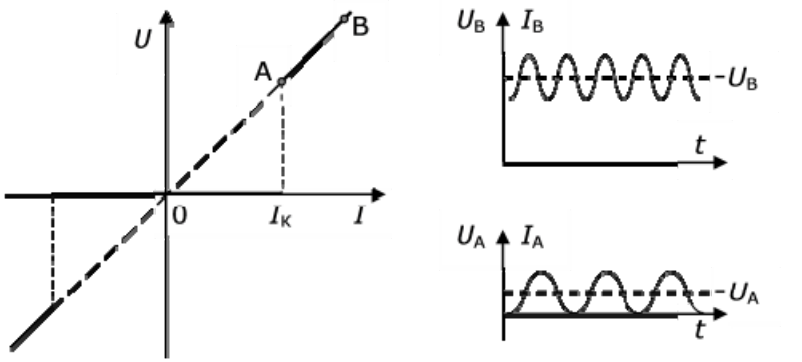


Рис. 1.5. Ампер-вольтная характеристика перехода Джозефсона (слева) и генерация переменного тока при $I>I_k$ (справа)

Пока ток сквозь ПД не превосходит критическое значение, падения напряжения на переходе нет. Но если ток превосходит критический, на переходе падает напряжение, пропорциональное величине тока. При этом возникает джозефсоновская генерация. Типичные зависимости напряжения на переходе и тока сквозь переход показаны на рис. 1.5 справа для двух "рабочих точек" А и В, помеченных на статической характеристике слева. В рабочей точке А средний (по времени) ток равняется I_K, среднее напряжение на ПД $U_A = 2\Delta/e$, круговая частота колебаний тока $\omega_A = 4\Delta/\hbar$. В рабочей точке В напряжение на ПД $U_B = I_B R$, а круговая частота колебаний тока $\omega_B = 2eU_B/\hbar$.

Приведенные характеристики, строго говоря, касаются идеального ПД. Реальные ПД имеют дополнительно также "геометрические" емкость и индуктивность. Электрическая емкость определяется геометрическими размерами, прежде всего площадью, реального перехода Джозефсона. В ПД туннельного типа (рис. 1.3.а) она значительно больше, чем в ПД

типа "мостик" (рис. 1.3.б,в). А суммарная индуктивность ПД, как коэффициент пропорциональности между ЭДС самоиндукции и скоростью изменения тока, не является постоянной, а нелинейно зависит от разности фаз φ:

$$L = h/(2e \cdot I_K \cdot \cos\varphi) \qquad (1.10)$$

Собственные электрическая емкость и индуктивность ПД могут существенно влиять на его характеристики в переходных режимах и на переменном токе. Выполнение ПД нанометровых размеров практически снимает эти проблемы.

6.7.4. Сквиды с переходами Джозефсона и их применение

6.7.4.1. Сквид с одним переходом Джозефсона

Если ПД входит в состав замкнутого сверхпроводящего контура (рис. 1.6.а), то наблюдаются дополнительные эффекты. Такой контур называют сквидом с одним переходом Джозефсона.

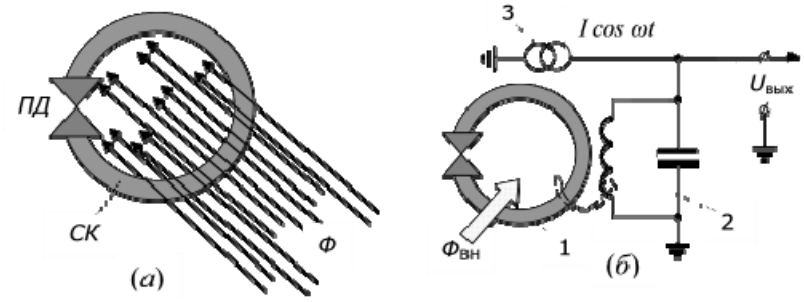


Рис. 1.6. (а) Сквид з одним переходом Джозефсона (ПД), СК – сверхпроводящий контур; Ф – магнитный поток сквозь контур. (б) схема магнитометра переменного тока, 1 – сквид с одним ПД, 2 – индуктивно связанный с ним колебательный контур, $\Phi_{ВН}$ – внешний магнитный поток

Состояние такого сквида однозначно связано с величиной Φ суммарного магнитного потока, пронизывающего его контур. В самом деле, квантовая интерференция куперовских пар приводит к тому, что в стационарном состоянии замкнутый сверхпроводящий контур может пронизывать лишь магнитный поток, равный целому числу квантов магнитного потока Φ_0. Поэтому сверхпроводящий ток в контуре всегда имеет такую величину, чтобы создаваемый им магнитный поток через контур дополнял (или уменьшал) магнитный поток $\Phi_{ВН}$, созданный внешними источниками, до величины, кратной Φ_0. Если внешний магнитный поток $\Phi_{ВН}$ начинает изменяться, то по закону магнитной индукции Фарадея в контуре возникает ЭДС, под действием которой по закону Джозефсона (1.6) начинает изменяться и разность фаз φ на ПД. А это по закону (1.4) вызывает изменение электрического тока через ПД такое, чтобы суммарный магнитный поток сквозь контур стал кратным Φ_0.

Разность фаз φ однозначно "отслеживает" изменения потока $\Phi_{ВН}$. Если это изменение очень быстрое, так что благодаря значительной ЭДС индукции ток через ПД превышает критическое значение, то излишек тока переносится обычными электронами проводимости, которые не являются когерентными к куперовских парам. При наличии не сверхпроводящего тока в контур "прорываются" дополнительные кванты магнитного потока до тех пор, пока разность $|\Phi - \Phi_{ВН}|$ не станет меньше Φ_0. Лишь тогда ток через ПД снова падает ниже критического значения, и сверхпроводящее состояние ПД восстанавливается.

Благодаря тому, что сквид с одним ПД "реагирует" на наименьшие изменения магнитного потока, его часто используют для создания очень чувствительных сенсоров магнитного поля. Одна из наиболее распространенных схем такого сенсора, которую называют магнитометром переменного тока, показана на рис. 1.6.б. Кроме сквида 1 с одним ПД, в нем используют миниатюрный высокодобротный колебательный контур 2, индуктивно связанный со сквидом. Индуктивная связь условно изображена штриховой стрелкой. На

колебательный контур от генератора 3 подают переменный ток с частотой, близкой к резонансной частоте колебательного контура 2. В этом случае импеданс контура становится очень чувствительным к магнитному потоку $\Phi_{ВН}$. Выходным сигналом является падение переменного напряжения $U_{ВЫХ}$ на колебательном контуре.

6.7.4.2. Сквид с двумя переходами Джозефсона

Сквид с двумя ПД показан на рис. 1.7. Здесь СП – сверхпроводящие шины, ПД1 и ПД2 – переходы Джозефсона. Слева показана типичная схема применения такого сквида. Переходы Джозефсона ПД1 и ПД2 включены здесь в сверхпроводящий контур как два симметричных параллельных "плеча". В сверхпроводящем контуре циркулирует не затухающий сверхпроводящий ток $I_Ц$. Его направление и величина зависят от величины внешнего магнитного потока, который пронизывает контур. Ведь благодаря квантовой интерференции ток автоматически поддерживается таким, чтобы суммарный магнитный поток сквозь контур оставался кратным к Φ_0.

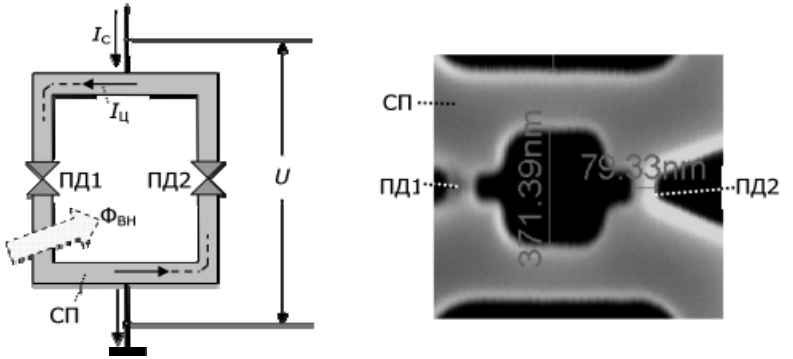


Рис. 1.7. Сквид с двумя переходами Джозефсона (ПД1 и ПД2), СП – сверхпроводящие шины. Слева – схема применения, справа – микрофотография одного из вариантов наноразмерного сквида, полученная с помощью растрового электронного микроскопа

Кроме того, извне в контур подается постоянный "ток смещения" I_C, который иногда называют еще "транспортным током". В результате через ПД1 течет электрический ток $(0.5I_C + I_Ц)$, а через ПД2 – электрический ток $(0.5I_C - I_Ц)$. Если в одном из плеч величина тока превышает критический ток ПД, то сверхпроводимость разрушается, ток смещения переключается в другое плечо, из-за чего и здесь сверхпроводимость разрушается. И на сквиде наблюдается падение напряжения U.

Благодаря квантовой интерференции падение напряжения зависит от величины внешнего магнитного потока $\Phi_{ВН}$, который пронизывает контур. И зависимость эта является периодической с периодом, равным кванту магнитного потока Φ_0 (рис. 1.8 слева). Эту зависимость называют сигнальной кривой. Амплитуда и положение сигнальной кривой зависят от тока смещения и от дополнительных магнитных потоков, пронизывающих сверхпроводящий контур. Справа показана функциональная схема магнитометра постоянного тока на сквиде С с двумя ПД. Величину постоянного электрического тока смещения задает генератор ГТС. Падение напряжения на сквиде воспринимается, усиливается и обрабатывается в электронном узле обработки сигналов УОС. Через катушку обратной связи КОС, магнитно-связанную со сквидом, в него подается дополнительный магнитный поток, который выбирают так, чтобы "рабочая точка" сквида всегда находилась на самом крутом участке сигнальной кривой. Благодаря этому прирост $\Delta\Phi_C$ измеряемого магнитного потока дает максимальный прирост ΔU выходного напряжения. Это позволяет автоматически поддерживать максимальную чувствительность магнитометра.

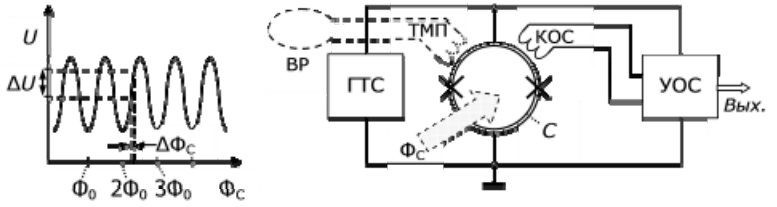


Рис. 1.8. Слева – зависимость напряжения на сквиде с двумя ПД от внешнего магнитного потока, пронизывающего сверхпроводящий контур. Справа – схема магнитометра постоянного тока: С – сквид, ГТС – генератор тока смещения, УОС –узел обработки сигналов, КОС – катушка обратной связи, ВР – выносная рамка

6.7.4.3. Градиометры магнитного поля

С помощью выносной рамки ВР и трансформатора магнитного потока ТМП, изображенных штриховой линией на рис. 1.8 справа, можно измерять изменения магнитного потока не только в месте непосредственного расположения сквида, но и на некотором расстоянии от него. Ведь изменения магнитного потока, пронизывающего выносную рамку, приводят к соответствующему изменению тока через катушку ТМП и связанного с ней магнитного потока сквозь контур сквида. Ориентируя выносную рамку перпендикулярно к осям OX, OY, OZ (рис. 1.9 слева), можно измерять изменения компонент $\Delta\Phi_x$, $\Delta\Phi_y$ и $\Delta\Phi_z$ магнитного потока.

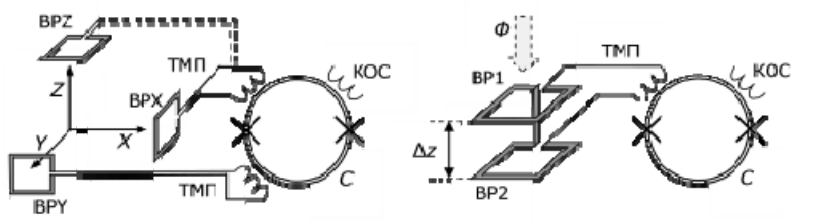


Рис. 1.9. Слева – варианты ориентации выносной рамки для измерения компонент магнитного потока, ориентированных вдоль координатных осей. Справа – схема расположения пары выносных рамок для градиометра магнитного поля 1-го порядка. Обозначения как на рис. 1.8

Справа на рис. 1.9 показана схема расположения двух одинаковых выносных рамок (ВР1 и ВР2) с целью использования магнитометра постоянного тока на сквиде С с двумя ПД для измерения градиента магнитного поля. Поскольку выносные рамки ВР1 и ВР2 включены здесь навстречу одна другой, то в контур сквида трансформируется

лишь разность $(\Delta\Phi = \Phi_2 - \Phi_1)$ магнитных потоков, пронизывающих эти выносные рамки.

Зная расстояние Δz между ними, легко вычислить градиент магнитного потока

$$\frac{\partial\Phi}{\partial z} \approx \frac{\Delta\Phi}{\Delta z} \qquad (1.11)$$

А если расположить рядом две такие пары рамок и включить их навстречу одна другой (рис. 1.10), то магнитный поток, который трансформируется в контур сквида, будет равняться разности $(\Delta(\Delta\Phi) = \Delta\Phi_2 - \Delta\Phi_1)$.

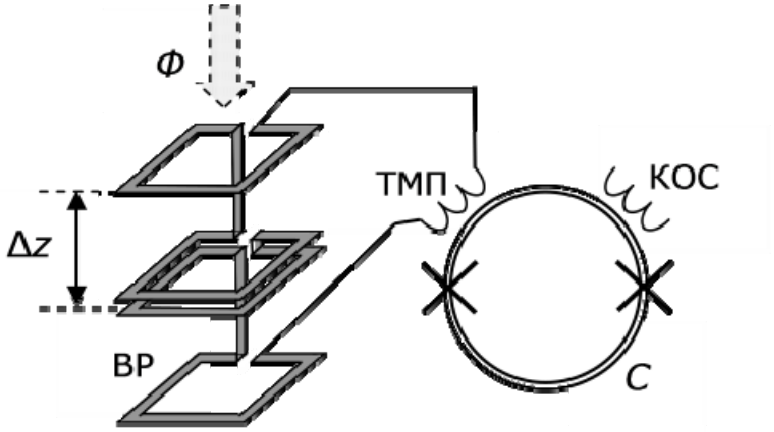


Рис. 1.10. Схема расположения двух пар выносных рамок для градиометра магнитного поля 2-го порядка. Обозначения как на рис. 1.8

Зная расстояние Δz между парами рамок, легко вычислить производную магнитного потока 2-го порядка

$$\frac{\partial^2\Phi}{\partial^2 z} = \frac{\Delta(\Delta\Phi)}{(\Delta z)^2} \quad (1.12)$$

Магнитометры постоянного тока на сквиде С с двумя ПД в таких случаях называют градиометрами 1-го и 2-го порядка.

6.7.4.4. Измерение слабых магнитных полей

Чувствительность магнитометров характеризуют минимальным изменением магнитного потока, которое можно зафиксировать, отнесенным к единичной частотной полосе. Сейчас чувствительность наилучших сверхпроводящих магнитометров достигает $10^{-6}\Phi_0/\sqrt{\Gamma ц} \approx 2 * 10^{-21}$Тл$/\sqrt{\Gamma ц}$.

Чувствительность относительно магнитной индукции достигает 10^{-14} Тл. И очень важно, что чувствительность эта не зависит от уровня постоянной составляющей магнитного поля, т.е. совсем небольшие изменения можно измерять на фоне относительно сильного постоянного магнитного поля, например, магнитного поля Земли.

Как мы уже отмечали, физические принципы работы сквидов допускают их масштабирование, т.е. уменьшение их размеров вплоть до значений порядка длины когерентности куперовских пар (порядка 10 нм). С помощью наноразмерных сквидов удается измерять предельно слабые магнитные поля, создаваемые даже отдельными наночастицами. Принцип измерения показан на рис. 1.11 слева, где изображены сверхпроводящий контур наноразмерного сквида с двумя ПД и ряд случайно расположенных магнитных наночастиц (МНЧ).

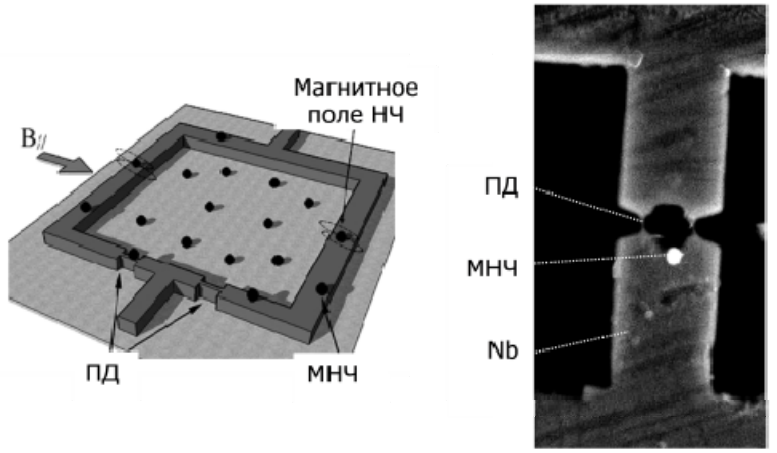


Рис. 1.11. Слева – схема измерения магнитного поля отдельных наночастиц (НЧ). Справа – микрофотография в сканирующем электронном микроскопе магнитной наночастицы (МНЧ) диаметром 165 нм на краю сквида. ПД – переходы Джозефсона (здесь "мостики")

Магнитное поле наночастиц, расположенных внутри контура, замыкается в нем и поэтому не создает дополнительный магнитный поток сквозь контур. Магнитное поле наночастиц, расположенных на краю сверхпроводящего контура (оно условно показано пунктирными линиями) с одной стороны пронизывает контур, а с другой проходит мимо него и потому создает дополнительный магнитный поток, который можно измерять. Пример реализации измерения показан на рис. 1.11 справа. Сверхпроводящий контур с внутренним отверстием 500×500нм сделан здесь из ниобия, переходы Джозефсона (ПД) выполнены в виде "мостиков". Магнитная наночастица (МНЧ) диаметром 165 нм из сплава железа с платиной размещена на краю сверхпроводящего контура с помощью прецизионного зондового манипулятора.

В 2009 г. можно было регистрировать магнитные поля от наночастиц с магнитным моментом порядка 100 магнетонов Бора. Сейчас считается возможной регистрация с помощью магнитометров на сквидах магнитных полей от отдельных молекул и даже атомов.

6.7.5. Многоканальные магнитометры на сквидах

6.7.5.1. Магнитокардиографы

На базе сверхчувствительных магнитометров на сквидах создан целый ряд многоканальных интеллектуальных сенсоров. Одним из показательных примеров являются магнитокардиографы – интеллектуальные сенсоры, позволяющие регистрировать и отслеживать изменения магнитного поля, связанные с функционированием сердца, и делать на основе этого важные для медицинской диагностики выводы. В качестве примера, опишем коротко магнитокардиографический комплекс "Кардиомагскан", функциональная схема которого показана на рис. 1.12 вверху.

Многоканальный магнитометр 1 на сквидах воспринимает в четырех точках (Измерительные сквиды) вертикальную компоненту ритмических изменений магнитного поля, обусловленных работой сердца пациента, а также все три пространственные компоненты (Референтные сквиды) фонового магнитного поля, усиливает их и передает в электронный блок 2. В состав этого блока входят микропроцессор и мультиплексор, управляющие порядком считывания сигналов от разных сквидов, обрабатывающие полученные сигналы и передающие данные в виде цифровых кодов персональному компьютеру (ПК). Все сквиды находятся при температуре, ниже критической. Необходимая температура поддерживается благодаря криостату 3 с жидким гелием. Одной заправки криостата (11 л жидкого гелия) хватает для непрерывной работы магнитометра на протяжении 5 суток, т.е. ее хватает на всю рабочую неделю.

Криостат 3 и многоканальный магнитометр 1 конструктивно объединены в криогенный модуль 4. Пациент спокойно лежит на кровати, которую можно передвигать так, чтобы точно установить сквиды в нужную позицию относительно сердца пациента (рис. 1.12, внизу). С целью получения опорных сигналов с помощью электрокардиографа 5 регистрируется также стандартная электрокардиограмма. Опорные сигналы ЭКГ передаются в электронный блок 2, где используются для определения моментов считывания магнитных сигналов, связанных с работой сердца, и на ПК. Электронный блок 2 автоматически компенсирует внешнее магнитное поле, контролирует уровень жидкого гелия в криостате, вырабатывает

стандартные сигналы, с помощью которых можно проверять работу и характеристики каждого из каналов, регулировать их.

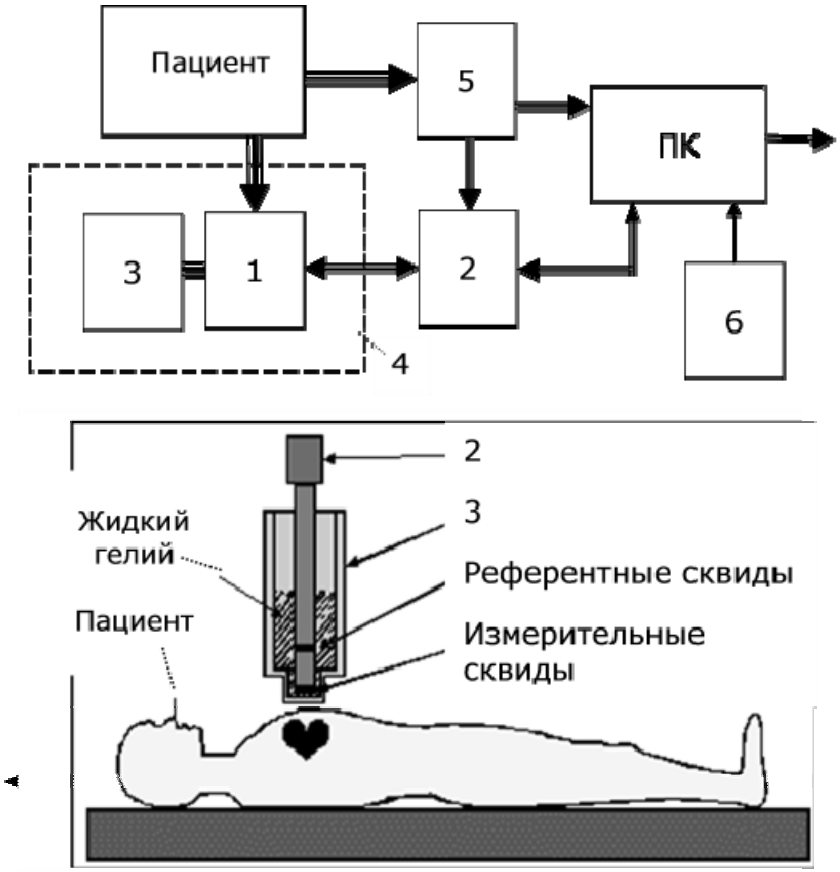


Рис. 1.12. Вверху – функциональная схема магнитнокардиографа "Кардиомагскан": 1 – многоканальный магнитометр на сквидах; 2 – электронный блок; 3 – криостат с жидким гелием; 4 – криогенный модуль; ПК – персональный компьютер; 5 – электрокардиограф; 6 – программное обеспечение. Внизу – схема расположения пациента и криогенного модуля

Программное обеспечение 6 системы состоит из двух автономных пакетов программ, работа которых распределена во времени.

Первый пакет используется во время проведения измерений, а второй пакет – после проведения обследований всей группы пациентов – с целью дальнейшего детального анализа полученных результатов. Программные модули первого пакета обеспечивают прием информации от электронного блока 2, контроль ее качества и размещение в базе данных, цифровую фильтрацию полученных сигналов, прием и анализ ЭКГ сигналов, формирование и выдачу на экран ПК магнитокардиограмм, синхронизированных с электрокардиограммой. Они поддерживают также обратную связь с блоком 2 с целью автоматической оптимизации условий измерения в зависимости от конкретных обстоятельств и особенностей организма пациента, выполняют усреднение полученных данных по нескольким сердечным циклам, предоставляют возможность редактирования полученных данных квалифицированным специалистом, организуют работу с базой данных и т.д.

Второй программный пакет содержит модули анализа пространственно-временных изменений магнитного поля сердца, построение их изображения на экране монитора, вычисление ряда медико-диагностических параметров, характеризующих динамику работы участков сердца, нарушение координации сердечных ритмов и т.п. В этот пакет программ входят также модули, которые решают непростую т.н. "обратную задачу" теории поля: по пространственному распределению магнитного поля найти распределение его источников. По этим данным строятся и выводятся на экран монитора изображения источников магнитного поля (электрических токов) в соответствующем сечении сердечной мышцы в разных фазах сердечного цикла, определяется локализация в сердце аритмогенных зон и т.д. Достигается пространственная разрешающая способность (точность локализации источников сигналов) порядка 1 мм.

Магнитокардиография на основе сквидов позволяет, например, бесконтактным способом зарегистрировать работу сердца еще не родившегося ребенка, своевременно уловить угрожающие сбои в его работе.

6.7.5.2. Магнитоэнцефалографы и томографы

По аналогичному принципу построены также магнитоэнцефалографы, позволяющие регистрировать слабые переменные магнитные поля, связанные с работой мозга человека, обнаруживать имеющиеся там

нарушения активности, локализовать места нарушений. На рис. 1.13 показаны фотографии разных вариантов конструктивной реализации магнитноэнцефалографа на сквидах. Вверху в центре на вставке показана схема расположения измерительных сквидов относительно мозга пациента. Анализ данных от разных сквидов позволяет точно локализовать участки мозговой активности.

Для исследования и диагностики мозга требуется значительно больше каналов измерения – десятки или даже сотни. Наноразмеры сквидов позволяют реализовать даже десятки тысяч каналов. Все дело в стоимости электроники и в скорости обработки сигналов. В то время, как для исследования и диагностирования работы сердца достаточным является частотный диапазон сигналов до 100 Гц, для изучения и диагностирования работы мозга человека требуется уже частотный диапазон до 1000 Гц. Поскольку магнитные сигналы от мозга значительно слабее, чем магнитные сигналы от сердца, то помещение, где расположены магнитоэнцефалографы, приходится и значительно лучше экранировать от внешних магнитных полей. В лучших современных образцах магнитоэнцефалографов положение участков мозговых нарушений тоже можно локализовать с точностью порядка 1 мм.

Уже созданы даже 200-канальные томографы на сквидах, которые позволяют снимать и исследовать условные "срезы" мозга человека.

Рис. 1.13. Различные варианты реализации магнитноэнцефалографа на сквидах и схема фиксации активности отдельных участков мозга пациента (на вставке вверху в центре)

6.7.6. Растровые микроскопы на сквидах

Еще одним примером интеллектуальных сенсоров на сквидах является растровый сквид-микроскоп. Его общая функциональная схема показана на рис. 1.14 слева.

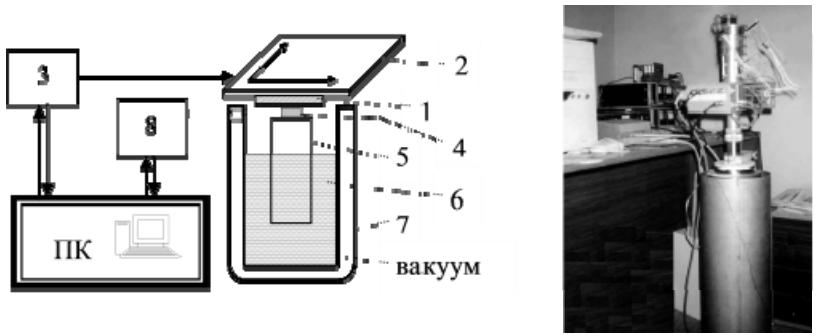


Рис. 1.14. Слева – функциональная схема растрового сквид-микроскопа: 1 – исследуемый образец, 2 – координатный стол, 3 –узел управления столом, 4 – чувствительные элементы на сквидах, 5 – хладопровод, 6 – жидкий азот или гелий, 7 – криостат, 8 – электронный блок; справа – растровый микроскоп ССМ-77

Исследуемый образец 1 устанавливают на прецизионном координатном столе 2, работой которого управляет электронный узел 3. Рядом с образцом, вплотную к нему находится микроминиатюрный магниточувствительный элемент 4 на сквидах. Чтобы поддерживать температуру ниже критической, он установлен на хладопроводе 5, конец которого погружен в жидкий азот или гелий 6 в криостате 7.

Сигналы от магниточувствительных элементов подаются в электронный блок 8, где усиливаются, фильтруются, обрабатываются и в виде цифровых кодов передаются в ПК. По командам оператора компьютер организует перемещение вдоль координат X и Y координатного стола, на котором установлен исследуемый образец, и измерение в каждом положении, т.е. в каждой точке поверхности образца проекций вектора индукции магнитного поля и/или его градиента. Таким образом на экране монитора формируется увеличенное в десятки-тысячи раз двух- или трехмерное изображение магнитного поля объекта.

На рис. 1.14 справа приведена фотография криогенной части растрового микроскопа на сквидах ССМ-77, созданного на физическом факультете МГУ. На его основе в ИЗМИРАН России с использованием "высокотемпературных" сквидов, для охлаждения которых достаточно жидкого азота, выпускается растровый микроскоп ССМ-300. Он уже

может исследовать объекты, которые находятся при обычной комнатной температуре до 300 К.

По сравнению с традиционными методами магнитной, ультразвуковой и радиографической дефектоскопии сквид-микроскопия благодаря своей сверхвысокой чувствительности позволяет обнаруживать скрытые, значительно меньшие по размерам, глубоко погруженные в материал дефекты, даже под защитным слоем. Она позволяет обнаруживать протекающие в образце коррозионные и вихревые слабые электрические токи, снимать карты токов, которые текут в многослойных электронных платах и в микросхемах. С ее помощью тестируют самые ответственные детали турбин, ракет, самолетов, осуществляют магнитные исследования геологических, минералогических, археологических образцов. Проверяют, например, подлинность купюр или важнейших документов, на которые нанесены скрытые знаки специальными магнитными чернилами и т.п. На таком микроскопе можно проводить не только пассивные, но и активные исследования, когда в исследуемом образце гальваническими, индукционными или другими методами специально возбуждают электрические или магнитные поля и изучают реакции объекта на них.

Пространственная разрешающая способность таких растровых микроскопов определяется размером магниточувствительного элемента 4 (рис. 1.14). Для достижения нанометровой разрешающей способности была разработана специальная технология формирования нанометровых сквидов на кончике зонда (англ. "SQUID on a tip"). Объясним ее с помощью рис. 1.15. Пустую внутри тонкую трубку из чистого плавленого кварца при нагревании вытягивают в сверхтонкое острие – зонд, – с двух сторон которого формируют продольные вмятины (ложбинки). После шлифовки и полировки торца на острие и на его торец в высоком вакууме напыляют алюминий, который становится сверхпроводником при температурах ниже 1,2 К.

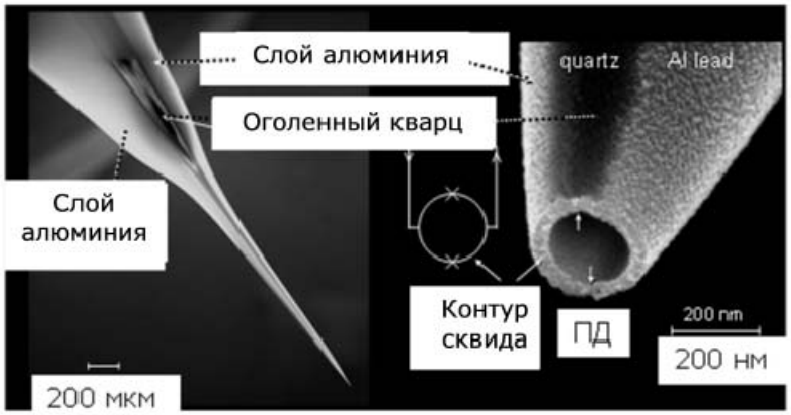


Рис. 1.15. Слева – зонд растрового сквид-микроскопа из плавленого кварца; справа с дополнительным 500-кратным увеличением – острие этого зонда, на торце которого сформован сквид с двумя переходами Джозефсона (ПД)

Напыление осуществляют под углом так, чтобы в ложбинки алюминий не попадал. Небольшой налет удаляют химическим травлением. На торце в местах выхода вмятин образуются "слабые" мостики – два перехода Джозефсона (ПД), и возникает наносквид с диаметром отверстия около 200 нм.

Типичная сигнальная кривая магнитометра постоянного тока на таком сквиде показана на рис. 1.16 слева. Период зависимости от изменения магнитного потока (в данном случае – от изменения напряженности магнитного поля) составляет здесь около 60 мТл, амплитуда изменения напряжения – около 50 мВ, критический ток – около 1,6 мкА чувствительность в рабочей точке – около 3 В/Тл.

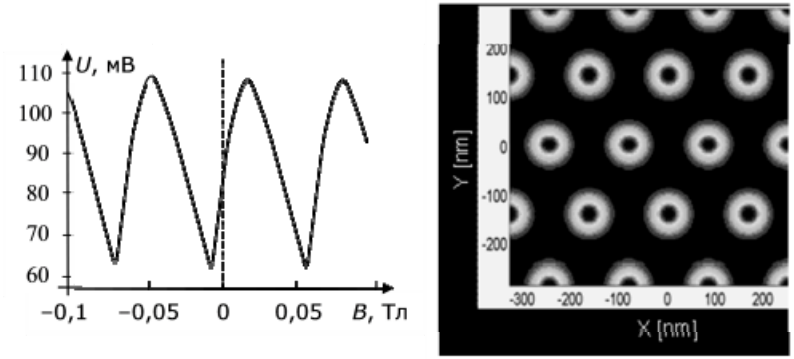


Рис. 1.16. Слева – типичная сигнальная кривая магнитометра со сквидом на торце острия диаметром 200 нм. Справа – пример изображения в растровом сквид-микроскопе, полученного с помощью такого наносквида

Справа на рис. 1.16 показан пример изображения участка экспериментального образца магнитного диска с созданной на его поверхности матрицей из наностолбиков ферромагнетика, расположенных на расстояниях 132 нм друг от друга. Как можно видеть, реальная разрешающая способность растрового микроскопа с описанным зондом существенно лучше, чем размер наносквида, и составляет около 50 нм. С его помощью удается уверенно регистрировать наночастицы с магнитным моментом до 65 магнетонов Бора (при частотном диапазоне 1 Гц).

6.7.7. Другие применения

6.7.7.1. Стандарт Вольта

Нестационарный эффект Джозефсона и формула (1.8) устанавливают точную связь между напряжением на переходе Джозефсона и частотой джозефсоновской генерации. Это было использовано в области метрологии. Очень важные для науки и техники стандарты разных физических величин имеют, к сожалению, разную точность. Наиболее точно сейчас измеряется частота, относительная точность измерения которой на высоких частотах достигает 10^{-14}. А вот точность измерений электрического напряжения, которые базировались на

эталоне из гальванических элементов, была гораздо меньше – порядка 10^{-6}. Формула (1.8) теоретически позволила заменить измерение напряжения измерением частоты. Сложность практической реализации сверхпроводящего эталона напряжения заключалась в том, что падение напряжения на одном переходе Джозефсона очень мало – меньше 1 мВ. Однако ученым удалось изготовить микросхему из примерно 1500 последовательно соединенных переходов Джозефсона, которая дает на выходе стандартное напряжение 1 В с точностью порядка 10^{-10}. Указанная точность, кстати, определяется здесь не точностью измерения частоты, а точностью, с которой физики знают сегодня значение отношения универсальных постоянных e/h. Именно сквиды с ПД позволили значительно, приблизительно в 20 раз, повысить точность определения этого отношения, а вместе с ним и точность, с которой ученым известно значение кванта магнитного потока (1.1).

6.7.7.2. Радиотехнические применения

На переходах Джозефсона строят сейчас также много различных радиотехнических устройств субмиллиметрового диапазона длины волны. Это, например, параметрические усилители для названного частотного диапазона с рекордно низкими собственными шумами (т.н. "шумовая температура" составляет здесь меньше 4 К; сравните это с тем, что СВЧ усилитель на лампах бегущей волны имеет шумовую температуру порядка 10000 К). Относительно легко реализуются на этой элементной базе генераторы и приемники сигналов субмиллиметрового диапазона, модуляторы и демодуляторы, преобразователи частоты, детекторы.

Об одном из интересных применений схем на переходах Джозефсона в вычислительной технике мы расскажем вам в следующей лекции.

Краткие итоги

Существование "критического" значения напряженности магнитного поля, при превышении которого состояние сверхпроводимости исчезает, позволило создать довольно оригинальные криоэлектронные логические вентили электрического тока, названные криотронами. Принцип действия криотронов допускает их масштабирование до

нанометровых размеров. Чем меньшие размеры криотронов, тем меньшие токи они потребляют и, соответственно, тем меньше рассеивают тепла. При размерах меньше 100 нм собственное время переключения логических схем на криотронах становится меньше 10 пс.

Очень ценные для применений квантовые свойства имеют сквиды с переходами Джозефсона (ПД). Сквид с одним ПД "реагирует" на наименьшие изменения магнитного потока, поэтому его часто используют для создания очень чувствительных сенсоров магнитного поля. Одним из наиболее распространенных является магнитометр переменного тока. На сквидах с двумя ПД строят очень чувствительные магнитометры постоянного тока. Чувствительность наилучших сверхпроводящих магнитометров достигает сейчас $2 * 10^{-21}$ Вб/$\sqrt{\text{Гц}}$. Чувствительность относительно магнитной индукции достигает 10^{-14} Тл. И очень важно, что чувствительность эта не зависит от уровня постоянной составляющей магнитного поля, т.е. совсем небольшие изменения можно измерять на фоне относительно сильного постоянного магнитного поля, например, магнитного поля Земли.

С помощью выносных рамок, индуктивно связанных с контуром сквида, можно строить магнитометры, которые измеряют все три пространственные компоненты магнитного поля, а также чувствительные градиометры 1-го, 2-го и высших порядков, которые измеряют первую, вторую или высшие производные напряженности магнитного поля по координате. С помощью наноразмерных сквидов удается измерять даже очень слабые магнитные поля, создаваемые наночастицами с магнитным моментом порядка 100 магнетонов Бора и меньше.

На базе сверхчувствительных магнитометров на сквидах создан целый ряд многоканальных интеллектуальных сенсоров. Это – магнитокардиографы, позволяющие отслеживать изменения магнитного поля, связанные с функционированием сердца, и делать на основе этого важные диагностические медицинские выводы; магнитоэнцефалографы, позволяющие регистрировать слабые переменные магнитные поля, связанные с работой мозга человека, обнаруживать имеющиеся там нарушения активности, локализовать места нарушений. Это также растровые сквид-микроскопы,

позволяющие обнаруживать скрытые, глубоко погруженные в материал магнитные дефекты, даже под защитным слоем; коррозионные и вихревые слабые электрические токи, протекающие в исследуемом образце, снимать карты токов, протекающих в многослойных электронных платах и в микросхемах, и т.д. Для достижения нанометровой разрешающей способности разработана специальная технология формирования нанометровых сквидов на торце острия зонда размером порядка 100 нм.

Используя нестационарный эффект Джозефсона, удалось создать новый стандарт Вольта для метрологии с точностью порядка 10^{-10}, а также определить отношение универсальных физических постоянных e/h с точностью, приблизительно в 20 раз лучше, чем было известно до сих пор.

На наноразмерных переходах Джозефсона строят также много разных радиотехнических устройств субмиллиметрового диапазона длин волн: параметрические усилители с рекордно низкими собственными шумами, генераторы и приемники СВЧ радиосигналов, модуляторы и демодуляторы, преобразователи частоты, детекторы, позволяющие передавать и принимать информацию с очень высокой скоростью.

6.8. Быстрая одноквантовая логика

С использованием переходов Джозефсона (ПД) предложено много вариантов логических элементов, с помощью которых можно организовать быстрые компьютерные вычисления. Мы рассмотрим здесь лишь один из наилучших вариантов, который в англоязычных публикациях называют "*Rapid Single-Flux-Quantum*" или сокращенно "*RSFQ*". В русскоязычной литературе этот вариант называют "*быстрой одноквантовой логикой*" или сокращенно "*БОК логикой*".

6.8.1. Динамические свойства перехода Джозефсона

Чтобы объяснить принципы действия БОК логики, рассмотрим сначала детальнее динамические свойства ПД. В предыдущем разделе (рис. 2.4) описана стационарная ампер-вольтная характеристика ПД,

которую можно построить при условии, что через ПД пропускается калиброванный электрический ток I. Рассмотрим теперь случай, когда кроме постоянного электрического тока смещения I_C, несколько меньшего, чем критический ток I_K, (рис. 2.1 слева) на ПД в какой-либо момент времени t_0 действует очень короткий импульс напряжения (справа вверху). Его амплитуда должна быть достаточна для того, чтобы суммарный ток через ПД превысил I_K. В интервале времени, когда ток через ПД превышает критическое значение I_K, на переходе появляется падение напряжения $2\Delta/e$ и согласно уравнению (2.6) предыдущего раздела происходит быстрое изменение фазы сверхпроводящего тока. Когда входное напряжение исчезает, электрический ток через возбужденный ПД еще относительно долго продолжает колебаться (штриховая кривая 1), постепенно затухая до стационарного значения I_C.

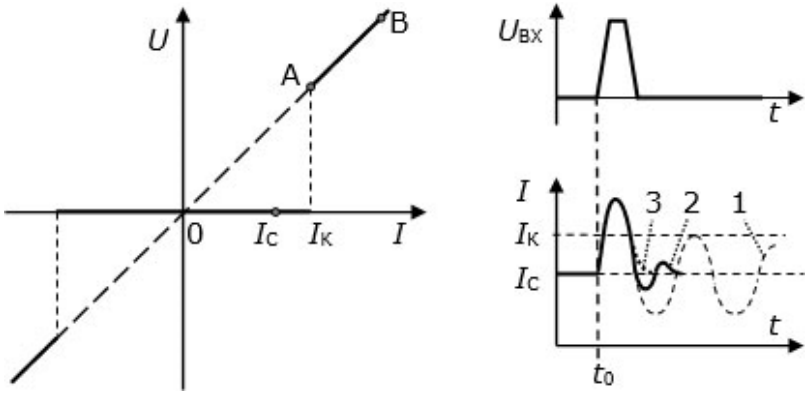


Рис. 2.1. Слева – ампер-вольтная характеристика ПД; справа вверху – короткий импульс напряжения, внизу – динамическая реакция тока через ПД на этот импульс: 1 и 2 – при отсутствии и при наличии демпфирования, 3 – при значительном демпфировании

Если ПД демпфировать, добавляя элемент, частично поглощающий энергию колебаний, то колебания затухают быстрее (кривая 2). Если же добавить элемент, который быстро поглощает энергию колебаний,

то колебания исчезают совсем (пунктирная кривая 3). Оказалось, что этого легко достичь шунтированием ПД пленочным проводником, изготовленным из не сверхпроводящего металла. Такой шунт практически не влияет на свойства ПД в сверхпроводящем состоянии, а при выходе из сверхпроводящего состояния принимает на себя лишь часть избыточного (сверх I_K) тока. Для построения логических элементов важным является то, что *шунтированный таким образом ПД переключается очень быстро – за единицы пикосекунд, а при наноразмерах – и еще быстрее.*

6.8.2. БОК триггер

Динамические свойства шунтированного ПД эффективно используются в быстрой одноквантовой логике. Опишем сначала схему и способ функционирования БОК триггера. На рис. 2.2 показана принципиальная электрическая схема т.н. R-S-триггера. Условные изображения шунтированного ПД и сверхпроводящих шин показаны справа в рамках, обведенных штриховыми линиями. Для сверхпроводящих шин такое изображение выбрано потому, что их импеданс носит в основном индуктивный характер.

Основой R-S-триггера является сквид, образованный двумя шунтированными переходами Джозефсона *ПД1* и *ПД2* и замкнутым сверхпроводящим контуром L. Как вы уже знаете, суммарный магнитный поток сквозь отверстие такого контура всегда должен быть кратным кванту магнитного потока Φ_0. Базовое состояние сквида, когда в сверхпроводящем контуре незатухающий ток I_{II} циркулирует против часовой стрелки, принимают за логический "0", а когда ток циркулирует по часовой стрелке, – за логическую "1". На сквид подается ток смещения I_C, подобранный так, чтобы оба эти состояния были стабильными.

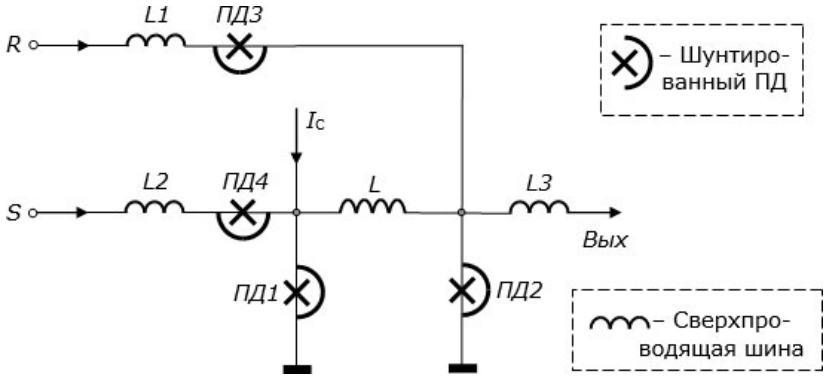


Рис. 2.2. Принципиальная схема R-S-тригера – одного из основных элементов БОК логики (справа в штриховых рамках расшифрованы условные изображения)

R-S-триггер имеет два входа, которые принято называть R– и S-входами. Вход R соединен с ПД2 через сверхпроводящую шину L1 и ПД3, а вход S соединен с ПД1 через сверхпроводящую шину L2 и ПД2. Выходной сигнал берется с перехода Джозефсона ПД2 через сверхпроводящую шину L3.

Триггер функционирует следующим образом. Пусть сначала он находится в состоянии "0". В этом состоянии через ПД1 течет ток $I_1 = 0.5I_c - I_R$, близкий к критическому току I_K, а через ПД2 течет ток $I_2 = 0.5I_c - I_R$, далекий от критического тока. Когда на вход S действует короткий импульс напряжения, вызванный им дополнительный ток не приводит к превышению критического тока в переходах Джозефсона ПД2 и ПД4, но вызывает превышение его в переходе ПД1. Из-за этого сверхпроводящее состояние здесь на короткое время разрушается, и в контур сквида "просачивается" квант магнитного потока. Направление циркуляции тока в контуре изменяется на противоположное, и сквид переходит в состояние "1". Поскольку ПД2 остается в сверхпроводящем состоянии, напряжение на выходе не появляется.

Если импульс напряжения действует на вход S, когда триггер находится в состоянии "1", то к приходу импульса через ПД1 течет ток

$I_1 = 0.5I_c - I_R$. Параметры ПД4 подобраны так, что в этом случае сверхкритический ток возникает именно в нем, однако связанный с этим квант магнитного потока действует извне и не просачивается в контур сквида. Поэтому состояние сквида в этом случае не изменяется.

Переход сквида из состояния "1" в состояние "0" может произойти лишь при поступлении импульса напряжения на вход R. В этом случае сверхкритический ток возникает на переходе ПД2. При этом из контура сквида "вытекает" квант магнитного потока, направление циркуляции тока в контуре изменяется на противоположное, и сквид переходит в состояние "0". На выходе триггера появляется кратковременный импульс напряжения.

Если ОК импульс поступает на вход R, когда триггер находится в состоянии "0", то через ПД2 течет ток $I_2 = 0.5I_c - I_R$. Параметры перехода Джозефсона ПД3 (как и ПД4) подобраны так, что в этом случае сверхкритический ток возникает именно в нем. Однако связанный с этим квант магнитного потока тоже действует извне и не просачивается в контур сквида. Поэтому состояние сквида ПД2 в этом случае не изменяется, и на выходе не возникает импульс напряжения.

Как видим, *ОК импульс может появиться на выходе R-S-триггера только в состоянии "1" при появлении одноквантового импульса на его входе R*. Это и используется при построении схем БОК логики.

Если ни на один из входов импульс не поступает, то состояние сквида не изменяется и на выходе ОК импульс не появляется. Это называют "режимом хранения информации". Одновременное поступление импульсов на оба входа считается запрещенным, так как следующее состояние сквида становится в таком случае неопределенным.

Вследствие закона Джозефсона (3.6) между импульсом напряжения на ПД и квантом магнитного потока сквозь замкнутый контур существует соотношение

$$\int U(t)dt = \Phi_0 = 2.07 \text{ мВ*пс}. \qquad (2.1)$$

В этом смысле (по величине "площади" импульса) выходной импульс напряжения всегда "стандартный". Его принято называть *одноквантовым импульсом напряжения* или сокращенно *ОК импульсом*. Он является вполне достаточным для надежного переключения следующих звеньев в логической сети обработки цифровой информации.

6.8.3. Основные схемы БОК логики

6.8.3.1. Принципы организации обработки информации в БОК схемах

Основные принципы кодирования и организации цифровой обработки информации в БОК схемах таковы:

а) используются одноквантовые тактовые импульсы, которые делят все время работы схем на такты;

б) появление на входе или на выходе схемы одноквантового импульса в течение такта рассматривается как логическая "1", а отсутствие импульса – как логический "0";

в) одноквантовый импульс, который появляется на выходе схемы в начале следующего такта, является логической функцией от ОК импульсов, действовавших на входах схемы в предыдущем такте.

Как видим, БОК логика – это тактовая импульсная логика.

6.8.3.2. D-элемент

На рис. 2.3 показана БОК схема, выполняющая простейшую одновходовую логическую функцию "Тождество" (повторитель). Она почти совпадает со схемой, изображенной на рис. 2.2, но прежний вход R стал входом T, на который подаются тактовые ОК импульсы.

Временная диаграмма работы схемы показана справа вверху. Вдоль горизонтали здесь отложено время, вдоль вертикали – напряжение.

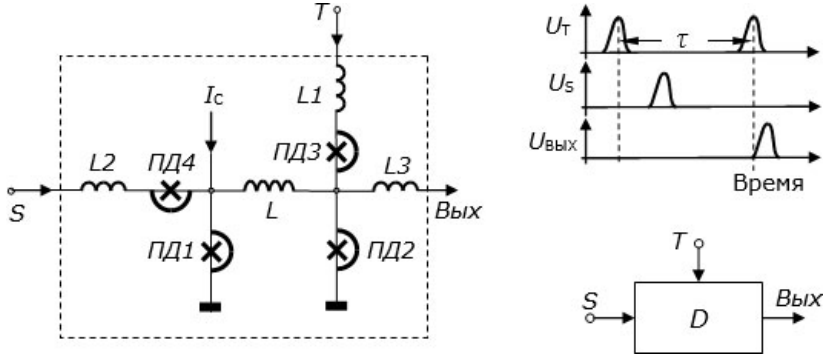


Рис. 2.3. Слева – принципиальная схема D–элемента БОК логики. Справа внизу – его обозначение в последующих схемах. Справа вверху – временная диаграмма работы элемента

Тактовые ОК импульсы U_T поступают на вход T с периодом T. Чтобы на входах S и T импульсы не могли появиться одновременно, период T должен быть больше, чем время распространения импульса между последовательно соединенными логическими элементами.

На вход S сигнальный импульс U_S может поступить в любой момент такового периода. Если он поступил, то, как описано выше, в контур сквида (с шунтированными переходами Джозефсона ПД1 и ПД2) просачивается квант магнитного потока, и сквид переходит из состояния "0" в состояние "1". Когда поступает следующий тактовый импульс, он кратковременно выводит ПД2 из сверхпроводящего состояния, из контура "вытекает" один квант магнитного потока, и сквид возвращается в состояние "0". При этом на выходе формируется одноквантовый импульс напряжения.

Если за период такта сигнальный импульс U_S на вход S не действует, сквид остается в состоянии "0". И когда на вход T поступает следующий тактовый импульс, он кратковременно выводит из сверхпроводящего состояния уже не ПД2, а ПД3. Связанный с этим квант магнитного потока в контур сквида не просачивается, и сквид

остается в состоянии "0". Сигнал на выходе не формируется. Таким образом, эта схема действует как повторитель сигнала на входе с задержкой на 1 такт.

Обведенная штриховой рамкой схема является базовой, так как используется во многих других схемах БОК логики. Ее принято называть "*D-элементом*". Дальше она условно будет обозначаться так, как показано справа внизу.

6.8.3.3. БОК инвертор

БОК схема, выполняющая другую простейшую одновходовую логическую функцию "Отрицание", показана на рис. 2.4.

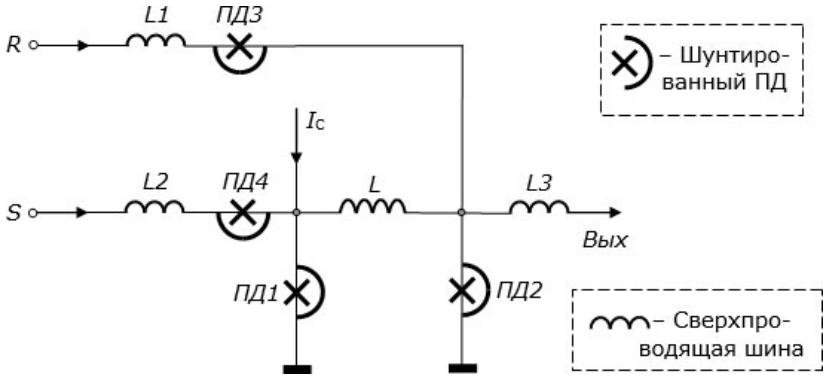


Рис. 2.4. Слева – принципиальная схема БОК инвертора. Справа внизу – его обозначение в схемах. Справа вверху – временная диаграмма работы БОК инвертора

От D-элемента она отличается тем, что здесь использован дополнительный шунтированный переход Джозефсона ПД5 и выход схемы присоединен именно к этому ПД. Справа вверху показана временная диаграмма работы этой схемы. В начале такта сквид (ПД1 и ПД2) этой схемы всегда находится в состоянии "0". Если за период такта на вход S действует сигнальный ОК импульс , то сквид переходит из состояния "0" в состояние "1". Когда поступает следующий тактовый импульс, он кратковременно выводит ПД2 из сверхпроводящего состояния, а ПД5 остается в сверхпроводящем

состоянии. Поэтому на выходе схемы импульс напряжения не формируется. Если же за период такта сигнальный ОК импульс на вход S не действует, то сквид остается в состоянии "0". И когда на вход T поступает следующий тактовый импульс, он кратковременно выводит из сверхпроводящего состояния именно дополнительный переход Джозефсона ПД5. В результате на выходе схемы формируется одноквантовый импульс напряжения.

6.8.3.4. БОК схемы дизъюнкции и конъюнкции

На рис. 2.5 показаны принципиальные схемы элементов БОК логики, выполняющие двухвходовые логические операции дизъюнкции и конъюнкции.

Схема слева состоит из двух D-элементов D1 и D2, выходы которых присоединяются к симметричным "плечам" выходной цепи. Одно плечо образовано переходами Джозефсона ПД1 и ПД2 и сверхпроводящей шиной L1, второе – переходами Джозефсона ПД3 и ПД4 и сверхпроводящей шиной L2. На каждое плечо подается ток смещения (и). Если за период такта ни на один из входов S1 и S2 не действует сигнальный ОК импульс , то на выходах D-элементов D1 и D2 во время следующего тактового импульса одноквантовый импульс не возникает. Соответственно не возникает он и на выходе всей схемы.

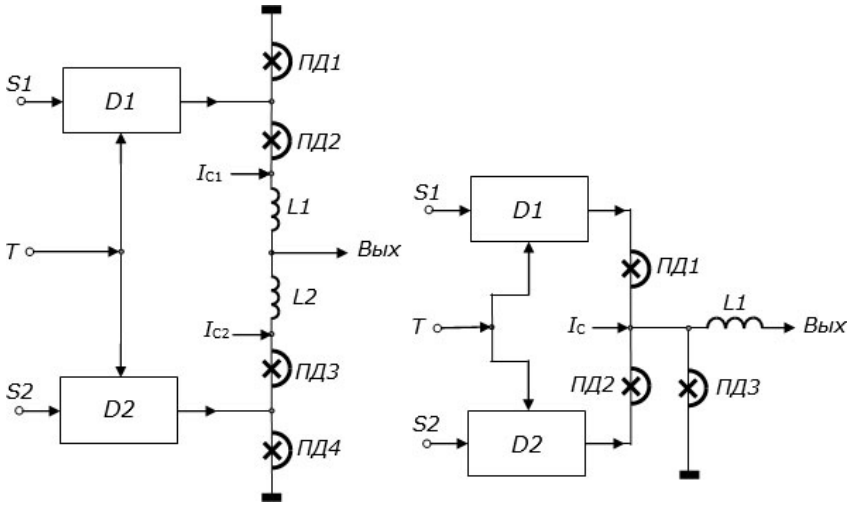


Рис. 2.5. БОК логические схемы с двумя входами: слева – дизъюнкция, справа – конъюнкция

Если же за период такта хотя бы на один из входов S1 или S2 действует сигнальный ОК импульс U_S, то на выходе соответствующего D-элемента D1 или D2 во время следующего тактового импульса формируется одноквантовый импульс. Это приводит к кратковременному выходу из состояния сверхпроводимости перехода Джозефсона ПД1 или ПД2. И на выходе схемы формируется одноквантовый импульс напряжения. В то же время квант магнитного потока, который генерируется, не просачивается в контур соответствующего сквида, и поэтому не влияет на его состояние.

Если на протяжении такта сигнальные ОК импульсы U_S действуют на оба входа S1 и S2, то на выходе обоих D-элементов D1 и D2 во время следующего тактового импульса формируются одноквантовые импульсы. Это приводит к кратковременному выходу из состояния сверхпроводимости обоих переходов ПД1 и ПД2. Поскольку это происходит одновременно, то на выходе схемы формируется один ОК импульс напряжения.

В схеме конъюнкции, изображенной справа, во время следующего тактового импульса на выходе ОК импульс напряжения появляется

лишь в том случае, если он формируется одновременно на выходах обоих D-элементов D1 и D2. Лишь в этом случае удается вывести из сверхпроводящего состояния переход ПД3. Когда ОК импульс напряжения появляется на выходе лишь одного из D-элементов D1 или D2, то из сверхпроводящего состояния кратковременно он выводит только один переход Джозефсона ПД1 или ПД2. Но это не приводит к формированию одноквантового импульса напряжения на выходе, так как переход ПД3 остается в сверхпроводящем состоянии. Сигнал на выходе схемы тем более не формируется, если за период такта сигнальный ОК импульс не действует на ни один из входов S1 и S2.

6.8.3.5. Генератор и формирователь тактовых БОК импульсов

На рис. 2.6.а показана принципиальная схема генерирования последовательности тактовых ОК импульсов напряжения, необходимых для работы БОК логики. Схема представляет собой сквид, через который пропускают постоянный ток смещения. Этот ток и параметры элементов подобраны так, что через ПД3 течет ток, немного меньший критического. Кроме того, на сквид извне подают переменный ток синхронизации, который задает частоту и фазу последовательности тактовых импульсов.

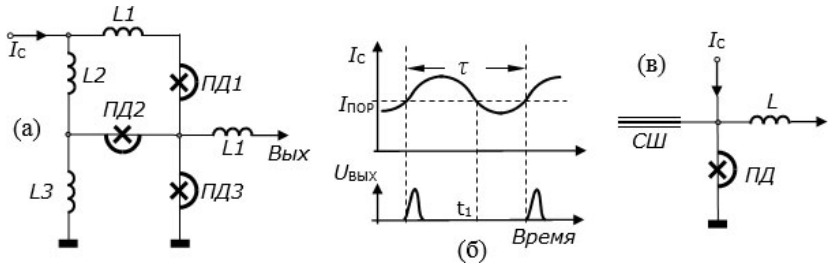


Рис. 2.6. (а) Принципиальная схема генерирования тактовых ОК импульсов напряжения. (б) Временные диаграммы. (в) Формирователь импульсов

Временная зависимость суммарного входного тока синхронизации IC показана на рис. рис. 2.6.б вверху. В момент, когда ток синхронизации при нарастании превышает пороговое значение $I_{пор}$, ток через ПД3 превышает критическое значение, и ПД3 кратковременно выходит из

состояния сверхпроводимости. На выходе формируется пикосекундный ОК импульс напряжения. А в контуре сквида (L3–ПД2–ПД3) направление циркуляции сверхпроводящего тока изменяется на противоположное. При этом ток через ПД1 и ПД2 возрастает выше критического, так как циркулирующий в контуре ток прибавляется к I_C. И на определенное время эти переходы выходят из сверхпроводящего состояния.

Потом ток синхронизации IC начинает уменьшаться, и в момент t_1, когда он становится меньше порогового значения $I_{пор}$, ток через ПД1 и ПД2 становится меньше критического, и эти переходы возвращаются в сверхпроводящее состояние. Это приводит к изменению направления циркуляции тока в сквиде, который возвращается в исходное состояние. В следующем периоде, когда ток синхронизации I_C опять начинает возрастать, все повторяется.

Ток синхронизации I_C не обязательно должен быть синусоидальным, он может иметь любую другую форму, главное, чтобы он при уменьшении и при последующем возрастании периодически проходил через пороговое значение $I_{пор}$.

Когда тактовые ОК импульсы напряжения передаются вдоль относительно длинных сверхпроводящих шин, они постепенно "размываются" – становятся ниже и заметно растягиваются во времени. (Напомним, что речь идет о пикосекундных импульсах). Чтобы восстановить их форму, используется формирователь импульсов, показанный на рис. 2.6,в. Ток смещения I_C здесь лишь немного меньше, чем критический ток. Поэтому, когда поступает даже "размытый" тактовый импульс, ток через переход Джозефсона ПД почти сразу превышает критический, и на выходе формируется короткий ("острый") одноквантовый импульс напряжения нужной формы. После этого ПД снова возвращается в сверхпроводящее состояние.

6.8.4. Преимущества наноэлектронной элементной базы БОК логики

Время задержки сигналов в БОК схемах не превышает 3τ, где

$$\tau_0 \approx \sqrt{\frac{C}{I_K}}. \quad (2.2)$$

I_K – критический ток перехода Джозефсона, C – его емкость, которая пропорциональна площади контакта. Значение I_K для обеспечения флуктуационной стабильности при температуре порядка 4,2 К выбирают на уровне 100 мкА и выше.

Наиболее употребительная технология с использованием ниобия при размерах ПД порядка 3 х 3 мкм^2 обеспечивает значение $\tau_0 \approx 3$пс. БОК логические схемы могут работать в этом случае с тактовой частотой порядка 100 ГГц. С уменьшением размеров ПД время задержки быстро уменьшается и при размерах порядка 200 х 200 нм^2 достигает насыщения

$$\tau_{0MIN} \approx \frac{h}{\Delta}, \quad (2.3)$$

где h – постоянная Планка, Δ– полуширина "запрещенной зоны" (энергетической "щели") куперовской пары электронов. Для сверхпроводника из ниобия $\tau_{0MIN} \approx 0,5$пс. При использовании сверхпроводников с более широкой запрещенной зоной куперовских пар электронов τ_{0MIN} удается уменьшить даже до 0,1 пс.

Поэтому *при использовании наноразмерных переходов Джозефсона БОК логические схемы могут работать с тактовой частотой порядка 700 ГГц и даже свыше 1 ТГц.*

Вдобавок, как оказалось, наноразмерные ПД (размерами меньше, чем 300x300 нм^2) уже не нуждаются во внешнем шунтировании. Их собственное сопротивление является уже достаточным для демпфирования. А это заметно уменьшает площадь, требуемую для размещения логических схем, содействует повышению поверхностной плотности их интеграции и упрощает технологию изготовления.

Рассеяние энергии в БОК логических схемах определяется не диссипацией энергии внутри ПД (обычно она меньше, чем 10^{-18} Дж), а диссипацией энергии на резисторах, через которые подается ток смещения. Эти потери энергии оцениваются величиной порядка 1 мкВт на одну логическую схему. Поэтому *рассеяние тепла в большой интегральной БОК схеме, состоящей даже из* 10^5 *логических вентилей не превышает 100 мВт*.

Технологические допуски на параметры элементов БОК логики довольно широки. Наиболее жестким требованием является обеспечение заданного значения критического тока через ПД с точностью порядка $\pm 5\%$. Для других параметров допуски составляют $\pm 20 - 30\%$, что позволяет достигать высокого выхода при изготовлении даже очень сложных логических схем. Таких, например, как процессоры для выполнения сверхбыстрых арифметических действий над многоразрядными числами с плавающей запятой (порядка 30 млрд. операций/с) или сверхбыстрые блоки оперативной памяти (частота записи и считывания порядка 100 Гбайт/с).

С использованием БОК логики уже реализованы также:

- высокоточные сверхбыстродействующие аналого-цифровые и цифро-аналоговые преобразователи (свыше 10^{13} преобразований в секунду),
- сверхскоростные телекоммуникационные коммутаторы,
- многоканальные БОК магнитометры для магнитной томографии мозга человека (фирма Conductus, США),
- 16-канальный автокоррелятор и другие важные сверхскоростные устройства.

У БОК устройств с низким энергопотреблением, как и у всех устройств на основе сверхпроводимости, есть лишь один недостаток, – они работают только при криогенных температурах. И во многих случаях на это идут.

С использованием переходов Джозефсона (ПД) предложено много вариантов логических элементов, с помощью которых можно организовать быстрые компьютерные вычисления. Одним из интереснейших вариантов является быстрая одноквантовая логика (БОК логика). В ней используют шунтированные переходы Джозефсона (ШПД), в которых параллельно ПД включен шунт, например, в виде пленки из не сверхпроводящего металла. ШПД переключается очень быстро – за единицы пикосекунд, а при наноразмерах – даже за доли пикосекунды.

Квантовые свойства ШПД приводят к тому, что при его переключении на выходе появляются стандартные пикосекундные импульсы напряжения в том смысле, что $\int U(t)dt = \Phi_0 = 2.07$ мВ*пс. Их принято называть ОК импульсами – одноквантовыми импульсами напряжения. Они оказались достаточными для переключения последующих звеньев в логической сети обработки цифровой информации.

Эта обработка в БОК схемах базируется на следующих принципах: (а) используются ОК тактовые импульсы, которые делят все время работы схем на такты; (б) появление на входе или выходе схемы одноквантового импульса рассматривается как логическая "1", а отсутствие импульса – как логический "0"; (в) одноквантовый импульс, который появляется на выходе схемы в начале следующего такта, является логической функцией от ОК импульсов, действовавших на входах схемы в предыдущем такте.

Базовым "кирпичиком" БОК логики является "D-элемент", который состоит из сверхпроводящего контура с двумя ШПД и еще двух ШПД, последовательно подключенных ко входу тактового и ко входу сигнального импульсов соответственно. С использованием "D-элементов" и дополнительных ШПД построены логические схемы дизъюнкции и конъюнкции, инвертор, триггеры, и т.п.

Генератор тактовых ОК импульсов строится на трех ШПД, частота генерации задается внешним источником переменного тока, который при уменьшении и следующем нарастании периодически проходит через пороговое значение тока.

Когда ОК тактовые импульсы напряжения передаются вдоль относительно длинных сверхпроводящих шин, они постепенно "размываются". Чтобы восстановить их форму, используется формирователь импульсов, построенный на одном ШПД и превращающий входной "размытый" импульс в короткий ("острый") одноквантовый импульс напряжения.

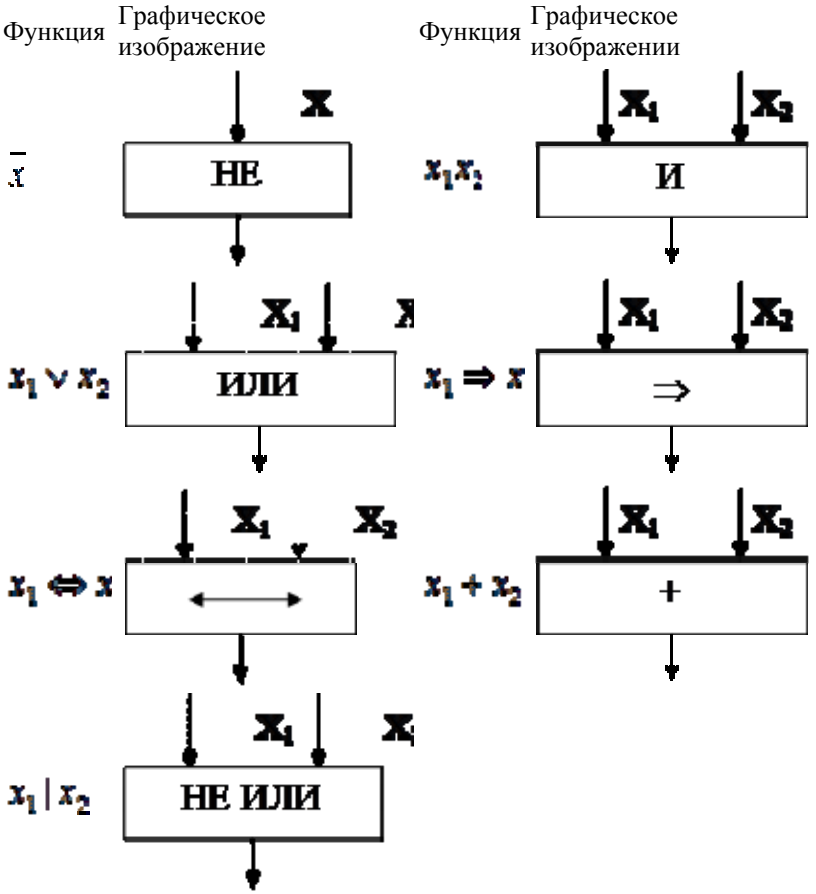
При микронных размерах время задержки сигналов в БОК схемах составляло несколько пикосекунд и уменьшалось с уменьшением размеров ШПД. При размерах меньше 300 х 300 нм^2 время задержки достигает насыщения и зависит лишь от полуширины "запрещенной зоны" куперовской пары электронов. Для сверхпроводника из ниобия, например, время задержки составляет приблизительно 0,5 пс. При использовании сверхпроводников с более широкой "запрещенной зоной" его удается уменьшить даже до 0,1 пс. Тогда БОК логические схемы могут работать с тактовой частотой выше 1 ТГц. Оказалось, что наноразмерные ПД уже не нуждаются во внешнем шунтировании, так как их собственное электрическое сопротивление становится достаточным для демпфирования. Это уменьшает площадь, требуемую для размещения логических схем, содействует повышению поверхностной плотности их интеграции и упрощает технологию изготовления.

С использованием БОК логики реализованы процессоры для выполнения сверхбыстрых арифметических действий над многоразрядными числами с плавающей запятой (порядка 30 млрд. операций/с), сверхбыстрые блоки оперативной памяти (частота записи и считывания порядка 100 Гбайт/с), а также высокоточные сверхбыстродействующие цифро-аналоговые и аналого-цифровые преобразователи (свыше 10^{13} преобразований в секунду), сверхскоростные телекоммуникационные коммутаторы, многоканальные БОК магнитометры для магнитной томографии мозга человека и другие важные сверхскоростные устройства.

7. Логические схемы

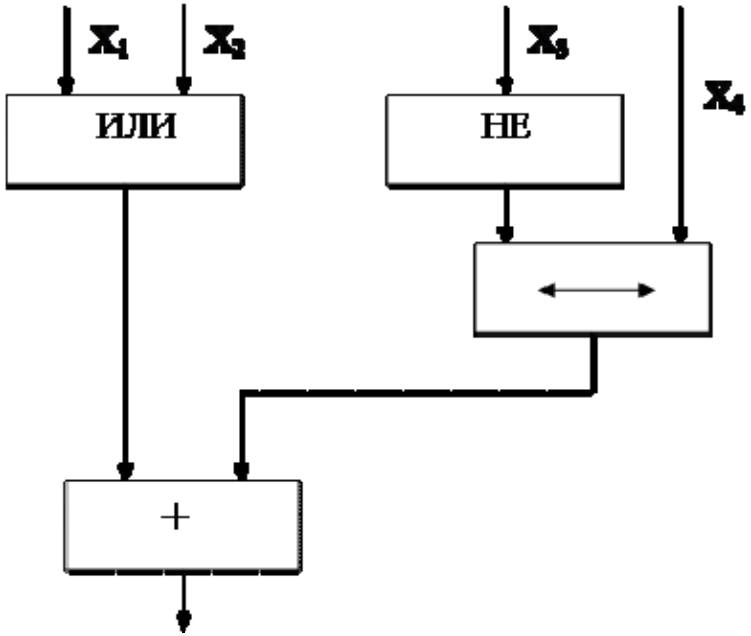
7.1. Логические элементы элементарных булевых функций.

Устройства, реализующие элементарные булевы функции, называются **логическими элементами**. Логические элементы изображаются в виде прямоугольников, внутри которых помещаются условные названия или символы соответствующих функций $f(x_1, x_2, \ldots, x_n)$:



Из данных логических элементов путем соединения входа одного из них с выходом другого можно строить все более сложные логические схемы. Для полученных таким образом схем легко записывают соответствующие им булевы функции.

Например, схема

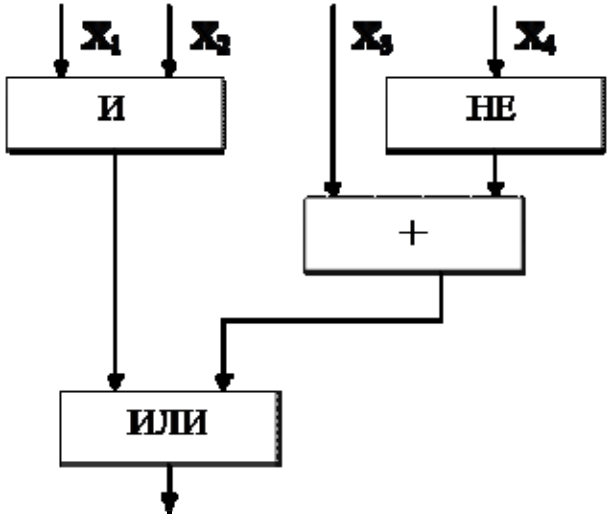


реализуется булевой функцией

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) + (\overline{x_3} \Rightarrow x_4)$$

Нетрудно для любой булевой функции построить реализующую ее логическую схему.

Например, булева функция $f(x_1, x_2, x_3) = x_1 x_2 \vee (\overline{x_3} + x_2)$ реализуется логической схемой



Двоичный сумматор

Рассмотрим построение логической схемы на примере одноразрядного сумматора, выполняющего арифметическое сложение двоичных чисел x_k и y_k, k-го разряда и переноса из младшего разряда P_{k-1}.

Пусть S_k – получаемая сумма, а P_k – перенос в старший разряд, тогда получаем следующую таблицу истинности такого сумматора.

x_k	y_k	P_{k-1}	S_k	P_k

Отсюда получаем

$$S_k = \overline{x_k y_k} P_{k-1} \vee \overline{x_k} y_k \overline{P_{k-1}} \vee x_k \overline{y_k} \overline{P_{k-1}} \vee x_k y_k P_{k-1};$$

$$P_k = \overline{x_k} y_k P_{k-1} \vee x_k \overline{y_k} P_{k-1} \vee x_k y_k \overline{P_k} \vee x_k y_k P_k = x_k y_k \vee (x_k \vee y_k) P_{k-1}$$

.

Построим схему, соответствующую данному сумматору.

Для этого вначале упростим выражение для S_k. Как легко заметить, выражение для S_k не упрощается, при использовании предыдущих методов. Для упрощения выражения функции S_k используем выражение функции P_k.

Поэтому будем рассматривать P_i как переменную величину. В результате получаем следующую таблицу, которая содержит избыточные наборы переменных:

x_k	y_k	P_{k-1}	S_k	P_k

Отсюда

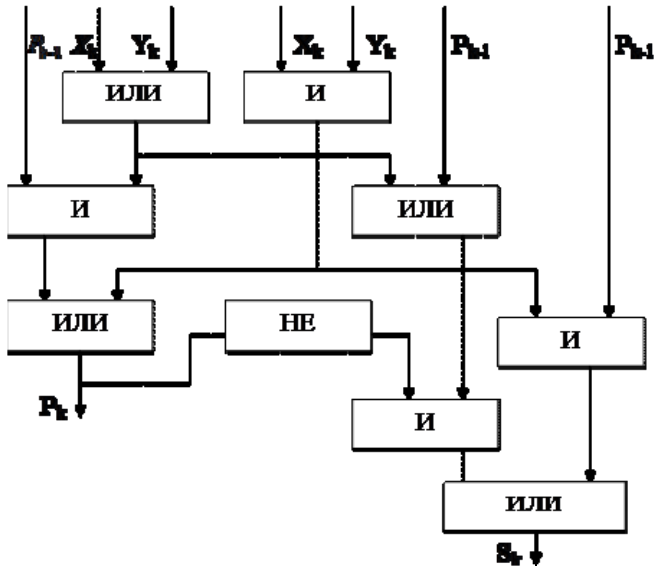
$$S_k = \overline{x_k y_k} P_{k-1} \overline{P_k} \vee \overline{x_k} y_k \overline{P_{k-1} P_k} \vee x_k \overline{y_k P_{k-1} P_k} \vee x_k y_k P_{k-1} P_k.$$

Используя методы, которые будут рассмотрены в позже, нетрудно упростить выражение для S_k:

$$S_k = x_k \overline{P_k} \vee y_k \overline{P_k} \vee P_{k-1} \overline{P_k} \vee x_k y_k P_{k-1} =$$
$$= (x_k \vee y_k \vee P_{k-1}) \overline{P_k} \vee x_k y_k P_{k-1};$$

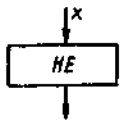
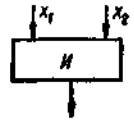
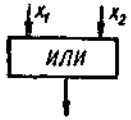
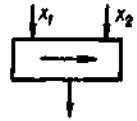
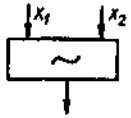
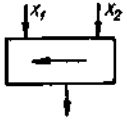
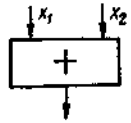
где $P_k = x_k y_k \vee (x_k \vee y_k) P_{k-1}$.

Теперь строим логическую схему:

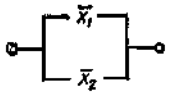
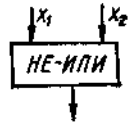
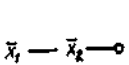
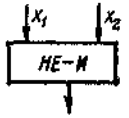


Таблица

Логические элементы, реализующие элементарные булевы функции

Функция	Нормальная форма	Контактная схема	Графическое изображение элемента	Название элемента
Отрицание $\bar{x}$	$\bar{x}$			Инвертор
Конъюнкция x_1x_2	x_1x_2			Совпадение
Дизъюнкция $x_1 \vee x_2$	$x_1 \vee x_2$			Разделение
Импликация $x_1 \to x_2$	$\bar{x}_1 \vee x_2$			Разделение с запретом
Эквиваленция $x_1 \sim x_2$	$x_1x_2 \vee \bar{x}_1\bar{x}_2$			Равнозначность
Отрицание импликации $x_1 \leftarrow x_2$	$x_1\bar{x}_2$			Совпадение с запретом
Сумма по модулю 2 $x_1 + x_2$	$\bar{x}_1x_2 \vee x_1\bar{x}_2$			Неравнозначность

Продолжение табл.

Функция	Нормальная форма	Контактная схема	Графическое изображение элемента	Название элемента
Штрих Шеффера x_1/x_2	$\bar{x}_1 \vee \bar{x}_2$			Разделение с двумя запретами
Стрелка Пирса $x_1 \downarrow x_2$	$\bar{x}_1\bar{x}_2$			Совпадение с двумя запретами

7.2. Логические схемы

Подобно суперпозиции функций логические схемы образуются *суперпозицией элементов* посредством объединения их внешних узлов (полюсов). При этом множество всех узлов схемы разбивается на *входные, выходные* и *внутренние узлы.* Например, на рис.1, *а* показана схема, реализующая функцию

$y = (x_1/x_2) + (\bar{x}_3 \to x_1)$, которая имеет три входных, один выходной и три внутренних узла.

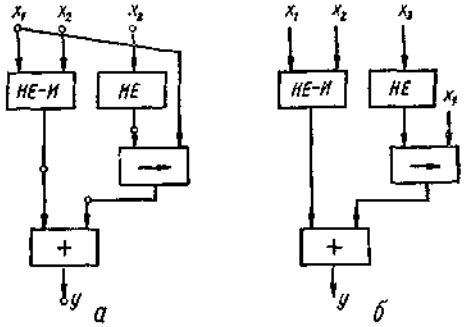


Рис. 1. Логическая схема *(а)* и ее упрощенное изображение (б).

Обычно для упрощения узлы на схемах не изображаются и во избежание излишних пересечений входы рассредоточиваются с указанием связанных с ними переменных (рис.1, *б).*

Корректно построенные схемы должны удовлетворять следующим условиям:

1) не допускать замкнутых контуров, которые могут привести к неоднозначности сигналов на входах элементов;

2) любой вход элемента должен быть связан только С ОДНИМ входом схемы или выходом другого элемента;

3) выходы элементов, не являющиеся выходами схемы и не связанные со входами других элементов, считаются лишними и исключаются из схемы.

Не составляет большого труда записать булеву функцию для данной логической схемы. Так же просто строится логическая схема для данного аналитического выражения булевой функции. Однако задача проектирования логических схем состоит в том, чтобы обеспечить наиболее экономичную реализацию булевой функции в некотором базисе, который обусловлен имеющимся в распоряжении инженера набором логических элементов или выбирается по соображениям наибольшей простоты реализации данного класса функций.

7.3. Реализация в различных базисах.

Прежде всего исходная функция преобразуемся к такому виду, чтобы она представляла собой суперпозицию только тех функций, которые входят в данный базис. Например, в базисе, состоящем из отрицания, конъюнкции и дизъюнкции, функция из (7.2) преобразуется к виду $y = (x_1/x_2) +$

$+ (\bar{x}_3 \to x_1) = \overline{x_1x_2} + (x_3 \vee x_1) \quad (\overline{x_1x_2} \vee x_3 \vee x_1)\,\overline{(x_1x_2 \vee x_3 \vee x_1)}.$

Ее реализация в системе базисных элементов {*НЕ, И, ИЛИ*} показана на рис. 2, *а.*

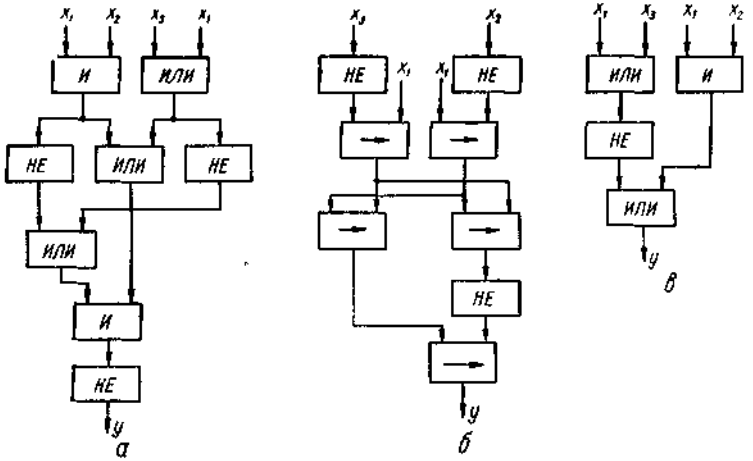


Рис 211. Логические схемы, реализующие функцию $y = (x_1/x_2) + (\bar{x}_3 \to x_1)$:

а — в базисе *{НЕ, И, ИЛИ}; б —в* базисе *{НЕ, →}; в* — упрощенная схема в базисе *{НЕ, И, ИЛИ}.*

Если в качестве базиса приняты отрицание и импликация, то функцияпреобразуется по формулам: $x_1 \vee x_2 = \bar{x}_1 \to x_2$; $x_1 x_2 = \overline{x_1 \to \bar{x}_2}$; $x_1/x_2 = x_1 \to \bar{x}_2$; $x_1 \downarrow x_2 = \overline{\bar{x}_1 \to x_2}$; $x_1 \sim x_2 = (x_1 \to \bar{x}_2) \to \bar{x}_1 \to x_2$; $x_1 + x_2 = (\bar{x}_1 \to \bar{x}_2) \to x_1 \to x_2 = (x_2 \to x_1) \to \overline{x_1 \to x_2}$.

Так, для рассматриваемого примера имеем:

$$y = (x_1/x_2) + (\bar{x}_3 \to x_1) =$$
$$= (x_1 \to \bar{x}_2) + (\bar{x}_3 \to x_1) = ((\bar{x}_3 \to x_1) \to (x_1 \to \bar{x}_2)) \to \overline{(x_1 \to \bar{x}_2) \to (\bar{x}_3 \to x_1)}.$$

Соответствующая логическая схема в базисе $\{НЕ, \to\}$ изображена на рис. 2, *б*.

Аналогично реализуются схемы и в других базисах. Как правило, в практике используются неминимальные базисы, так как минимальные базисы не всегда обеспечивают наиболее экономичную реализацию булевых функций.

7.4. Упрощение формул.

Между формулой, выражающей булеву функцию, и функциональной схемой, реализующей эту функцию, имеется функциональное соответствие. Однако, поскольку одна и та же функция может быть

выражена различными формулами, ее реализация неоднозначна. Всегда можно построить много различных логических схем, соответствующих данной логической функции. Такие схемы называют *эквивалентными.*

Из множества эквивалентных схем можно выделить наиболее экономичную или хотя бы достаточно простую схему путем упрощения формулы, соответствующей данной функции. Обычно принято считать более простыми те формулы, которые содержат меньшее количество вхождений переменных и символов логических операций. Задача упрощения аналитических выражений решается в конкретном базисе с помощью тождественных преобразований. Чаще всею эту задачу связывают с базисом, состоящим из отрицания, дизъюнкции и конъюнкции, коюрый будем называть *булевым базисом.* После того как формула выражена через основные операции, она упрощается на основании тождеств булевой алгебры.

Например, функция из (7.3) упрощается следующим образом: $y = (x_1/x_2) + (\bar{x}_3 \to x_1) = \overline{x_1 x_2} + (x_3 \vee x_1) = x_1 x_2 (x_3 \vee x_1) \vee \overline{x_1 x_2} x_3 \vee \overline{x_1} = x_1 x_2 \vee \overline{x_1 x_2}\,(x_3 \vee x_1) = x_1 x_2 \vee \overline{x_1} \vee x_3$ Соответствующая логическая схема показана на рис. 2, в.

7.5. Минимальные формы.

Как было показано ранее, любая булева функция представима в совершенной нормальной форме (дизъюнктивной или конъюнктивной). Более того, такое представление является первым шагом перехода от табличного задания функции к ее аналитическому выражению. В дальнейшем будем исходить из дизъюнктивной формы, а соответствующие результаты для конъюнктивной формы получаются на основе принципа двойственности.

Каноническая задача синтеза логических схем в булевом базисе сводится к минимизации булевых функций, т. е. к представлению их в дизъюнктивой нормальной форме, которая содержит наименьшее число букв (переменных и их отрицаний). Такие формы *называют минимальными.* При каноническом синтезе предполагается, что на входы схемы подаются как сигналы x_i, так и их инверсии $\bar{x}_i$.

Формула, представленная в дизъюнктивной нормальной форме, упрощается многократным применением *операции склеивания* $ab \vee a\bar{b} = a$ и *операций поглощения* $a \vee ab = a$ и $a \vee \bar{a}b = a \vee b$ (дуальные тождества для конъюнктивной нормальной формы имеют вид: $(a \vee b)(a \vee \bar{b}) = a$; $a(a \vee b) = a$ и $a(\bar{a} \vee b) = ab$).

Здесь под *a* и *b* можно понимать любую формулу булевой алгебры. В результате приходим к такому аналитическому выражению, когда дальнейшие преобразования оказываются уже невозможными, т. е. получаем *тупиковую форму.*

Среди тупиковых форм находится и минимальная дизъюнктивная форма, причем она может быть не единственной. Чтобы убедиться в том, что данная тупиковая форма является минимальной, необходимо найти все тупиковые формы и сравнить их по числу входящих в них букв.

Пусть, например, функция задана в совершенной нормальной дизъюнктивной форме: $y = \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$. Группируя члены и применяя операцию склеивания, имеем

$y = (\bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3) \vee (x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3) \vee x_1x_2x_3 = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_2x_3$. При другом способе группировки получим $y = \bar{x}_1x_2\bar{x}_3 \vee (\bar{x}_1x_2x_3 \vee x_1x_2x_3) \vee (x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3) = \bar{x}_1x_2\bar{x}_3 \vee x_2x_3 \vee x_1\bar{x}_2$. Обе тупиковые формы не являются минимальными. Чтобы получить минимальную форму, нужно догадаться повторить в исходной формуле один член (это всегда можно сделать, так как $x \vee x = x$). В первом случае таким членом может быть $\bar{x}_1x_2x_3$. Тогда $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee (x_1x_2x_3 \vee \bar{x}_1x_2x_3) = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_2x_3$.

Добавив член $x_1\bar{x}_2x_3$, получим:

$y = x_1\bar{x}_2 \vee x_1\bar{x}_2 \vee (x_1x_2x_3 \vee x_1\bar{x}_2x_3) = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_3$.

Перебрав все возможные варианты, можно убедиться, что две последние формы являются минимальными.

Работа с формулами на таком уровне подобна блужданию в потемках. Процесс поиска минимальных форм становится более наглядным и целеустремленным, если использовать некоторые графические и аналитические представления и специально разработанную для этой цели символику.

7.6. Многомерный куб.

Каждой вершине *n*-мерного куба, можно поставить в соответствие конституенту единицы. Следовательно, подмножество отмеченных вершин является отображением на *n*-мерном кубе булевой функции от *n* переменных в совершенной дизъюнктивной нормальной форме. На рис. 3 показано такое отображение для функции из (7.5).

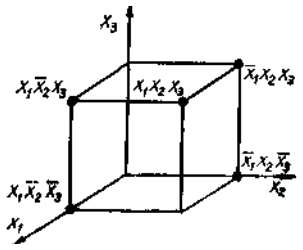


Рис.3. Отображение на трехмерном кубе функции, представленной в совершенной дизъюнктивной нормальной форме

Для отображения функции от *n* переменных, представленной в любой дизъюнктивной нормальной форме, необходимо установить соответствие между ее минитермами и элементами *n*-мерного куба. Минитерм (*n*— 1)-го ранга φ_{n-1} можно рассматривать как результат склеивания двух минитермов *n*-го ранга (конституент единицы), т. е. $\varphi_{n-1} = \varphi_{n-1}x_i \vee \varphi_{n-1}\bar{x}_i$. На *n*-мерном кубе это соответствует замене двух вершин, которые отличаются только значениями координаты x_i, соединяющим эти вершины ребром (говорят, что ребро *покрывает* инцидентные ему вершины). Таким образом, минитермам *(n* — 1)-го порядка соответствуют ребра *n*-мерного куба. Аналогично устанавливается соответствие минитермов (*n* — 2)-го порядка граням *n*-мерного куба, каждая из которых покрывает четыре вершины (и четыре ребра). Элементы *n*-мерного куба, характеризующиеся *s* измерениями, называют *s-кубами.* Так, вершины являются 0-кубами, ребра — 1-кубами, грани — 2-кубами и т.д. Обобщая приведенные рассуждения, можно считать, что миитерм *(n* - s)-го ранга в дизъюнктивной нормальной форме для функции *n* переменных отображается s-кубом, причем каждый s-куб покрывает все те s-кубы низшей размерности, которые связаны только с его вершинами. В качестве примера на рис. 4 дано отображение функции трех переменных $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_3$.

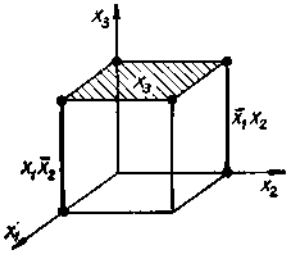


Рис.4. Покрытие функции $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_3$ совокупностью s-кубов

Здесь минитермы $\bar{x}_1 x_2$ и $x_1 \bar{x}_2$ соответствуют 1-кубам ($s = 3 - 2 = 1$), а минитерм x_3 отображается 2-кубом ($s = 3 - 1 = 2$).

Итак, любая дизъюнктивная нормальная форма отображается на n-мерном кубе совокупностью s-кубов, которые покрывают все вершины, соответствующие конституентам единицы (0-кубы).

Справедливо и обратное утверждение: если некоторая совокупность s-кубов покрывает множество всех вершин, соответствующих единичным значениям функции, то дизъюнкция соответствующих этим s-кубам минитермов является выражением данной функции в дизъюнктивной нормальной форме. Говорят, что такая совокупность s-кубов (или соответствующих им минитермов) образует *покрытие функции.*

Стремление к минимальной форме интуитивно понимается как поиск такого покрытия, число s-кубов которого было бы поменьше, а их размерность s — побольше. Покрытие, соответствующее минимальной форме, называют *минимальным покрытием.* Например, для функции из (7.5) покрытие на рис. 5, *а* соответствует неминимальной форме $y = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_1 x_3 \vee x_2 x_3$, а покрытия на рис. 5, *б* и *в* — минимальным формам $y = \bar{x}_1 x_2 \vee x_1 x_2 \vee x_2 x_3$ и $y = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_1 x_3$.

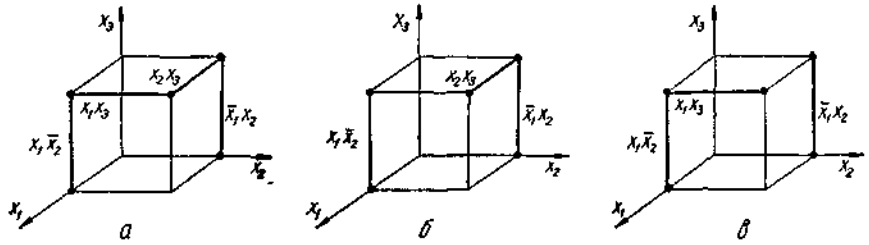


Рис. 5. Покрытия функции $y = \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3$;
а — неминимальное; б, в — минимальные

Отображение функции на n-мерном кубе наглядно и просто при $n \leq 3$. Четырехмерный куб можно изобразить, как показано на рис. 6, где отображены функция четырех переменных и ее минимальное покрытие, соответствующие выражению $y = x_1 \bar{x}_3 \vee x_2 x_4 \vee \bar{x}_1 x_3 \bar{x}_4$.

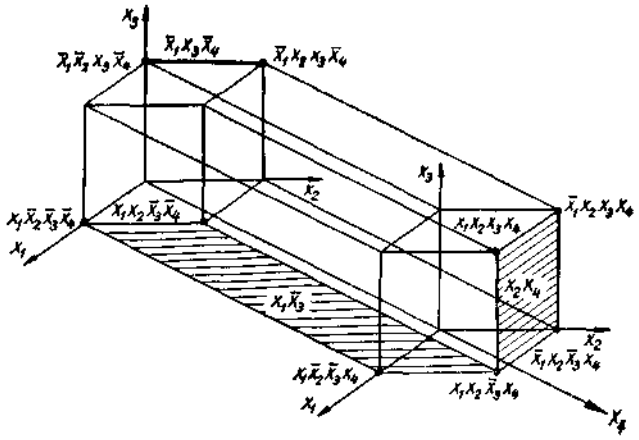


Рис.6. Отображение функции $y = x_1 \bar{x}_3 \vee x_2 x_4 \vee \bar{x}_1 x_3 \bar{x}_4$ на четырехмерном кубе.

Использование этого метода при $n > 4$ требует настолько сложных построений, что теряются все его преимущества.

7.7. Карты Карно.

В другом методе графического отображения булевых функций используются *карты Карно,* которые представляют собой специально организованные таблицы соответствия. Столбцы и строки таблицы соответствуют всевозможным наборам значений не более двух переменных, причем эти наборы расположены в таком порядке, что каждый последующий отличается от предыдущего значением только одной из переменных. Благоларя этому и соседние клетки таблицы по горизонтали и вертикали отличаются значением только одной переменной. Клетки, расположенные по краям таблицы, также считаются соседними и обладают этим свойством. На рис. 7 показаны карты Карно для двух, трех н четырех переменных.

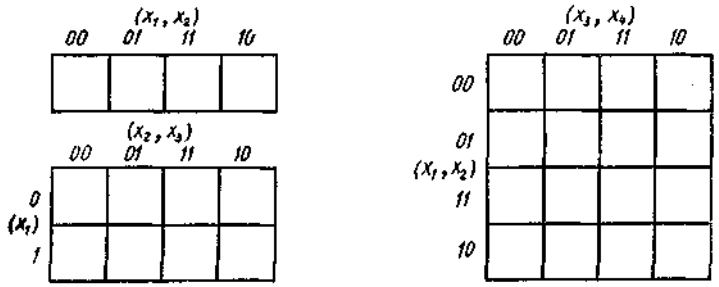


Рис. 7. Карты Карно для двух, трех и четырех переменных

Как и в обычных таблицах соответствия, клетки наборов, на которых функция принимает значение 1, заполняются единицами (нули обычно не вписывают, им соответствуют пустые клетки).
Например, на рис.8, *а* показана карта Карно для функции, отображение которой на четырехмерном кубе дано на рис. 6.

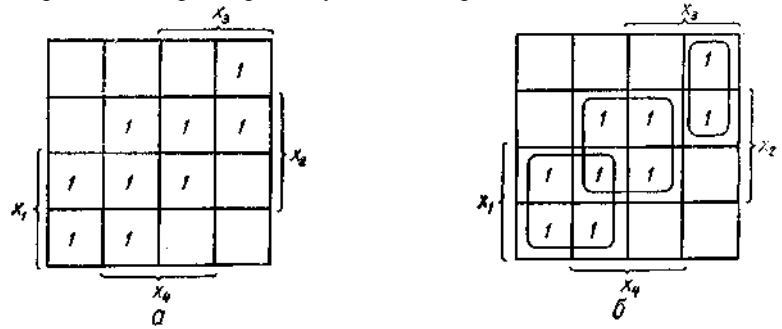


Рис.8. Отображение на карте Карно функции четырех переменных (а) и ее минимального покрытия (б).

Для упрощения строки и столбцы, соответствующие значениям 1 для некоторой переменной, выделяются фигурной скобкой с обозначением этой переменной.
Между отображениями функции на *n*-мерном кубе и на карте Карно имеет место взаимно-однозначное соответствие. На карте Карно s-кубу соответствует совокупность 2 соседних клеток, размещенных в строке, столбце, квадрате или прямоугольнике (с учетом соседства противоположных краев карты). Поэтому все положения, изложенные в (7.6), справедливы и для карт Карно. Так, на рис. 8, *б* показано покрытие единиц карты, соответствующее минимальной

дизъюнктивной форме $y = x_1\bar{x}_3 \vee x_2x_4 \vee \bar{x}_1x_3\bar{x}_4$ рассматриваемой функции.
Считывание минитермов с карты Карно осуществляется по простому правилу. Клетки, образующие s-куб, дают минитерм *(n* — s)-го ранга, в который входят те *(n* — *s)* переменные, которые сохраняют одинаковые значения на этом s-кубе, причем значениям 1 соответствуют сами переменные, а значениям 0 — их отрицания. Переменные, которые не сохраняют свои значения на s-кубе, в минитерме отсутствуют. Различные способы считывания приводят к различным представлениям функции в дизъюнктивной нормальной форме (рис.9).

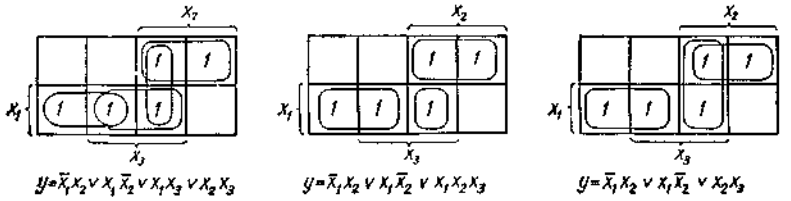


Рис.9. Способы считывания с карты Карно дизъюнктивной нормальной формы булевой функции

Использование карт Карно требует более простых построений по сравнению с отображением на *n*-мерном кубе, особенно в случае четырех переменных. Для отображения функций пяти переменных используются две карты Карно на четыре переменные, а для функций шести переменных — четыре таких карты. При дальнейшем увеличении числа переменных карты Карно становятся практически непригодными.
Известные в литературе *карты Вейча* отличаются только другим порядком следования наборов значений переменных и обладают теми же свойствами, что и карты Карно.

7.8. Комплекс кубов.

Несостоятельность графических методов при большом числе переменных компенсируется различными аналитическими методами представления булевых функций. Одним из таких представлений является *комплекс кубов,* использующий терминологию многомерного логического пространства в сочетании со специально разработанной символикой.
Комплекс кубов *K (y)* функции $y = f(x_1, x_2, \ldots, x_n)$ определяется как объединение множеств $K^s(y)$ всех ее s-кубов (s =0, 1, ..., *n),*

т. е. $K(y) = \bigcup K^s(y)$, причем некоторые из $K^s(y)$ могут быть пустыми. Для записи s-кубов и минитермов функции от n переменных используются слова длины n, буквы которых соответствуют всем n переменным. Входящие в минитерм переменные называются *связанными* и представляются значениями, при которых минитерм равен единице (1 для x_i и 0 для $\bar{x}_i$). Не входящие в минитерм переменные являются *свободными* и обозначаются через x. Например, 2-куб функции пяти переменных, соответствующий минитерму $x_2\bar{x}_3x_5$, запишется как (xl0x1). 0-кубы, соответствующие конституентам единицы, представляются наборами значений переменных, на которых функция равна единице. Очевидно, в записи s-куба всегда имеется s свободных переменных. Если все n переменных свободны, что соответствует n-кубу, то это означает тождественность единице рассматриваемой функции. Таким образом, для функций, не равных тождественно единице, $K^n(y) = \varnothing$.

Множество всех s-кубов $K(y)$ записывается как совокупность слов, соответствующих каждому s-кубу. Для удобства будем располагать слова s-кубов в столбцы, а их совокупность заключать в фигурные скобки. Например, комплекс кубов, соответствующий представлению функции на трехмерном кубе (рис.10, а), выражается как $K(y) = K^0 \cup K^1 \cup K^2$, где

$$K^0 = \begin{Bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{Bmatrix}; \quad K^1 = \begin{Bmatrix} 0 & 0 & 0 & 0 & x & x \\ 0 & x & x & 1 & 0 & 1 \\ x & 0 & 1 & x & 1 & 0 \end{Bmatrix}; \quad K^2 = \begin{Bmatrix} 0 \\ x \\ x \end{Bmatrix}.$$

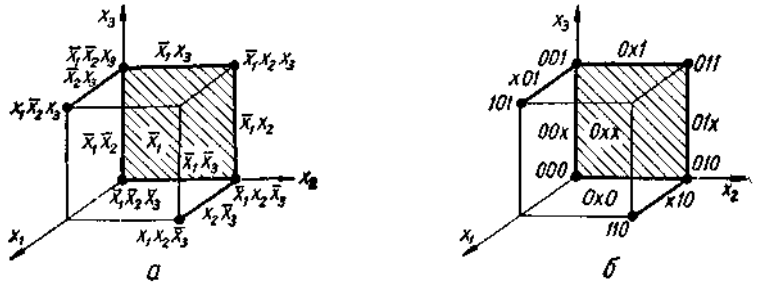


Рис.10. Комплекс кубов функции трех переменных (а) и его символическое представление *(б)*.

Для сравнения на рис.10, *б* изображен комплекс кубов в принятых обозначениях.

Комплекс кубов образует *максимальное покрытие функции.* Исключая из него все те s-кубы, которые покрываются кубами высшей размерности, получаем покрытия, соответствующие тупиковым формам. Так, для рассматриваемого примера (рис. 219) имеем тупиковое покрытие

$$C = \begin{Bmatrix} x & x & 0 \\ 0 & 1 & x \\ 1 & 0 & x \end{Bmatrix},$$

которое соответствует функции $y = \bar{x}_2x_3 \vee x_2\bar{x}_3 \vee \bar{x}_1$. В данном случае это покрытие является и минимальным.

Для двух булевых функций операция дизъюнкции соответствует объединению их комплексов кубов $K(y_1 \vee y_2) = K(y_1) \cup K(y_2)$, а операция конъюнкции — пересечению комплексов кубов $K(y_1y_2) = K(y_1) \cap K(y_2)$. Отрицанию функции соответствует дополнение комплекса кубов, т. е. $K(\bar{y}) = \overline{K(y)}$, причем $\overline{K(y)}$ определяется всеми вершинами, на которых функция принимает значение 0. Таким образом, имеет место взаимно-однозначное соответствие (изоморфизм) между алгеброй булевых функций и алгеброй множеств, представляющих комплексы кубов.

Представление функций и виде комплексов кубов менее наглядно, однако его важнейшие достоинства состоят и гом, что снимаются ограничения но числу переменных и облегчается кодирование информации при использовании вычислительных машин.

7.9. Реализация функций в различных формах.

Реализация функции в дизъюнктивной нормальной форме представляет собой логическую схему *И—ИЛИ.* Напрмер, функция $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_2x_3$ реализуется логической схемой (рис.11, *а).*

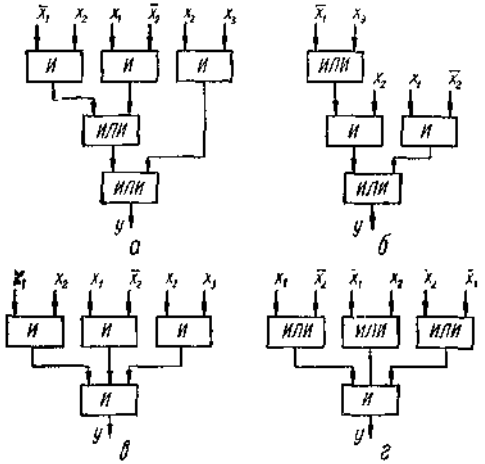


Рис. 11. Реализация функции $y = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_2 x_3$:
а—схемой *И-ИЛИ;* б —упрощенной схемой; в — двухуровневой схемой *И—ИЛИ; г —* двухуровневой схемой *ИЛИ-И*

Более экономичная реализация получается, если общий множитель вынести за скобки: $y = x_2(\bar{x}_1 \vee x_3) \vee x_1\bar{x}_2$ (рис.11, б). При использовании элементов со многими входами получаем двухуровневую логическую схему *И—ИЛИ* (рис. 11, *в).*
В соответствии с принципом двойственности , заменяя в дизъюнктивной нормальной форме операции конъюнкции на дизъюнкции, операции дизъюнкции на конъюнкции и беря отрицание каждой переменной, получаем конъюнктивную нормальною форму, которая выражает функцию $\bar{y}$, обратную к *y.* Ее реализация с помощью многовходовых элементов представляет собой двухуровневую логическую схему *ИЛИ—И.* Для рассматриваемой функции $\bar{y} = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_3)$ соответствующая реализация показана на рис.11, *г.* Если требуется получить схему для данной функции *y,* то используется инвертор или элемент, реализующий операцию *НЕ—И.*
Конъюнктивную нормальную форму можно получить и другим путем. Для этого используются рассуждения и методы, дуальные рассмотренным по отношению к дизъюнктивным нормальным формам. На многомерном кубе ищется покрытие множества вершин для нулевых значений функции, а на карте Карно — покрытие нулевых клеток. Рассматриваемый пример иллюстрируется па рис. 12, *а и б.*

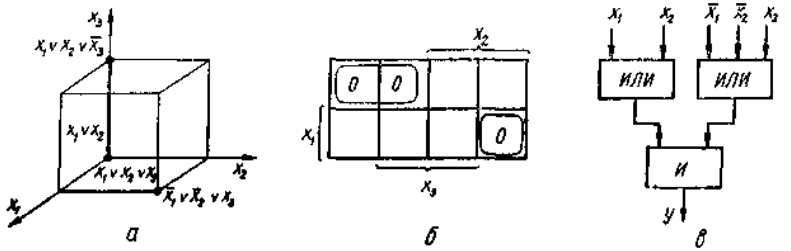


Рис. 12. Считывание конъюнктивной нормчльной формы булевой функции с куба *(а),* с карты Карно *(б)* и ее реализация логической схемой *(в).*

Соответствующая конъюнктивная нормальная форма $y = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ реализуется схемой (рис. 12, *в).*
Комплекс кубов этой функции и его дополнение имеют вид:

$$K(y) = \left\{\begin{array}{ccccc|cccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & x \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & x & 1 \\ 0 & 1 & 0 & 1 & 1 & x & x & 1 & 1 \end{array}\right\}; \quad \overline{K(y)} = \left\{\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & x \end{array}\right\},$$

а их покрытия

$$C = \left\{\begin{array}{ccc} 0 & 1 & x \\ 1 & 0 & 1 \\ x & x & 1 \end{array}\right\}; \quad \overline{C} = \left\{\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 0 & x \end{array}\right\}.$$

Покрытию $\overline{C}$ соответствует дизъюнктивная нормальная форма для отрицания функции $\bar{y} = x_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2$, откуда можно получить приведенное выше выражение функции в конъюнктивной нормальной форме.

7.10. Многовыходные схемы.

Схемы, реализующие несколько функций, можно представить как простое объединение схем, реализующих каждую функцию отдельно. Но такой путь, как правило, является неэкономичным. Часто бывает целесообразно преобразовать совокупность данных функций к такому виду, чтобы реализующие их схемы содержали общие части, а схема с многими выходами представляла собой единое целое.
Задача сводится к выбору для каждой функции такого покрытия, которое включало бы возможно большее число s-кубов, содержащихся в покрытиях других функций. Этому требованию удовлетворяют, например, покрытия для трех функций (рис13).

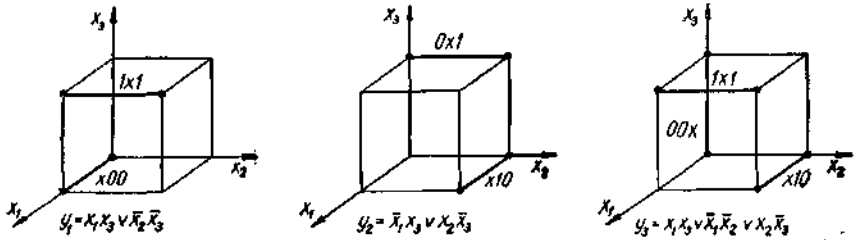


Рис. 13. Покрытия для трех выходных функций.

Соответствующая трехвыходная схема показана на рис.14.

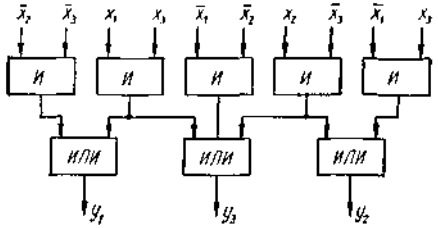


Рис. 14. Логическая схема с тремя выходами

Если бы для функции y_3 было выбрано другое покрытие, то схема получилась бы менее экономичной.

В этом параграфе описаны различные методы представления булевых функций применительно к задаче минимизации. При небольшом числе переменных эта задача обозрима, и ее можно решить простым перебором различных вариантов. Для функции многих переменных разработаны формальные методы минимизации, которые рассматриваются в следующем параграфе.

7.11. Постановка задачи минимизации булевых функций.

Минимизация схемы и булевом базисе сводится к поиску минимальной дизъюнктивной формы, которой соответствует минимальное покрытие. Общее число букв, входящих в нормальную форму, выражается *ценой покрытия* $c = \sum_s q_s (n - s)$, где q_s — число s-кубов, образующих покрытие данной функции от n переменных. Используются и другие определения цены покрытия, например $c' = c + q$, где q — общее число

всех кубов покрытия. Во всех случаях минимальное покрытие характеризуется наименьшим значением его цены.

Обычно задача минимизации решается в два шага. Сначала ищут сокращенное покрытие, которое включает все s-кубы максимальной размерности, но не содержит ни одного куба, покрывающегося каким-либо кубом этого покрытия. Соответствующую дизъюнктивную нормальную форму называют *сокращенной,* а ее минитермы— *простыми импликантами.* Для данной функции сокращенное покрытие является единственным, но оно может быть избыточным вследствие того, что некоторые из кубов покрываются совокупностями других кубов.

На втором шаге осуществляется переход от сокращенной к тупиковым дизъюнктивным нормальным формам, из которых выбираются минимальные формы. Тупиковые формы образуются путем исключения из сокращенного покрытия всех избыточных кубов, без которых оставшаяся совокупность кубов еще образует покрытие данной функции, но при дальнейшем исключении любого из кубов она уже не покрывает множества всех вершин, соответствующих единичным значениям функции, т. е. перестает быть покрытием.

Куб сокращенного покрытия, который покрывает вершины данной функции, не покрываемые никакими другими кубами, не может оказаться избыточным и всегда войдет в минимальное покрытие. Такой куб, как и соответствующая ему импликанта, называют *экстремалью (существенной импликантой)*, а покрываемые им вершины—*отмененными вершинам и.* Множество экстремалей образует *ядро покрытия.* Ясно, что при переходе от сокращенного покрытия к минимальному прежде всего следует выделить все экстремали. Если множество экстремалей не образует покрытия, то оно дополняется до покрытия кубами из сокращенного покрытия.

Приведенные определения иллюстрируются на рис. 5, где сокращенное покрытие Z (см. рис. 5, а) и минимальные покрытия $C'_{\min}$ (рис. 5, *в)* и $C''_{\min}$ (см. рис. 5, *в)* выражаются следующим образом:

$$Z = \begin{Vmatrix} 1 & 1 & x & 0 \\ 0 & x & 1 & 1 \\ x & 1 & 1 & x \end{Vmatrix}; \quad C'_{\min} = \begin{Vmatrix} 1 & x & 0 \\ 0 & 1 & 1 \\ x & 1 & x \end{Vmatrix}; \quad C''_{\min} = \begin{Vmatrix} 1 & 1 & 0 \\ 0 & x & 1 \\ x & 1 & x \end{Vmatrix}.$$

Сокращенная форма представляет собой дизъюнкцию четырех простых импликант, т. е. $y = x_1\bar{x}_2 \vee x_1x_3 \vee x_2x_3 \vee \bar{x}_1x_2$. Экстремалями являются простые импликанты $x_1\bar{x}_2$ и $\bar{x}_1x_2$, которым соответствуют 1-кубы *(10x)* и (*01x*), а отмеченные вершины — $x_1\bar{x}_2\bar{x}_3$ и $\bar{x}_1x_2\bar{x}_3$ или соответственно (100) и (010).

7.12. Метод Квайна — Мак-Класки.

Этот метод используется в случаях, когда функция задана в дизъюнктивной совершенной нормальной форме (или таблицей соответствия). Приведение к сокращенной форме осуществляется последовательным применением операции склеивания $ax_i \vee a\bar{x}_i = a$, где a — конъюнкции переменных, отличных от x_i. Множеству конституент единицы, представленных в исходной форме, соответствует совокупность 0-кубов K^0, а операции склеивания — объединение двух 0-кубов, которые отличаются только одной координатой. Результатом такого объединения является 1-куб, в котором различные координаты исходных 0-кубов замещены символом x. Сравнивая попарно все 0-кубы, получаем множество 1-кубов K^1. Применяя к K^1 операцию склеивания, находим множество 2-кубов K^2 и т. д. Этот процесс продолжается до тех пор, пока получаемое из K^s очередное K^{s+1} не окажется пустым множеством. В результате множество K^0преобразуется в комплекс кубов $K = \{K^0, K^1, K^2, \ldots, K^s\}$, причем $s \leq n$.

Для выделения из K множества простых импликант $Z \subset K$ при каждой операции склеивания необходимо отмечать каким-либо знаком (напримгр, меткой $\vee$) те кубы, которые объединяются в кубы высшей размерности. Очевидно, неотмеченные кубы и образуют множество простых импликант Z. Чтобы уменьшить число сравниваемых пар при операции объединения целесообразно разбить множество K^s на классы, в каждом из которых содержатся s-кубы с одинаковым числом единиц (или нулей), и упорядочить эти классы по возрастающему числу единиц. Так как объединяться могут только такие два s-куба, число единиц в которых точно на одну больше или меньше, то достаточно ограничиться попарным сравнением s-кубов одного класса с s-кубами соседнего класса.

На втором шаге при извлечении экстремалей и образовании минимального покрытия используется таблица покрытий. Ее строки соответствуют простым импликантам, а столбцы — конституентам единицы дизъюнктивной совершенной нормальной формы данной функции, которые представляются 0-кубами (вершинами) комплекса кубов. В клетку таблицы записывается метка, если простая импликанта в данной строке покрывает вершину в данном столбце. Экстремалям соответствуют те строки таблицы, которые содержат единственную метку в каком-либо столбце. Удаляя строки экстремалей и все столбцы, в которых эти строки имеют метки, получаем более

простую таблицу. На основе этой таблицы выбираем простые импликанты, которые дополняют выделенное множество экстремалей до минимального покрытия функции.

7.13. Пример минимизации функции.

Рассмотрим в качестве примера функцию четырех переменных $y = f(x_1, x_2, x_3, x_4)$, заданную таблицей соответствия

x_1	0000	0000	1111	1111
x_2	0000	1111	0000	1111
x_3	0011	0011	0011	0011
x_4	0101	0101	0101	0101
y	0001	1101	0101	1100

Ей соответствует дизъюнктивная совершенная нормальная форма $y = \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3x_4 \vee x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4$. Множество 0-кубов после разбиения и упорядочения записывается следующим образом:

$$K^0 = \left\{\begin{array}{c|cc|ccc} \check{0} & \check{0} & \check{0} & \check{1} & \check{1} & \check{0} & \check{1} & \check{1} \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right\}.$$

Объединяя кубы и отмечая те из них, которые покрываются кубами большей размерности, имеем:

$$K^1 = \left\{\begin{array}{cc|ccccccc} \check{0} & \check{x} & 0 & 0 & x & 1 & \check{x} & 1 & \check{1} \\ 1 & 1 & x & 1 & 0 & 0 & 1 & x & 1 \\ 0 & 0 & 1 & x & 1 & x & 0 & 0 & 0 \\ x & 0 & 1 & 1 & 1 & 1 & 1 & 1 & x \end{array}\right\}; \quad K^2 = \left\{\begin{array}{c} x \\ 1 \\ 0 \\ x \end{array}\right\}.$$

Простым импликантам соответствуют неотмеченные кубы. Составляем таблицу покрытия Z, которому соответствует сокращенная форма $y = \bar{x}_1x_3x_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_2x_3x_4 \vee x_1\bar{x}_2x_4 \vee x_1\bar{x}_3x_4 \vee x_2x_3$:

K^* / Z	0100	0011	0101	1001	1100	0111	1011	1101	Обозначения импликант
0 x 1 1		∨				∨			A
0 1 x 1			∨			∨			B
x 0 1 1		∨					∨		C
1 0 x 1				∨			∨		D
1 x 0 1				∨				∨	E
x 1 0 x	∨		∨		∨			∨	F

Извлекаем единственную экстремаль *(x10x),* которой соответствует минитерм $x_2\bar{x}_3$, и упрощаем таблицу к виду:

K_1^* / Z_1	0011	1001	0111	1011
0 x 1 1	∨		∨	
0 1 x 1			∨	
x 0 1 1	∨			∨
1 0 x 1		∨		∨
1 x 0 1		∨		

В качестве дополнительных целесообразно выбрать кубы (0*x*11) и (10*x*1), так как они совместно с экстремалью *(x10x)* образуют покрытие функции, минимальная форма которой имеет вид: $y = \bar{x}_1x_3x_4 \vee x_1\bar{x}_2x_4 \vee x_2\bar{x}_3$. Соответствующее этой функции минимальше покрытие иллюстрируется на четырехмерном кубе и на карте Карно (рис. 15).

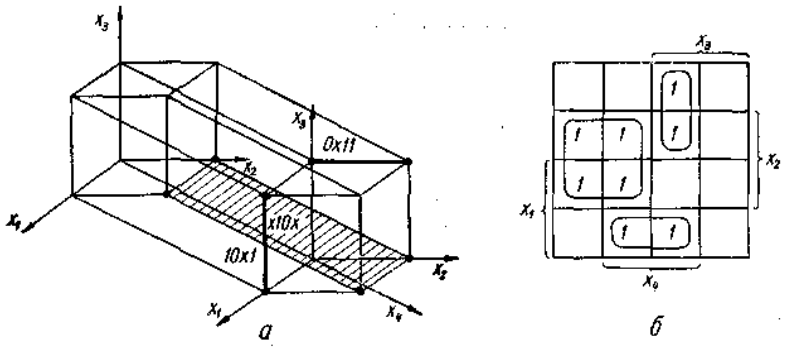


Рис.15. Минимальное покрытие функции на четырехмерном кубе (а) и карте Карно *(б).*

7.14. Алгебраический метод.

Выбор минимального покрытия на заключительном этапе формализуется с помощью *алгебраического метода,* предложенного С. Петриком. Простые импликанты обозначаются какими-либо символами (обычно для этой цели используются прописные буквы латинского алфавита), и по столбцам таблицы покрытий записываются дизъюнкции тех импликант, которые отмечены в данном столбце. Смысл этой записи вытекает из того, что любая из отмеченных импликант покрывает данную вершину. Покрытию функции соответствует конъюнкция всех записанных дизъюнкций. Раскрывая скобки и упрощая выражения на основе тождеств булевой алгебры (упрощать можно и до раскрытия скобок), переходим к дизъюнктивной форме, каждый член которой представляет собой конъюнкцию простых импликант и соответствует некоторому тупиковому покрытию рассматриваемой функции. Сравнивая все тупиковые покрытия и отбирая те из них, которые характеризуются минимальной ценой, приходим к одному или нескольким минимальным покрытиям.

Так,

$$\wedge(C \vee D)(E \vee F) = F(A \vee C)(A \vee B)(D \vee E)(C \vee D) = F(A \vee AB \vee \vee AC \vee BC)(CD \vee CE \vee D \vee DE) = F(A \vee BC)(D \vee CE) = ADF \vee$$

для примера из (7.13) имеем: $F(A \vee C)(B \vee F)(D \vee E)F(A \vee B) \wedge \vee ACEF \vee BCDF \vee BCEF$. Итак, получаем четыре тупиковых покрытия

$$C_1 = \begin{Vmatrix} 0 & 1 & x \\ x & 0 & 1 \\ 1 & x & 0 \\ 1 & 1 & x \end{Vmatrix}; \quad C_2 = \begin{Vmatrix} 0 & x & 1 & x \\ x & 0 & x & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & x \end{Vmatrix},$$

$$C_3 = \begin{Vmatrix} 0 & x & 1 & x \\ 1 & 0 & 0 & 1 \\ x & 1 & x & 0 \\ 1 & 1 & 1 & x \end{Vmatrix}; \quad C_4 = \begin{Vmatrix} 0 & x & 1 & x \\ 1 & 0 & x & 1 \\ x & 1 & 0 & 0 \\ 1 & 1 & 1 & x \end{Vmatrix},$$

цены которых
$c_1 = 2(4-1) + 1(4-2) = 8$"и$c_2 = c_3 = c_4 = 3(4-1) + 1(4-2) = 11$, т.е. $C_{\min} = C_1$.

Алгебраические преобразования упрощаются, если исходить из таблицы покрытий, получаемой после извлечения экстремалей. Тогда результатом таких преобразований являются множества простых импликант, дополняющих совокупность экстремалей до тупиковых покрытий. Сравнивая эти множества по их цене, выбираем минимальные дополнения, которые совместно с множеством экстремалей образуют минимальные покрытия.

7.15. Метод Блейка—Порецкого.

При минимизации функции мтодом Квайна—Мак-Класки требуется предварительно представить ее в совершенной дизъюнктивной нормальной форме, что часто связано с дополнительными преобразованиями.

Если исходить из произвольной дизъюнктивной нормальней формы, то для получения промежуточной сокращенной формы можно воспользоваться прямым *методом Блейка—Порецкого.* Он основан на тождестве

$$ac \vee b\bar{c} = ac \vee b\bar{c} \vee ab,$$

называемом *операцией обобщенного склеивания.* Действительно, $ac \vee b\bar{c} = ac \vee abc \vee b\bar{c} \vee ab\bar{c} = ac \vee b\bar{c} \vee ab(c \vee \bar{c}) = ac \vee b\bar{c} \vee ab$. Разумеется, входящие в это тождество буквы могут представлять любые булевы формулы и, в частности, конъюнкции переменных.

Можно показать, что произвольная дизъюнктивная нормальная форма приводится к сокращенной применением всех возможных обобщенных склеиваний с последующим устранением мипитермов на основе операции поглощения $a \vee ab = a$. При этом возможны

следующие случаи.

1) Конъюнкция a содержит переменную x_j, а конъюнкция b — отрицание той же переменной $\bar{x}_j$ (или наоборот). Тогда $ab = 0$ и в результате операции обобщенного склеивания не получаются новые минитермы. Таким образом, следует подвергать этой операции только те пары минитермов, в которых единственная переменная представлена как x_j и $\bar{x}_j$.

2) Конъюнкция a содержит только те переменные, которые входят в конъюнкцию b (или наоборот), т. е. $b = ac$. Тогда $ax_i \vee b\bar{x}_i = ax_i \vee ac\bar{x}_i = ax_i \vee ac\bar{x}_i \vee ac = ax_i \vee ac = ax_i \vee b$, т. е. минитерм исходной дизъюнктивной нормальной формы поглощается минитермом, образованным в результате обобщенного склеивания. Пусть, например, функция из (7.13) задана некоторым покрытием, которое соответствует дизъюнктивной нормальной форме: $y = x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_1\bar{x}_2x_3x_4$. Применяя операцию обобщенного склеивания к парам $(x_2\bar{x}_3\bar{x}_4,\ \bar{x}_1x_2x_4)$; $(x_1x_2\bar{x}_3,\ x_1\bar{x}_2x_4)$; $(x_1x_2\bar{x}_3,\ \bar{x}_1x_2x_4)$; $(x_1\bar{x}_2x_4,\ \bar{x}_1\bar{x}_2x_3x_4)$; $(\bar{x}_1x_2x_4,\ \bar{x}_1\bar{x}_2x_3x_4)$ и учитывая, что в двух последних парах происходит поглощение минитермов, получаем:

$$y = (x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_1x_2\bar{x}_3) \vee (x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4 \vee (\bar{x}_1x_2x_4 \vee \bar{x}_1x_3x_4).$$

Удаляя

$$\vee\ x_1\bar{x}_3x_4) \vee (x_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_4 \vee x_2\bar{x}_3x_4) \vee (x_1\bar{x}_2x_4 \vee \bar{x}_2x_3x_4) \vee$$

одинаковые члены $(a \vee a = a)$ и группируя старые и новые мипитермы, имеем: $y = (x_2\bar{x}_3\bar{x}_4 \vee \vee \bar{x}_1x_2x_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4) \vee (\bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_3x_4 \vee x_2\bar{x}_3x_4 \vee \bar{x}_2x_3x_4 \vee \vee \bar{x}_1x_3x_4)$. Очевидно, при дальнейшем обобщенном склеивании имеет смысл рассматривать только пары, образованные новыми минитермами со всеми минитермами полученной дизъюнктивной нормальной формы. Такими парами являются: $(x_2\bar{x}_3\bar{x}_4,\ x_1\bar{x}_3x_4)$; $(x_2\bar{x}_3\bar{x}_4,\ x_2\bar{x}_3x_4)$; $(\bar{x}_1x_2x_4,\ x_1\bar{x}_3x_4)$; $(\bar{x}_1x_2x_4,\ \bar{x}_2x_3x_4)$; $(x_1x_2\bar{x}_3,\ \bar{x}_1x_2\bar{x}_3)$; $(x_1\bar{x}_2x_4,\ x_2\bar{x}_3x_4)$; $(x_1\bar{x}_2x_4,\ \bar{x}_1x_3x_4)$; $(\bar{x}_1x_3\bar{x}_3,\ x_1\bar{x}_3x_4)$; $(\bar{x}_1x_2\bar{x}_3,\ \bar{x}_1x_3x_4)$; $(x_1\bar{x}_3x_4,\ \bar{x}_2x_3x_4)$; $(x_2\bar{x}_3x_4,\ \bar{x}_1x_3x_4)$. Применяя к каждой паре операции обобщенного склеивания и поглощения в соответствии с приведенными выше правилами, находим: $y = \bar{x}_1x_2x_4 \vee \vee x_1\bar{x}_2x_4 \vee x_1\bar{x}_3x_4 \vee \bar{x}_2x_3x_4 \vee \bar{x}_1x_3x_4 \vee x_2\bar{x}_3$. Единственный новый минитерм $x_2\bar{x}_3$ в паре с любым из остальных минитермов не приводит

к появлению новых минитермов. Поэтому полученная форма является сокращенной. Она, как и должно быть, совпадает с найденной в (7.13).

7.16. Склеивание и поглощение кубов.

Геометрически операции ообобщенного склеивания и поглощения соответствуют некоторым операциям над кубами, имеющими противоположные грани. В результате получается новый куб, который либо располагается между исходными кубами, либо поглощает один из кубов или оба куба. Преобразования, выполненные в (7.15) иллюстрируются на рис.16.

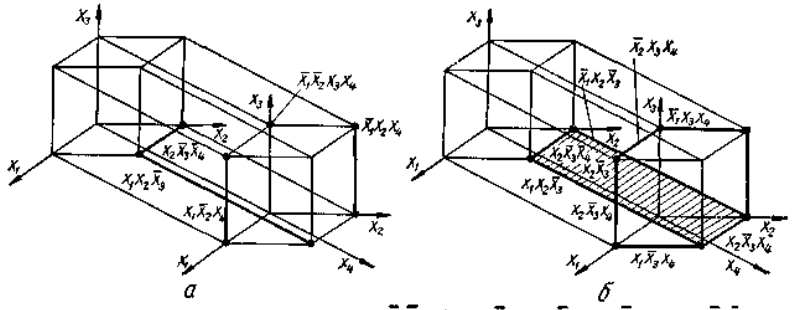


Рис.16. Покрытие функции $y = x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_3x_4 \vee \bar{x}_1x_2x_1 \vee \bar{x}_1\bar{x}_2x_3x_4$; *а* — исходное; *б* — промежуточное.

Исходной дизъюнктивной нормальной форме соответствует некоторое покрытие (рис.16, *а),* которое преобразуется к промежуточному покрытию (рис.16, *б).* Сокращенной нормальной форме соответствует покрытие, получаемое из рис.16, *б* поглощением кубов $x_2\bar{x}_3\bar{x}_4$, $\bar{x}_1x_2\bar{x}_3$, $x_2\bar{x}_3x_4$ и $x_1x_2\bar{x}_3$ кубом $x_2\bar{x}_3$.

Операции над кубами удобно выполнять в символической форме. Сравнивая в исходном покрытии C_0 попарно кубы, имеющие противоположные значения 0 и 1 только для одной координаты, образуем множество новых кубов C^*_0. Координаты этих кубов можно определить с помощью *операции покоординатного произведения (*),* задаваемой таблицей:

*	0	1	x
0	0	x	0
1	x	1	1
x	0	1	x

Так, для рассматриваемого примера имеем:

$$C_0 = \begin{Bmatrix} x & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & x & x & 1 \\ 0 & x & 1 & 1 & 1 \end{Bmatrix}; \quad C^*_0 = \begin{Bmatrix} 0 & 1 & x & x & 0 \\ 1 & x & 1 & 0 & x \\ 0 & 0 & 0 & 1 & 1 \\ x & 1 & 1 & 1 & 1 \end{Bmatrix},$$

где кубы множества C^*_0 получены в результате операции покоординатного произведения над следующими парами кубов из C_0:

$$\begin{Bmatrix} x & 0 \\ 1 & 1 \\ 0 & x \\ 0 & 1 \end{Bmatrix}; \quad \begin{Bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & x \\ x & 1 \end{Bmatrix}; \quad \begin{Bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & x \\ x & 1 \end{Bmatrix}; \quad \begin{Bmatrix} 1 & 0 \\ 0 & 0 \\ x & 1 \\ 1 & 1 \end{Bmatrix}; \quad \begin{Bmatrix} 0 & 0 \\ 1 & 0 \\ x & 1 \\ 1 & 1 \end{Bmatrix}.$$

Объединяя множества C_0 и C^*_0, выполняем операции поглощения в соответствии с тождествами $a \vee a = a$ и $a \vee ab = a$. Это соответствует удалению из множества $C_0 \cup C^*_0$ повторяющихся кубов, а также тех кубов, которые покрываются другими кубами (куб покрывает все кубы низшей размерности, если отличные от x координаты покрывающего куба совпадают с соответствующими координатами покрываемых кубов). В нашем примере повторяющихся кубов нет, а куб (0011) поглощается кубом (x011) или (0x11). В результате получаем промежуточное покрытие

$$C_1 = \left\{\begin{array}{cccc|ccccc} x & 1 & 1 & 0 & 0 & 1 & x & x & 0 \\ 1 & 1 & 0 & 1 & 1 & x & 1 & 0 & x \\ 0 & 0 & x & x & 0 & 0 & 0 & 1 & 1 \\ 0 & x & 1 & 1 & x & 1 & 1 & 1 & 1 \end{array}\right\},$$

где исходные и новые кубы разделены пунктирной линией. Дальше операция обобщенного склеивания выполняется над покрытием C_1 покоординатным произведением кубов, расположенных справа от разделяющей линии, с каждым кубом из C_1, который подлежит склеиванию. Получаем множество новых кубов

$$C^*_1 = \begin{Bmatrix} 1 & x & x & 0 & x & 1 & x & x & 0 & 1 & 0 \\ 1 & 1 & 1 & x & 1 & x & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & x & x & x \\ x & x & 1 & 1 & x & 1 & 1 & 1 & 1 & 1 & 1 \end{Bmatrix}.$$

После операции поглощения в множестве $C_1 \cup C^*_1$ имеем следующее преобразованное покрытие:

$$C_2 = \left\{\begin{array}{cc|ccc|c} 1 & 0 & 1 & x & 0 & x \\ 0 & 1 & x & 0 & x & 1 \\ x & x & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & x \end{array}\right\}.$$

Продолжая склеивание кубов последней группы (она содержит единственный 2-куб) со всеми кубами из C_2, получаем множество

$$C_2^* = \begin{Vmatrix} 1 & 0 \\ x & 1 \\ 0 & x \\ 1 & 1 \end{Vmatrix},$$

объединяя которое с C_2 операциями поглощения, приходим снова к C_2, так как C^*_2 не содержит новых кубов. Отсюда следует, что покрытие C_2 соответствует сокращенной дизъюнктивной нормальной форме данной функции.

Ниже приведена более рациональная запись преобразования произвольной дизъюнктивной нормальной формы к сокращенной форме:

$$\left\{\begin{array}{cccc|cccc|cccccccc|cc} \check{x} & \check{1} & 1 & 0 & \check{0} & 1 & \check{x} & x & \check{1} & x & \check{x} & \check{0} & \check{x} & \check{1} & \check{x} & \check{x} & \check{0} & \check{1} & \check{1} & \check{0} \\ 1 & 1 & 0 & 1 & 0 & 1 & x & 0 & 1 & 1 & 1 & x & 1 & x & 0 & 1 & 1 & 0 & x & 1 \\ 0 & 0 & x & x & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & x & x & x \\ 0 & x & 1 & 1 & 1 & x & 1 & 1 & 1 & x & x & 1 & 1 & x & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right\}.$$

C_0 C_0^* C_1^* C_2^*

На каждом этапе над поглощаемыми кубами ставятся метки V (или соответствующие столбцы вычеркиваются). По окончании преобразования сокращенное покрытие определяется совокупностью неотмеченных кубов.

7.17. Частично определенные функции.

В практике нередко приходится иметь дело с такими функциями, которые определены не на всех наборах значений переменных. Подобные случаи встречаются, когда по условиям функционирования некоторые из наборов не используются и поэтому безразлично, какие значения принимает функция на этих наборах. Это обстоятельство можно использовать при минимизации функции, доопределив ее на *безразличных наборах* так, чтобы обеспечить наиболее экономичную реализацию.

Пусть дана частично определенная функция $y = f(x_1, x_2, \ldots, x_n)$. Обозначим через $y^1 = f^1(x_1, x_2, \ldots, x_n)$ функцию, которая доопределена на всех безразличных наборах единицами, а через $y^0 = f^0(x_1, x_2, \ldots, x_n)$ — нулями. Задача оптимального доопределения данной функции сводится к выбору из сокращенного покрытия для функции y^1 минимального количества кубов максимальной размерности, совокупность которых покрывала бы все вершины функции y^0. Такая совокупность кубов и образует минимальное

покрытие частично определенной функции y. При этом оно может покрывать и некоторые вершины, соответствующие безразличным наборам, что означает доопределение функции на этих наборах единичными значениями.

7.18. Преобразователь кодов.

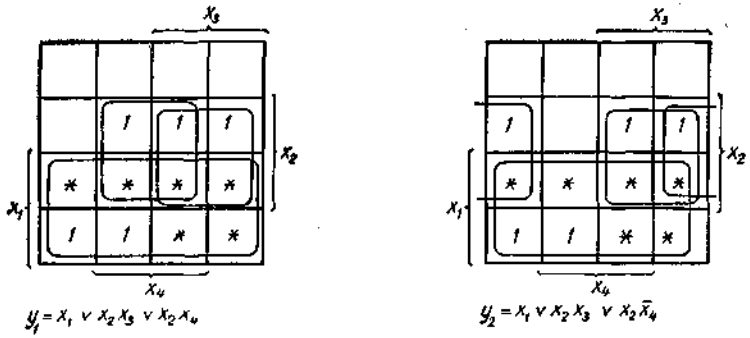
Примером частично определенных функций может служить таблица соответствия преобразования кода прямого замещения в двоично-десятичный код 2421:

Десятичное число		0 1 2 3 4 5 6 7 8 9	Избыточные наборы
Код прямого замещения	x_1	0 0 0 0 0 0 0 0 1 1	1 1 1 1 1 1
	x_2	0 0 0 0 1 1 1 1 0 0	0 0 1 1 1 1
	x_3	0 0 1 1 0 0 1 1 0 0	1 1 0 0 1 1
	x_4	0 1 0 1 0 1 0 1 0 1	0 1 0 1 0 1
Двоично-десятичный код 2421	y_1	0 0 0 0 0 1 1 1 1 1	Функции не определены
	y_2	0 0 0 0 1 0 1 1 1 1	
	y_3	0 0 1 1 0 1 0 0 1 1	
	y_4	0 1 0 1 0 1 0 1 0 1	

Код прямого замещения представляет собой обычное представление одноразрядного десятичного числа в двоичной системе счисления, т. е. $x_1 \cdot 2^3 + x_2 \cdot 2^2 + x_3 \cdot 2^1 + x_4 \cdot 2^0 = 8x_1 + 4x_2 + 2x_3 + x_4$. Код 2421 соответствует представлению числа в виде $y_1 \cdot 2^1 + y_2 \cdot 2^2 + y_3 \cdot 2^1 + y_4 \cdot 2^0 = 2y_1 + 4y_2 + 2y_3 + y_4$.

Таким образом, преобразователь кодов представляет собой схему с четырьмя входами и четырьмя выходами.

Проиллюстрируем минимизацию схемы на картах Карно (рис.17) с учетом положений о многовыходных схемах, изложенных ранее.



Ркс. 17. Минимальные покрытия выходных функций преобразователя кодов.

Используя избыточные наборы, которые отмечены на карте звездочками, образуем минимальные покрытия для каждой из четырех функций которые включали бы возможно больше однотипных кубов.

Соответствующая логическая схема показана на рис.18.

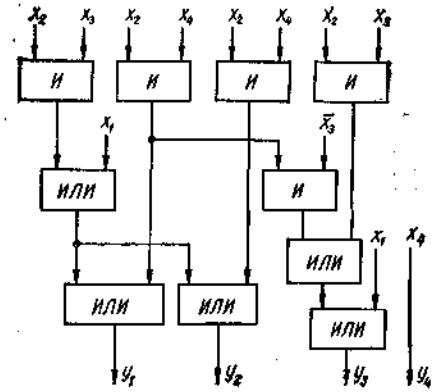


Рис.18. Логическая схема преобразователя кодов.

7.19. Сумматор.

Другим примером логической схемы, который дает повод использовать частично определенные функции, является одноразрядный сумматор, выполняющий арифметическое сложение двоичных чисел x_k и y_k k-го разряда и переноса из младшего разряда p_{k-1}. В результате должна получаться сумма s_k и перенос в старший разряд $p_к$. Таблица соответствия такого сумматора имеет вид:

x_k	0	0	0	0	1	1	1	1
y_k	0	0	1	1	0	0	1	1
p_{k-1}	0	1	0	1	0	1	0	1
s_k	0	1	1	0	1	0	0	1
p_k	0	0	0	1	0	1	1	1

Отображение функций s_k и $p_к$ на трехмерных кубах показана на рис.19.

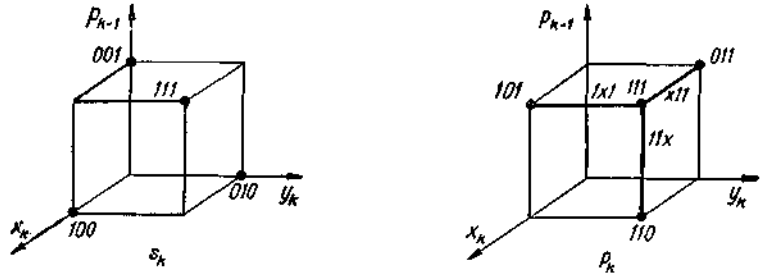


Рис. 19. Отображение выходных функций сумматора на трехмерных кубах.

Их дизъюнктивные нормальные формы имеют вид: $s_k = \bar{x}_k\bar{y}_k p_{k-1} \vee \bar{x}_k y_k \bar{p}_{k-1} \vee x_k\bar{y}_k\bar{p}_{k-1} \vee x_k y_k p_{k-1}$ и $p_k = x_k p_{k-1} \vee x_k y_k \vee y_k p_{k-1}$, соответствующие минимальным покрытиям. Как видно, выражение для s_k не поддается минимизации изложенными ранее методами. Единственная возможность — это использовать вынесение за скобки: $s_k = (x_k\bar{y}_k \vee \bar{x}_k y_k)\bar{p}_{k-1} \vee (x_k y_k \vee \bar{x}_k\bar{y}_k) p_{k-1}$. В подобных случаях для минимизации применяется прием, основанный на использовании более простой реализации функции $p_k = f(x_k, y_k, p_{k-1})$ в качестве составной части другой функции s_k. При этом p_k рассматривается как переменная, т. е. $s_k = \varphi(x_k, y_k, p_{k-1}, p_k)$. Но таблица соответствия для s_k теперь содержит избыточные наборы переменных, которые отмечены звездочками:

x_k	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
y_k	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
p_{k-1}	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
p_k	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
s_k	0	*	1	*	1	*	*	0	1	*	*	0	*	0	*	1

Используем для минимизации полученной частично определенной функции s_k карту Карно (рис. 20).

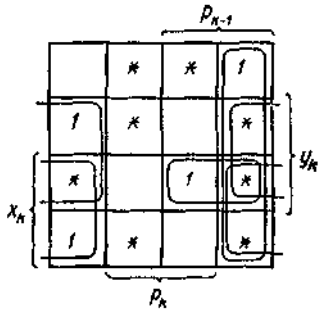


Рис.20. Минимизация функции s_k сумматора на карте Карно.

Минимальному покрытию соответствует выражение $s_k = x_k\bar{p}_k \vee y_k\bar{p}_k \vee p_{k-1}\bar{p}_k \vee x_k y_k p_{k-1}$. После вынесения за скобки получаем подготовленные к реализации выражения: $s_k = (x_k \vee y_k \vee p_{k-1})\bar{p}_k \vee x_k y_k p_{k-1}$; $p_k = x_k y_k \vee (x_k \vee y_k) p_{k-1}$. Соответствующая схема показана на рис.21.

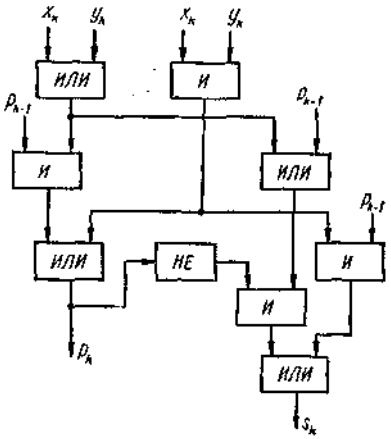


Рис. 21. Логическая схема сумматора.

7.20. Минимизация в других системах.

В реальных условиях проектирование логических схем основывается на использовании некоторого конкретного набора элементов. Обычно стремятся стандартизировать такие элементы с тем, чтобы при одинаковой конструкции они позволяли в зависимости от способа включения реализовать различные логические функции. Например, комплект интегральных схем может включать многовходовые транзисторные вентили *НЕ—ИЛИ* и *НЕ—И,* а также полусумматоры, реализующие сумму по модулю 2 (неравнозначность). С помощью схем на ферритах обычно реализуются отрицание, дизъюнкция, конъюнкция, штрих Шеффера и стрелка Пирса. Один из пневмисторных модулей, наряду с этими функциями, позволяет реализовать также импликацию и отрицание импликации. В связи с этим перед разработчиком возникает задача представления и минимизации функции в различных функционально полных системах элементов. Известны методы получения канонических форм для логических функций в любом базисе на основе табличного задания или преобразования другого базиса. Что же касается проблемы минимизации в общем виде, то она остается пока открытой. Обычно применяются частные методы минимизации, аналогичные разработанным для булевого базиса. Часто минимальное представление в булевом базисе используется как исходное и при реализации в

других базисах, соответствующее выражение функции в которых получается на основе тождественных преобразований.

8. Контактные схемы

Для математического описания электротехнических устройств, состоящих из контактов и промежуточных реле, функционирующих в дискретные моменты времени применяются *контактные схемы*. С помощью *контактных схем* можно представить любую булеву функцию.

Определение:

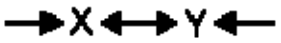
Контактная схема (англ. *contact circuit*) представляет собой ориентированный ациклический граф, на каждом ребре которого написана переменная или ее отрицание.

Определение:

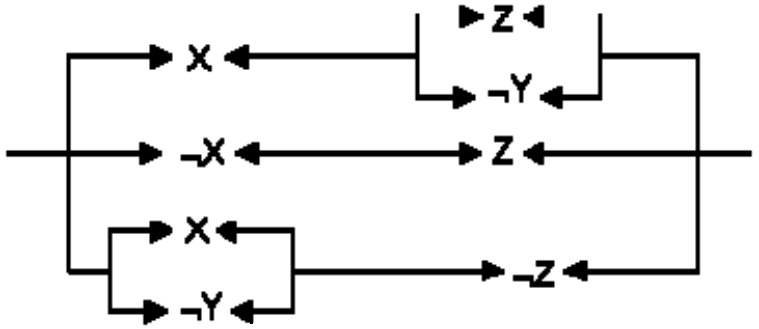
Контакт (англ. *contact*) — ребро схемы, помеченное символом переменной или ее отрицанием. Каждому ребру в схеме сопоставляется какая то переменная (не обязательно каждой переменной сопоставляется ребро)

Контактом будем называть устройство, которое в процессе работы может быть в двух состояниях: контакт может быть замкнут или разомкнут. Контакт X на чертеже будем изображать следующим образом:

Контакты можно соединять между собой различными способами:



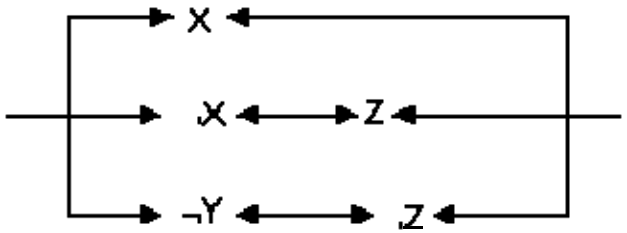
Первое соединение называется *параллельным*, второе – *последовательным*. Контакты, соединенные между собой, будем называть контактной схемой. Будем предполагать наличие у схемы двух выделенных точек входа и выхода. Схему назовем замкнутой, если существует последовательность замкнутых контактов $X_1,X_2,...,X_n$ такая, что X_i соединен с X_{i+1}, X_1 соединен с входом, X_n – с выходом. Схему, не являющуюся замкнутой, назовем разомкнутой. Каждому контакту поставим в соответствие высказывание, которое истинно тогда и только тогда, когда контакт замкнут. Высказывание и контакт будем обозначать одной буквой. Пусть схема S построена из контактов $X_1,X_2,...,X_n$ с помощью параллельного и последовательного соединений. Тогда по схеме S можно построить формулу логики высказываний F_S так, что параллельному соединению соответствует дизъюнкция, последовательному – конъюнкция. Например, схеме S_0 соответствует формула



F_{S0}=X&(ZÚØY)]Ú[ØX&Z]Ú[(XÚØY)&ØZ].

(Через ØV обозначается контакт, который замкнут тогда и только тогда, когда V разомкнут). Формула F_S "представляет схему в следующем смысле: схема S замкнута в том и только в том случае, если Fs принимает значение И. Контактным схемам соответствуют формулы, в построении которых участвуют лишь связки &,Ú,Ø, причем отрицание применяется только к атомарным формулам. Нетрудно понять, что по всякой такой формуле F можно восстановить схему, которую формула F «представляет».

Пусть схемам S и T соответствуют формулы F_S и F_T в описанном выше смысле. Тогда если схемы S и T эквивалентны (т.е. замкнуты и разомкнуты одновременно), то F_S и F_T равносильны, и обратно. Этот факт используется для решения задачи минимизации контактных схем, которая состоит в том, чтобы по данной схеме S найти схему T, эквивалентную S и содержащую меньше контактов. Один из путей решения этой задачи состоит в переходе к формуле F_S и в отыскании формулы G, равносильной F_S и содержащей меньше вхождений атомарных формул (разумеется, G построена только с помощью &, Ú и Ø, причем Ø применяется лишь к атомарным формулам). Так, например, формула F_S равносильна формуле XÚ(ØX&Z)Ú(ØY&ØZ). Следовательно, приведенная выше схема эквивалентна следующей схеме



которая состоит на три контакта меньше

8.1. Принцип работы

Определение:

Замкнутый контакт (англ. *closed contact*) — контакт схемы, над которым написана 0 или значение переменной равно 0.

Определение:

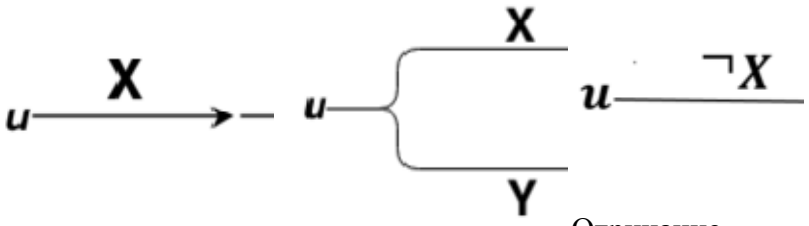
Разомкнутый контакт (англ. *open contact*) — контакт схемы, над которым написана 1 или значение переменной равно 1.

Пусть u и v — два полюса контактной схемы (из вершины u ребра только выходят, в вершину v ребра только входят), определяющую функцию $f(x_1, x_2 \dots, x_n)$. Тогда $f(x_1, x_2 \dots, x_n)$ принимает значение 1 при таком наборе значений переменных, если можно добраться из u в v только по разомкнутым контактам.

8.2. Построение контактных схем

Представление одного из базисов в контактных схемах

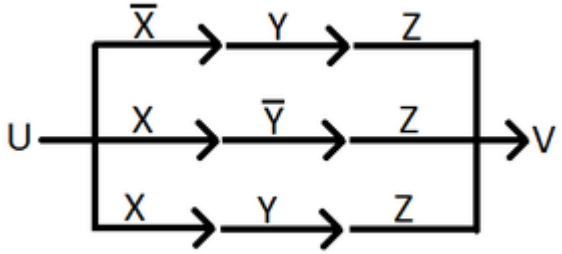
Любую булеву функцию можно представить в виде контактной схемы. Для этого необходимо привести её к ДНФ или КНФ, а затем построить, используя комбинации трех логических элементов:



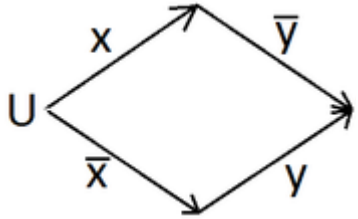
Конъюнкция Отрицание

Дизъюнкция

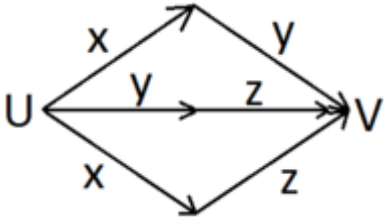
Пусть задана произвольная булева функция. Требуется построить для нее контактную схему, которая ее реализует. В качестве примера рассмотрим функцию, представленную в ДНФ:
$f = (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$. Каждой скобке ДНФ соответствует цепочка из последовательных соединенных контактов, определяемых переменными содержащимися в скобке. При этом, вся схема состоит из параллельных соединений указанных цепочек. Для приведенного примера соответствует схема приведена ниже.



Примеры построения некоторых функций



исключающее "или"



медиана

$x \oplus y = (\neg x \wedge y) \vee (x \wedge \cdot (x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y$

8.3. Задача о минимизации контактной схемы

Определение:

Две контактные схемы называются эквивалентными (англ. *equivalent contact circuits)*, если они реализуют одну и ту же булеву функцию.

Определение:

Сложностью контактной схемы (англ. *the complexity of the contact circuit*) называется число ее контактов.

Определение:

Минимальная контактная схема (англ. *minimal contact circuit*) — схема, имеющая наименьшую сложность среди эквивалентных ей схем.

Определение:

Дерево конъюнктов для nпеременных — двоичное ориентированное дерево глубиной n, такое что: поддеревья на одном и том же уровне одинаковы; и левое ребро любого узла помечено символом переменной $x_k (k \leqslant n)$, а правое помечено символом отрицания переменной x_k.

Задача минимизации контактных схем состоит в том, чтобы по данной схеме Sнайти схему T, эквивалентную Sи имеющую наименьшую сложность. Один из путей решения этой задачи состоит в следующем:

- Осуществляем переход от контактной схемы Sк её булевой функции $F(S)$.
- Упрощаем $F(S)$, то есть отыскиваем функцию G(на том же базисе, что и $F(S)$, равносильную $F(S)$и содержащую меньше вхождений операций дизъюнкции и конъюнкции. Для этой операции удобно использовать карты Карно.

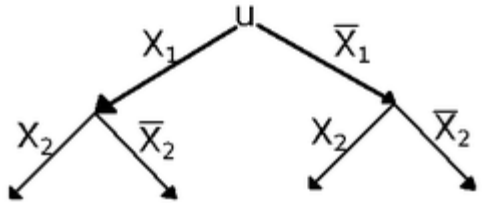
- Строим схему T, реализующую функцию G.

Теорема:

Любой булеву функцию можно представить контактной схемой, сложностью $O(2^n)$

Доказательство:

Пусть дана функция $f(x_1, x_2, \ldots, x_n)$ и она представлена в ДНФ



Дерево конъюнктов для 2-х переменных

Возьмем дерево конъюнктов для nпеременных (см. картинку). Очевидно, что от вершины uдо "нижних" вершин дерево можно добраться за $O(n)$, а ребер у такого дерева $O(2^n)$

Соединим нижние вершины, которые соответствуют конъюнктам функции, с вершиной vконтактами, над которыми написана 1. От этого в схему добавится не более, чем 2^nребер и тогда сложность останется $O(2^n)$.

В результате можно построить контактную схему для любой функции со сложностью $O(2^n)$

8.4. Моделирование контактных схем

КОНТАКТНАЯ СХЕМА

- специальная управляющая система, одна нз математических моделей реальных устройств, построенных из контактов реле. Контактная схема- модельный класс управляющих систем, и для него рассматриваются все те же задачи, что и для прочих классов управляющих систем; он особенно удобен при изучении "геометрических" свойств управляющих систем.

Контактная схема получается в результате приписывания каждому ребру некоторого графа с выделенными вершинами одной из букв конечного алфавита $\{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$. Выделенные вершины называются полюсами схемы. Ребро с приписанной ему буквой x_i $(\bar{x}_i)$ называется замыкающим (размыкающим) контактом. Последовательность контактов между полюсами *a* и *b* схемы *S,* соответствующая простой цепи в графе схемы *S,* называется цепью между полюсами *a* и *b* схемы S; конъюнкция соответствующих букв называется проводимостью данной цепи. Проводимость между полюсами a и b схемы есть функция алгебры логики $f_{ab}(x_1, \ldots, x_n)$, равная дизъюнкции проводимостей всех цепей между этими полюсами (в случае, если множество цепей между a и bпусто, $f_{ab}=0$; при *a,* совпадающем с b, $f_{aa}=1$). Всякой контактной схеме сопоставляется матрица проводимостей $\|f_{ab}\|$, элементами которой являются проводимости между парами полюсов. При этом $f_{ab} \cdot f_{bc} \leqslant f_{ac}$. Обратно, если задана матрица из функций алгебры логики $\|f_{ab}\|$ такая, что $f_{aa}=1$, $f_{ab}=f_{ba}$ и $f_{ab} \cdot f_{bc} \leqslant f_{ac}$ для любых *a*, *b, c,* то существует контактная схема с данными полюсами, для которых проводимости совпадают с заданными f_{ab}. В частности, для любой *f* существует двухполюсная схема, проводимость между полюсами которой равна *f*. В этом случае говорят, что схема реализует функцию *f*. Например, схема рис. 1 реализует линейную функцию $f = x_1 + \ldots + x_n + 1 \pmod 2$. Всякая функция алгебры логики реализуема некоторой контактной схемой.

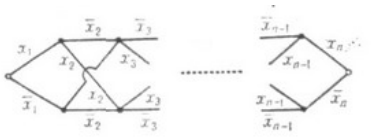


Рис. 1.

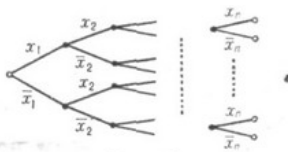


Рис. 2.

Иногда в контактной схеме множество всех полюсов разбито на два подмножества - входов и выходов. Контактная схема с r входами и s выходами называется контактным (r, s)-полюсником. Контактная схема , проводимости между любыми парами выходов (входов) которой равны нулю, называется разделительной относительно выходов (входов). Примером разделительного (относительно выходов) (1, 2^n)-полюсника может служить контактное *дерево* (рис. 2). Контактная схема называется плоской, если соответствующий ей граф, дополненный источниковым ребром (т. е. ребром, соединяющим полюсы, которому не приписана никакая буква рассматриваемого алфавита), является плоским. Плоская контактная схема *S** называется двойственной к плоской контактнойя схеме *S,* если граф Г* схемы *S** (с источниковым ребром) является двойственным к аналогичному графу Г схемы *S,* причем источниковому ребру графа Г соответствует таковое в графе *, а остальные соответствующие друг другу ребра несут одинаковые буквы (рис. 3). Схемы S и *S** имеют одинаковое число контактов и реализуют двойственные функции (принцип двойственности).

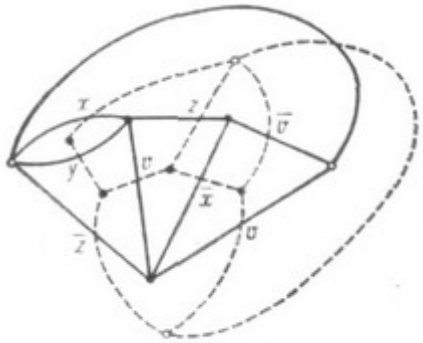


Рис. 3.

При замене контактов в схеме *S** на противоположные получается схема для отрицания функции, реализуемой схемой *S.* Для неплоских контактных схем, вообще говоря, не существует возможности перехода к схемам с тем же числом контактов, реализующим двойственные функции. П-схема (параллельно-последовательная схема) может быть определена индуктивно: контактная схема, состоящая из единственного контакта, соединяющего полюсы, есть П-схема; контактная схема , построенная из двух П-схем, соединенных параллельно или последовательно, есть П-схема. Существуют контактныея схемы , не являющиеся П-схемами, например, схема рис. 4. Контактная схема , двойственная к П-схеме, есть П-схема.

Рис. 4 .

Существует соответствие между П-схемамии формулами в базисе $\{\&, \vee, -\}$. При этом всякая П-схема реализует ту же самую функцию, что и соответствующая ей формула, и содержит столько же контактов, сколько букв содержит формула. Например, схеме рис. 5 соответствует формула

$$(x_1x_2 \vee \overline{x_1}\,\overline{x_2})\,(x_3x_4 \vee \overline{x_3}\,\overline{x_4}) \vee (x_1\overline{x_2} \vee \overline{x}_1x_2)\,(x_3\overline{x_4} \vee \overline{x}_3x_4).$$

Под сложностью контактной схемы понимается число всех ее контактов. Минимальное число контактов, достаточное для реализации контактной схемой произвольной функции алгебры логики, зависящей от ппеременных, асимптотически равно 2^n/n; минимальное число контактов, достаточное для реализации П-схемой, асимптотически равно 2^n/log n.

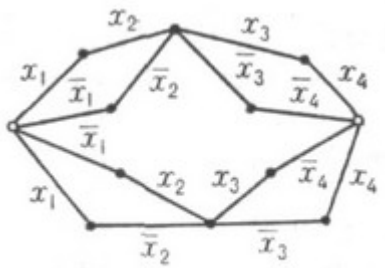


Рис. 5.

Две контактные схемы называются эквивалентными (при заданном взаимно однозначном соответствии между их полюсами), если проводимости между любой парой соответствующих полюсов этих схем совпадают. При замене в любой контактной схеме S любой ее подсхемы S' на эквивалентную получается схема, эквивалентная S. (При замене необходимо рассматривать как полюсы S' все попавшие в S' полюсы S и все вершины S', инцидентные не попавшим в S' контактам S.) Если S_1, S_2 – эквивалентные контактные схемы , то правило $S_1 \leftrightarrow S_2$ эквивалентного преобразования контактной схемы разрешает в любой схеме заменить подсхему, полученную из S_1 (или S_2) переименованием букв, на контактную схему , полученную из S_2 (S_1) тем же переименованием. Для каждого псуществует конечная полная система правил (рис. 6), позволяющая переводить друг в друга любые эквивалентные контактные схемы с числом переменных, не превосходящим n. Для класса же всех контактных схем (без ограничения числа переменных) конечной полной системы правил не существует (если при применении правил допускать только переименования букв).

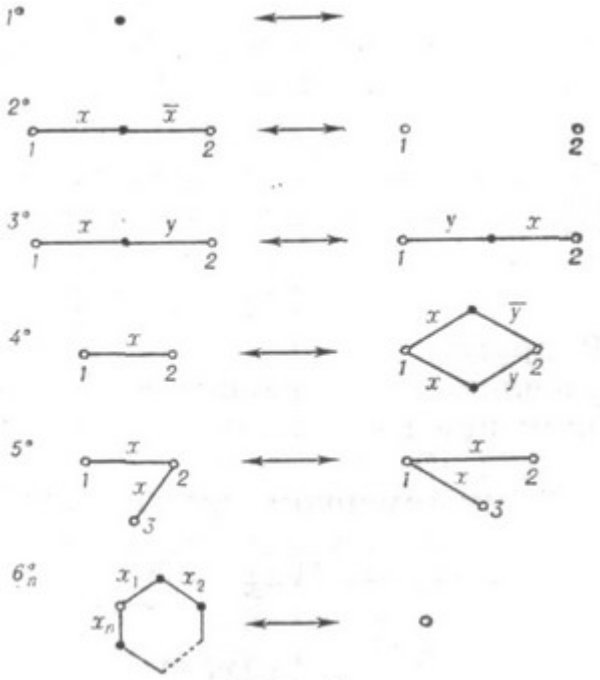


Рис. 6.

9. Логика высказывний

Логика предикатов в качестве составной и наиболее простой части содержит логику высказываний.

9.1. Высказывния и операции над ними

Высказывание – это повествовательное предложение, о котором можно

сказать истинно оно или ложно. Рассмотрим следующие предложения.

А. «Число Ö2 является иррациональным».

Б. «Неверно, что число Ö2 является иррациональным».

В. «Если число Ö2 является иррациональным, то число Ö2+1 также является иррациональным».

Г. «Который час?»

Д. «Идите решать задачу к доске»

Первые три предложения являются высказываниями, последние два – нет. Высказывания А и В истинны, высказывание Б – ложно. Более точно, *значение истинности* высказываний А и В *есть истина*, а *значение истинности* высказывания Б *есть ложь*. В дальнейшем истину будем обозначать цифрой 1, а ложь – цифрой 0.

Проанализируем высказывания А-В с точки зрения их «внутреннего строения». Высказывание А можно назвать простым. А высказывания Б и В – составными, полученными из простых высказываний А и Е=«число Ö2+1 является иррациональным». Этот простой пример показывает, что в языке (в данном случае, в русском языке) существуют способы построения одних высказываний из других. Эти способы мы будем называть *операциями.* В естественных языках (в том числе и в русском языке) существует много таких операций. Мы выделим в качестве основных пять операций.

Определение. Пусть X и Y – некоторые высказывания. Тогда высказывания:

1) «X и Y» называется конъюнкцией высказываний X и Y;

2) «X или Y» называется дизъюнкцией высказываний X и Y;

3) «не X» называется отрицанием высказывания X;

4) «если X, то Y» называется импликацией высказываний X и Y;

5) «X тогда и только тогда, когда Y» называется эквиваленцией высказываний X и Y.

Высказывание Б из вышеприведенного примера является отрицанием высказывание А, а высказывание В – импликацией высказываний А и Е. Введем следующие обозначения для операции: & – конъюнкция, Ú – дизъюнкция, Ø – отрицание, ® – импликация, « – эквиваленция. Так, Б=ØА, В=А®Е. Символы &, Ú, Ø, ®, « называются *связками.*

Зависимость значения истинности новых высказываний определяется *таблицей истинности связок* – таблицей 1.1.

Таблица 1.1

X	Y	X&Y	XÚY	ØX	X®Y	X«Y
1	1	1	1	0	1	1
1	0	0	1	0	0	0
0	1	0	1	1	1	0
0	0	0	0	1	1	1

Более точно, таблица 1.1 содержит пять таблиц истинности, по одной для каждой из связок. Эти пять таблиц для удобства объединены в одну.

Прокомментируем таблицы истинности дизъюнкции и импликации. В русском языке союз «или» понимается в двух смыслах: *разделительном* – или то, или другое, но не оба, и *соединительном* – или то, или другое, или оба. Как мы видим из таблицы 1.1 мы союз «или» будем понимать в соединительном смысле. Перейдем к импликации. Если дана импликация X®Y, то высказывание X называется *посылкой* импликации, а Y – *заключением*. Если посылка X импликации ложна, то вся импликация X®Y истинна (см. третью и четвертую строки таблицы 1.1). Это свойство импликации часто формулируют в виде следующего принципа: «из ложного утверждения(имеется в виду X) следует все что угодно (имеется в виду Y)». В силу этого следующее высказывание «если 2×2=5, то p – иррациональное число» является истинным, поскольку оно представляет собой импликацию,

посылка которой ложна. Подчеркнем, что при этом не надо искать доказательство или опровержение того, что p – иррациональное число. Аналогично, первая и третья строки таблицы 1.1 показывают нам. Что если заключение Y импликации истинно, то вся импликация X®Y также истинна. Это свойство импликации тоже формулируют в виде принципа: «истинное утверждение (имеется в виду Y) следует из чего угодно (имеется в виду X)». Из этого принципа сразу следует истинность высказывания «если p – иррациональное число, то 2×2=4», поскольку оно представляет собой импликацию с истинным заключением.

9.2. Формулы логики высказываний, интерпретация

В первом параграфе высказывания были введены как повествовательные предложения естественного языка, т.е. как лингвистические объекты. Для изучения этих объектов математическими средствами используется понятие формулы логики высказываний. Дадим соответствующие определения.

Определение. *Атомарными формулами логики высказываний* называются буквы U,V,W,X,Y,Z с индексами и без них, а также символы истины 1 и лжи 0.

Определение. *Формулами логики высказываний* называются

1) атомарные формулы;

2) выражения вида (F)&(G), (F)Ú(G), Ø(F), (F)®(G), (F)«(G), где F и G –

формулы логики высказываний.

На первый взгляд может показаться, что определение содержит «порочный круг»; «понятие формулы логики» высказываний определяется само за себя. На самом деле, это определение относится к так называемым индуктивным определениям. Такие определения вводят сначала базовые объекты (в нашем случае – атомарные формулы) и способы порождения новых объектов из уже полученных (в нашем случае – применение операций), введенных в

первом параграфе.

Приведем пример. Буквы X,Y,Z – атомарные формулы. В силу первого пункта определения эти буквы являются формулами логики высказываний, а в силу второго формулами являются выражения (X)&(Y), ((X)&(Y))®(Z). Мы видим, что если следовать строго определению, в формуле надо писать много скобок. Это неудобно для восприятия формулы. Чтобы уменьшить количество скобок условимся, во- первых, атомарные формулы в скобки не заключать, во-вторых, ввести приоритет (силу связывания) для связок. Будем считать, что Ø имеет наивысший приоритет, & и Ú имеют одинаковый приоритет, который выше, чем у® и «. Последние две связки имеют одинаковый приоритет. Используя эти соглашения формулу ((X)&(Y))®(Z) можно записать так: X&Y®Z. Отметим, что поскольку мы не упорядочили & и Ú по силе связывания, то выражение X&YÚZ не является формулой. Надо в этом выражении поставить скобки, определяющие порядок выполнения операций. Получатся две формулы (X&Y)ÚZ и X&(YÚZ).

В дальнейшем нам понадобится понятие подформулы. Попросту говоря, подформула формулы F – это «слитная» часть, которая сама является формулой. На строгом уровне понятие вводится следующим образом.

Определение. *Подформулой* атомарной формулы является она сама. *Подформулами* формулы ØF являются формула ØF и все ее подформулы. Подформулами формул F&G, FÚG, F®G, F«G являются они сами и все подформулы формул F и G/

Например, формула F=X&Y®XÚZ имеет шесть подформул: X,Y,Z,X&Y, XÚZ, X&Y®XÚZ.

Теперь нам надо соотнести понятие высказывания и формулы. На самом простом уровне формула – это *форма* для получения высказываний. Пусть, например, дана формула F=X&Y®Z. Поставим вместо X,Y и Z соответственно высказывания A_1=четырехугольник ABCD является параллелограммом, A_2=в четырехугольнике ABCD смежные стороны равны, A_3=в четырехугольнике ABCD диагонали перпендикулярны, то получим высказывание A_4= если четырехугольник ABCD является параллелограммом и его смежные стороны равны, то диагонали перпендикулярны (использованы естественные сокращения). Это высказывание получилось «по форме» F. Если вместо X,Y и Z подставить другие высказывания, то

получим новое высказывание, имеющее ту же «форму».

На строгом уровне сказанное в предыдущем абзаце оформляется в виде понятия интерпретации.

Обозначим через A – множество атомарных, а через F – множество всех формул логики высказываний. Зафиксируем некоторую совокупность высказываний P, удовлетворяющим следующим условиям: если совокупность P содержит два высказывания, то она содержит их конъюнкцию, дизъюнкцию, импликацию и эквиваленцию и отрицание (каждого из высказываний). *Интерпретацией в широком смысле* мы будем называть функцию

$\varphi: A \to P$

такую, что $\varphi(1)$ – истинное высказывание, а $\varphi(0)$ – ложное. Такая функция, определенная на множестве атомарных формул, естественным образом распространяется на множество всех формул. Выше был приведен пример интерпретации в широком смысле. В этом примере совокупность P содержала высказывания A_1, – A_4, а интерпретация φ на атомарных формулах X,Y,Z действовала так: $\varphi(X)=A_1$, $\varphi(Y)=A_2$, $\varphi(Z)=A_3$. Естественно расширение φ на множестве всех формул будем обозначать той же буквой. Тогда $\varphi(F)=A_4$.

В дальнейшем, на самом деле от высказываний $\varphi(F)$ нам, в основном, будут нужны только их истинные значения 1 и 0. Введем поэтому более узкое понятие интерпретации.

Определение. *Интерпретацией в узком смысле* (или просто *интерпретацией*) называется функция

$\varphi: A \to \{0,1\}$

такая, что $\varphi(0)=0$, $\varphi(1)=1$.

Используя таблицы истинности связок, интерпретацию можно расширить на множество всех формул. Приведем пример. Пусть $\varphi(X)=1$, $\varphi(Y)=0$, $\varphi(Z)=1$,

F=XÚY®Z, G=X&Y«Y&Z. Тогда $\varphi(F)=1$, $\varphi(G)=0$.

В заключение параграфа рассмотрим задачу, решение которой состоит в использовании выразительных возможностей логики высказываний. Прежде, чем дать соответствующее определение условимся о следующем обозначении. Если формула F построена из атомарных формул $X_1,\ldots,X_n$, то F будем обозначать через $F(X_1,\ldots,X_n)$. Более того, мы будем пользоваться последним обозначением, даже если некоторые из атомарных формул отсутствуют в записи формулы F (но всякая атомарная формула, входящая в F, содержится среди $X_1,\ldots,X_n$).

9.3. Равносильность и законы логики высказываний

Нетрудно привести примеры формул, которые «выражают одно и то же». Таковы, например, формулы XÚY и YÚX. Подобные формулы мы будем называть *равносильными*. Прежде, чем дать соответствующее определение, условимся о следующем обозначении.

Определение. Формулы F и G называются *равносильными*, если для любой интерпретации φ выполняется равенство $\varphi(F)=\varphi(G)$.

Убедимся в том, что формулы F=X®Y и G=ØXÚY равносильны. Ясно, что если интерпретации φ и ψ совпадают на X и Y, то $\varphi(F)=\psi(F)$ и $\varphi(G)=\psi(G)$. Следовательно, для проверки равенства $\varphi(F)=\varphi(G)$ из определения равносильности надо рассмотреть лишь интерпретации, которые различаются на X и Y (а таких интерпретаций четыре) и вычислить соответствующие значения $\varphi(F)$ и $\varphi(G)$. Другими словами, надо составить совместную таблицу истинности формул F и G (см. таблицу 1.2).

Таблица 1.2

X	Y	F=X®Y	ØX	G=ØXÚY
1	1	1	0	1
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

В таблице 1.2 для удобства вычисления значения интерпретаций на G введен

промежуточный столбец ØX. Мы видим, что столбцы формул F и G совпадают. Это означает, что формулы F и G равносильны.

Близким к понятию равносильности является понятие тождественной истинности.

Определение. Формула F называется *тождественно истинной* если для любой интерпретации j выполняется равенство j(F)=1.

Например, формула F=X&Y®X является тождественно истинной. Для проверки равенства j(F)=1 не надо рассматривать все интерпретации, а лишь четыре, которые различаются на атомарных формулах X и Y. Для таких интерпретаций надо вычислить значение формулы F, т.е. составить таблицу истинности формулы F (см. таблицу 1.3). Таблица 1.3 для удобства вычисления значения j(F) содержит промежуточный столбец X&Y.

Таблица 1.3

X	Y	X&Y	F=X&Y®X
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	1

Мы видим, что столбец формулы F состоит из одних единиц. Это означает, что формула F тождественно истинна.

Теорема 1.1. Формулы F и G равносильны тогда и только тогда, когда формула F«G является тождественно истинной.

Доказательство. Предположим, что формулы F и G равносильны и рассмотрим интерпретацию j. Ясно, что j(F«G)=j(F)«j(G). Поскольку значения истинности j(F) и j(G) совпадают, то по таблице истинности эквиваленции имеем равенства j(F)«j(G)=1. Это означает, что формула F«G тождественно истинна.

Предположим теперь, что формула F«G тождественно истинна и рассмотрим интерпретацию j. Имеем, что 1=j(F«G)=j(F)«j(G). Но из таблицы

истинности эквиваленции следует, что если j(F)«j(G)=1, то j(F)=j(G).

Теорема доказана.

В логике высказываний довольно часто приходится проводить преобразования формул, сохраняющие равносильность. Для таких преобразований используются так называемые *законы логики высказываний*. Приведем список этих законов.

Пусть F,G и H – некоторые формулы логики высказываний. Тогда следующие формулы равносильны:

1) F&1 и F;
2) FÚ1 и F;
3) F&0 и 0;
4) FÚ0 и F;
5) F&F и F;
6) FÚF и F;
7) F&G и G&F;
8) FÚG и GÚF;
9) F&(G&H) и (F&G)&H;
10) FÚ(GÚH) и (FÚG)ÚH;
11) F&(GÚH) и (F&G)Ú(F&H);
12) FÚ(G&H) и (FÚG)&(FÚH);
13) F&(FÚG) и F;
14) FÚ(F&G) и F;
15) F&ØF и 0;
16) FÚØF и 1;
17) Ø(F&G) и ØFÚØG;
18) Ø(FÚG) и ØF&ØG;
19) ØØF и F;
20) F®G и ØFÚG;
21) F«G и (F®G)&(G®F).

Доказательство этих равносильностей (законов логики высказываний) легко получается с помощью таблиц истинности. Отметим, что в примере на определение равносильности мы фактически доказали закон 20.

Прокомментируем список законов. Законы 5 и 6 называются *идемпотентностью*, 7 и 8 – *коммутативностью*, 9 и 10 – *ассоциативностью* соответственно конъюнкции и дизъюнкции. Ассоциативность конъюнкции означает, что в конъюнкции трех формул скобки можно ставить как угодно, а, следовательно, вообще не ставить. Из этого утверждения следует, что в конъюнкции четырех, пяти и т.д. любого конечного числа формул скобки можно ставить как угодно и поэтому вообще не ставить. Аналогичное замечание можно сделать и для дизъюнкции.

Законы 11 и 12 называются *дистрибутивностями*. Более точно, закон 11 – дистрибутивность конъюнкции относительно дизъюнкции, а закон 12 – дистрибутивность дизъюнкции относительно конъюнкции. Для применения этих законов в преобразованиях формул удобно иметь в виду следующий аналог. Заменим в законе 11 формулы F,G и H соответственно буквами a,b и c, знак & заменим умножением *, а знак Ú – сложением+. Мы получим известное числовое тождество

a*(b+c)-a*b+a*c (*)

Это тождество есть дистрибутивность умножения чисел относительно сложения. В школе применение этого равенства слева направо называется раскрытием скобок, а справа налево вынесением общего множителя. Отличие операций над высказываниями & и Ú от числовых операций * и+состоит в том, что для высказываний выполняются обе дистрибутивности, а для чисел только одна. Сложение не дистрибутивно относительно умножения.

Закон 15 называется *законом противоречия,* закон 16 – *законом исключенного третьего,* закон 19 – *снятием двойного отрицания*. Законы 13 и 14 называются *законами поглощения*, а законы 17 и 18 – *законами де Моргана* в честь известного французского математика и логика 19 века.

Имея законы логики высказываний, мы получаем еще один способ доказательства равносильности двух формул, наряду с построением совместной таблицы истинности. Этот способ состоит в переходе от одной формулы к другой с помощью законов. В его основе лежит следующее легко доказываемое утверждение: если в некоторой формуле F заменить подформулу G равносильной ей формулой G′, то получим формулу F′, равносильную исходной формуле F. Проиллюстрируем второй способ на следующем примере: доказать равносильность формул

F=[X&(Z®Y)]Ú[(X®Z)&Y] и

G=(XÚY)&(YÚØZ).

В силу закона 20, формулы Z®Y и X®Z равносильны соответственно

формулам ØZÚY и ØXÚZ, поэтому формула F равносильна формуле

F_1=[X&(ØZÚY)]Ú[(ØXÚZ)&Y].

Дважды применив дистрибутивность (закон 11) и пользуясь ассоциативностью связок & и Ú, получим, что формула F_1 равносильна формуле

F_2=(XÚØXÚZ)&(ØZÚYÚØXÚZ)&(XÚY)&(ØZÚYÚY).

В силу коммутативности дизъюнкции, законов 16 и 2, формулы XØXÚZ и ØZÚYÚØXÚZ равносильны 1. Применив теперь законы 1 и 6 и коммутативность дизъюнкции, получим, что формула F_2 равносильна G.

9.4. Логическое следствие

Одна из основных целей изучения логики состоит в получении формального аппарата для доказательства того, является ли данное утверждение следствием других.

Введем необходимые понятия.

Определение. Формула G называется *логическим следствием* формул $F_1,F_2,\ldots,F_k$, если для любой интерпретации j из того, что $j(F_1)=j(F_2)=\ldots=j(F_k)=1$ следует, что j(G)=1.

В качестве примера рассмотрим следующую задачу: выяснить логично ли рассуждение молодого человека из §9.2. Напомним, что это рассуждение мы перевели на ….

Приведем противоположный пример. Докажем, что формула G=Y®X не является логическим следствием формул F_1=XÚY, F_2=X®Y, F_3=Y. Для этого построим совместную таблицу истинности формул F_1,F_2,F_3 и G.

Таблица 1.4

X	Y	$F_1=X\acute{U}Y$	$F_2=X\circledR Y$	$F_3=X$	$G=Y\circledR X$
1	1	1	1	1	1
1	0	1	0	0	1
0	1	1	1	1	0
0	0	0	1	0	1

Мы видим (см. таблицу 1.4), сто если взять интерпретацию j, для которой j(x)=0, j(y)=1, (т.е. взять третью строку таблицы), то $j(F_1)=j(F_2)=j(F_3)=1$, но j(G)=0. Следовательно, формула G не является логическим следствием формул F_1,F_2,F_3.

Понятие логического следствия тесно связано с понятием выполнимости.

Определение. Множество формул $\{F_1,F_2,\ldots,F_l\}$ называется *выполнимым*, если существует интерпретация j такая, что $j(F_1)=j(F_2)=\ldots=j(F_l)=1$.

Проверку выполнимости множества формул $\{F_1,F_2,\ldots,F_l\}$ можно провести построением совместной таблицы истинности этих формул. Если найдется хотя одна строка, в которой в столбцах формул $F_1,F_2,\ldots,F_l$ стоят единицы, то это множество формул выполнимо. Если такой строки нет, то множество формул невыполнимо. Так, множество формул $\{F_1,F_2,F_3,G\}$ из предыдущего примера выполнимо, поскольку в таблице 1.4 в первой строке в столбцах этих формул стоят единицы.

Дальше нам понадобится следующее утверждение.

Теорема 1.2 Формула G является логическим следствием формул $F_1,F_2,\ldots,F_k$ тогда и только тогда, когда множество формул

$L=\{F_1,F_2,\ldots,F_k,\O G\}$

невыполнимо.

Доказательство. Пусть формула G является следствием множества формул $F_1,\ldots,F_k$. Предположим, от противного, что множество L выполнимо. Это означает, что существует интерпретация y такая, что $y(F_1)=\ldots=y(F_k)=y(\O G)=1$. Но если $y(F_1)=\ldots=y(F_k)=1$, то y(G)=1, поскольку G – логическое следствие формул $F_1,\ldots,F_k$. Полученное противоречие $y(\O G)=1$

и y(G)=1 доказывает, что множество формул $\{F_1,\ldots,F_k,\O G\}$ невыполнимо.

Пусть теперь множество формул L невыполнимо. Рассмотрим интерпретацию j такую, что $j(F_1)=\ldots=j(F_k)=1$. Поскольку L невыполнимо, то $j(\O G)=0$. Если $j(\O G)=0$, то j(G)=1. Следовательно, из равенств $j(F_1)=\ldots=j(F_k)=1$ следует равенство j(G)=1. Это означает, что G – логическое следствие множества формул $F_1,\ldots,F_k$.

9.5. Нормальные формы в логике высказываний

Среди множества формул, равносильных данной, выделяют формулы, имеющие ту или иную нормальные формы.

Дадим необходимые определения.

Определение. *Литералом* называется атомарная формула (кроме 1 и 0) или ее отрицание.

Элементарной конъюнкцией называется литерал или конъюнкция литералов.

Определение. Формула G имеет *дизъюнктивную нормальную форму* (сокращенно: ДНФ), если она является элементарной конъюнкцией или дизъюнкцией элементарных конъюнкций.

Например, формулы $X,\O Y,X\&\O Y,(X\&\O Y)\acute{U}(\O X\&Z)$ имеют ДНФ, а формулы $\O(X\&Y)$, $X\acute{U}Y\acute{U}1$, $X\circledR Y$ не имеют.

Теорема 1.3. Для всякой формулы F существует формула G, равносильная F и имеющая дизъюнктивную нормальную форму.

Теорема легко следует из рассмотрения следующего алгоритма, который по данной формуле F выдает (одну из формул) G, удовлетворяющих условию теоремы.

Прежде, чем привести алгоритм, условимся не различать формулы, которые получаются одна из другой применением коммутативности и ассоциативности конъюнкции и дизъюнкции, т.е. законов 7-10.

Алгоритм приведения к ДНФ.

Шаг 1. Используя законы 21 и 20 исключить из исходной формулы эквиваленцию и импликацию.

Шаг 2. С помощью законов 17-19 занести отрицание к атомарным формулам.

Шаг 3. Если формула содержит подформулу вида

$H_1\&(H_2ÚH_3)$,

то заменить ее на равносильную формулу

$(H_1\&H_2)Ú(H_1\&H_3)$.

Применение алгоритма проиллюстрируем на примере формулы

F=Ø(X«Y)&X.

Выполним первый шаг. Для этого, используя закон 21, заменим X«Y равносильной ей формулой (X®Y)&(Y®X). Затем в полученной формуле с помощью закона 20 исключим связку ®. Мы получим формулу

F_1=Ø[(ØXÚY)&(ØYÚX)]&X.

Перейдем ко второму шагу. Применение закона 17, приведет к формуле

F_2=[Ø(ØXÚY)ÚØ(ØYÚX)]&X.

Затем дважды воспользуемся законом 18 и снимем двойное отрицание (закон 19), получим формулу

F_3=[(X&ØY)Ú(Y&ØX)]&X.

Шаг 2 выполнен.

Выполнение шага 3 состоит из применения дистрибутивности к формуле F_3.

Это дает нам формулу

F_4=(X&ØY&X)Ú(Y&ØX&X).

Алгоритм на этом завершен. Формула F_4 имеет дизъюнктивную нормальную форму. Но эту формулу можно упростить. Действительно, формула Y&ØX&X в силу законов 15 и 3 равносильна 0, а формула X&ØY&X равносильна X&ØY (закон 5). Следовательно, формула F_4 равносильна формуле

F_5=X&ØY.

Формула F_5, как и F_4, имеет ДНФ и равносильна исходной формуле F.

Рассмотренный пример показывает, что формула F может иметь не одну равносильную формулу, именющую ДНФ. Иногда это обстоятельство является неудобным. Чтобы его исключить, вводится более узкое понятие, нежели ДНФ.

Определение. Формула G имеет *совершенную дизъюнктивную нормальную форму* (сокращенно: СДНФ) *относительно атомарных формул* $X_1,\ldots,X_n$, если выполнены следующие условия:

1) F=F(X1,…,Xn), т.е. в записи формулы участвуют только X1,…,Xn;

2) F имеет дизъюнктивную нормальную форму, т.е. F=C1ÚC2Ú…ÚCk, где C1,…,Ck – элементарные конъюнкции;

3) каждая элементарная конъюнкция содержит один и только один из литералов Xi или ØXi для любого i=1,…,n;

4) F не содержит одинаковых элементарных конъюнкций.

Например формулы X, ØX&Y, (ØX&Y)Ú(X&ØY) имеют СДНФ относительно содержащихся в них атомарных формул. Формулы Ø(X&Y), (X&Y)Ú(ØX&Z), (X&Y&X)&(ØX&ØY), (X&Y)Ú(X&ØY)&(Y&X) не имеют СДНФ (относительно содержащихся в них атомарных формул). Для первой формулы не выполняется второе условие, для второй и третьей – третье условие, для четвертой формулы не выполняется последнее условие из

определения СДНФ.

Теорема 1.4. Для любой выполнимой формулы F существует равносильная ей формула G, имеющая совершенную дизъюнктивную нормальную форму.

Как и теорема 1.3, эта теорема легко следует из соответствующего алгоритма, который по формуле F выдает формулу G, удовлетворяющую условию теоремы 1.4.

Алгоритм приведения к СДНФ.

Шаг 1 – Шаг 3 – те же, что и в алгоритме приведения к ДНФ.

Шаг 4. Если элементарная конъюнкция C не содержит атомарной формулы X_i, ни ее отрицания для некоторого i=1,…,n, то заменить C на две элементарные конъюнкции (C&X)Ú(C&ØX_i).

Шаг 5. Если элементарная конъюнкция C содержит два вхождения одного литерала, то одно из них вычеркнуть. Если же C содержит X_i и ØX_i для некоторого i=1,…,n, то вычеркнуть всю элементарную конъюнкцию.

Шаг 6. Если формула содержит одинаковые элементарные конъюнкции, то вычеркнуть одну из них.

Напомним, что «одинаковость» понимается с точностью до коммутативности и ассоциативности конъюнкции и дизъюнкции.

В качестве примера рассмотрим ту же формулу F=Ø(X«Y)&X, что и в примере на предыдущий алгоритм. Как мы видели, выполнение шагов 1-3 приводит к формуле F_4. Эта формула имеет ДНФ, но не имеет СДНФ, поскольку для нее не выполняется третье условие. Если для F_4 выполнить шаг 4, то в первой элементарной конъюнкции будет зачеркнуто одно из вхождений литерала X, а вторая элементарная конъюнкция будет вычеркнута вся. В результате мы получим формулу F_5. Она имеет СДНФ относительно X и Y.

Рассмотрим второй пример. Пусть G=(X&Y)Ú(X&ØZ). Эта формула имеет ДНФ, поэтому выполнение алгоритма приведения к СДНФ начинается с шага 4. При выполнении этого шага элементарная конъюнкция X&Y будет

заменена на (X&Y&Z)Ú(X&Y&ØZ), а X&ØZ – на (X&ØZ&Y)Ú(X&ØZ&ØY). В результате получим формулу

G_1=(X&Y&Z)Ú(X&Y&ØZ)Ú(X&ØZ&Y)Ú(X&ØZ&ØY).

Условия шага 5 для формулы G_1 ложны, поэтому этот шаг формулы G_1 не имеет. Формула G_1 содержит одинаковые элементарные конъюнкции – вторую и третью. При выполнении шестого шага будет зачеркнута одна из них и получится формула

G_2=(X&Y&Z)Ú(X&Y&ØZ)Ú(X&ØY&ØZ).

Это и есть формула, равносильная G и имеющая СДНФ относительно входящих в G атомарных формул.

Ответим на естественно возникающий вопрос о том, зачем в формулировке теоремы 1.4 требуется выполнимость формулы F? Нетрудно доказать, что если формула F невыполнима, т.е. при любой интерпретации j выполняется равенство j(F)=0, то после приведения F к ДНФ каждая элементарная конъюнкция будет содержать хотя бы одну пару противоположных литералов X и ØX. Но в таком случае на шаге 5 все элементарные конъюнкции будут вычеркнуты.

Имеется еще один способ приведения к СДНФ, основанный на построении таблицы истинности исходной формулы. Изложим этот способ на примере формулы

F=X_1&(X_2®X_3)

составим таблицу истинности формулы F (таблицу 1.5).

Таблица 1.5

X_1	X_2	X_3	X_2®X_3	F=X_1&(X_2®X_3)
1	1	1	1	1
1	1	0	0	0
1	0	1	1	1
1	0	0	1	1

0	1	1	1	0
0	1	0	0	0
0	0	1	1	0
0	0	0	1	0

Выделим строки, в которых в столбце F стоит 1. (Хотя бы одна такая строка должна быть, так как формула F выполнима.) Это будут первая, третья и четвертая строки. Каждой из выделенных строк поставим в соответствие элементарную конъюнкцию $X_1^{a1}X_2^{aa}X_3^{a3}$, где $X_i^{ai}=X_i$, если в столбце X_i этой строки стоит 1, и $X_i^{ai}=ØX_i$, если в столбце X_i этой строки стоит 0, где i=1,2,3. Так, первой строке будет поставлена в соответствие элементарная конъюнкция $X_1\&X_2\&X_3$, третьей – $X_1\&ØX_2\&X_3$, четвертой – $X_1\&ØX_2\&ØX_3$. Формула

$$G=(X_1\&X_2\&X_3)Ú(X_1\&ØX_2\&X_3)Ú(X_1\&ØX_2\&ØX_3)$$

имеет СДНФ относительно X_1,X_2,X_3. В то же время G имеет ту же таблицу истинности, что и F. Это означает, что G равносильна F. Следовательно, G – искомая формула.

Из других нормальных форм рассмотрим конъюнктивную нормальную форму. Она получается из ДНФ заменой & на Ú и Ú на &. Дадим точные определения.

Определение. *Элементарной дизъюнкцией* (или дизъюнктом) называется литерал или дизъюнкция литералов.

Определение. Формула G имеет *конъюнктивную нормальную форму* (сокращенно: КНФ)Б если она является элементарной дизъюнкцией или конъюнкцией элементарных дизъюнкций.

Например, формулы X,ØY, XÚØY, X&ØY, (XÚØY)&(XÚZ) имеют КНФ, а формулы X®Y, Ø(XÚY), (X&ØY)Ú(X&ØZ) не имеют.

Для конъюнктивных нормальных форм справедливо утверждение, аналогичное теореме 1.3.

Теорема 1.5 Для всякой формулы F существует формула G равносильная F

и имеющая конъюнктивную нормальную форму.

Доказательство теоремы легко следует из анализа алгоритма приведения к КНФ, который в свою очередь получается из алгоритма приведения к ДНФ, если шаг 3 заменить на

Шаг 3′. Если формула содержит подформулу вида

$$H_1Ú(H_2\&H_3),$$

то заменить ее на равносильную ей формулу

$$(H_1ÚH_2)\&(H_1ÚH_3).$$

В силу очевидной аналогии между ДНФ и КНФ примеров приведения к КНФ здесь приводить не будем.

10. Логика первого порядка

Ло́гика пе́рвого поря́дка, называемая иногда *логикой* или *исчислением предикатов* — формальное исчисление, допускающее высказывания относительно переменных, фиксированных функций и предикатов. Расширяет логику высказываний. В свою очередь является частным случаем логики высших порядков.

Логика высказываний обладает довольно слабыми выразительными возможностями. В ней нельзя выразить даже очень простые с математической точки зрения рассуждения. Рассмотрим, например, следующее умозаключение. «Всякое целое число является рациональным. Число 2 – целое. Следовательно, 2 – рациональное число». Все эти утверждения с точки зрения логики высказываний являются атомарными. Средствами логики высказываний нельзя вскрыть внутреннюю структуру и поэтому нельзя доказать логичность этого рассуждения в рамках логики высказываний. Мы рассмотрим расширение логики высказываний, которое называется логика предикатов первого порядка или короче: логика первого порядка.

10.1. Предикаты и операции над ними

Введем основное понятие.

Определение. Пусть M – непустое множество. Тогда n-*местным предикатом, заданным на M,* называется выражение, содержащее n переменных и обращающееся в высказывание при замене этих переменных элементами множества M.

Рассмотрим примеры. Пусть M есть множество натуральных чисел **N**. Тогда выражения «x – простое число», «x – четное число», «x – больше 10» являются одноместными предикатами. При подстановке вместо x натуральных чисел получаются высказывания: «2 – простое число», «6 – простое число», «3 – четное число», «5 больше 10» и т.д. Выражения «x больше y», «x делит y нацело», «x плюс y равно 10» являются двухместными предикатами. (Конечно, последнее выражение можно было записать и так: «x+y=10»). Примеры трехместных предикатов, заданных на множестве натуральных чисел: число z лежит между «x и y», «x плюс y равно z», «|x-y|=z».

Будем считать, что *высказывание – нульместный предикат,* то есть предикат, в котором нет переменных для замены.

Надо отметить, что местность предикатов не всегда равна числу *всех* переменных, содержащихся в выражении. Например, выражение «существует число x такое, что y=2x» на множестве натуральных чисел определяет одноместный предикат. По смыслу этого выражение в нем можно заменять только переменную y. Например, замена y на 6 дает истинное высказывание: «существует число x такое, что 6=2x», а замена y на 7 дает ложное (на множестве **N**) высказывание «существует число x такое, что 7=2x».

Предикат с заменяемыми переменными $x_1,\ldots,x_n$ будет обычно обозначаться заглавной латинской буквой. После которой в скобках указаны эти переменные. Например, $P(x_1,x_2)$, $Q(x_2,x_3)$, $R(x_1)$. Среди переменных в скобках могут быть и фиктивные.

На совокупности всех предикатов, заданных на множестве M, вводятся знакомые операции *конъюнкция, дизъюнкция, отрицание, импликация* и *эквиваленция*. Эти операции вводятся довольно очевидным образом. Приведем в качестве примера определение конъюнкции предикатов.

Определение. Предикат $W(x_1,\ldots,x_n)$ называется конъюнкцией предикатов $U(x_1,\ldots,x_n)$ и $V(x_1,\ldots,x_n)$, заданных на множестве M, если для любых $a_1,\ldots,a_n$ из M высказывание $W(a_1,\ldots,a_n)$ есть конъюнкция высказываний $U(a_1,\ldots,a_n)$ и $V(a_1,\ldots,a_n)$.

Легко по аналогии привести определения и других упомянутых выше операций.

В логике предикатов первого порядка вводятся и две новые операции. Называются они квантором общности и квантором существования. Эти операции рассмотрим вначале на примерах. Пусть дано выражение «существует x такой, что x+y=10». На множестве натуральных чисел это предложение определяет одноместный предикат P(y), так P(2) и P(9) – истинные высказывания, P(11) – ложное. Если обозначить предикат «x+y=10» через S(x,y) (а это предикат двухместный), то P(y) можно записать так: «существует x такой, что S(x,y)». В этом случае говорят, что предикат P(y) получен из предиката S(x,y) навешиванием квантора существования на x и пишут P(y)=($x)S(x,y). Рассмотрим другой пример. Выражение «для всех x справедливо, что y^3-x^2» определяет на множестве целых чисел одноместный предикат Q(y). Если предикат «y^3-x^2» обозначить через T(x,y), то Q(y) можно записать так: «для всех x справедливо T(x,y)». В таком случае говорят, что предикат Q(y) получен из предиката T(x,y) навешиванием квантора общности на x и пишут Q(y)=("x)T(x,y).

После этих примеров нетрудно дать определение в общем виде.

Определение. Пусть $P(x_1,\ldots,x_n)$ – предикат, заданный на множестве M, y – переменная. Тогда выражение «для всякого y выполняется $P(x_1,\ldots,x_n)$» – предикат, полученный из P *навешиванием квантора общности на переменную y,* а выражение «существует y такой, что выполняется $P(x_1,\ldots,x_n)$» – предикат, полученный из P *навешиванием квантора существования на переменную y*.

Обозначения операций были введены выше.

Заметим, что в определении не требуется, чтобы y была одна из переменных $x_1,\ldots,x_n$, хотя в содержательных примерах, которые будут ниже, квантор навешивается на одну из переменных $x_1,\ldots,x_n$. Указанное требование не накладывается, чтобы избежать усложнения определения формулы логики предикатов. Если y – одна из переменных $x_1,\ldots,x_n$, то после навешивания квантора на y новый предикат является (n-1)-местным, если y U { $x_1,\ldots,x_n$}, то местность нового предиката равна n.

Если предикат $W(x_1,\ldots,x_n)$ получен из предикатов $U(x_1,\ldots,x_n)$ и $V(x_1,\ldots,x_n)$ с помощью связок, то истинность высказывания $W(a_1,\ldots,a_n)$ определяется таблицами истинности этих связок. Пусть $W(x_1,\ldots,x_n)=("y)U(x_1,\ldots,x_n,y)$. Тогда высказывание $W(a_1,\ldots,a_n)$ истинно тогда и только тогда, когда для любого bÎM истинно высказывание $U(a_1,\ldots,a_n,b)$. Если же $W(x_1,\ldots,x_n)=(\$y)U(x_1,\ldots,x_n,y)$, то высказывание $W(a_1,\ldots,a_n)$ истинно в том и только в том случае, когда найдется bÎM, для которого высказывание $U(a_1,\ldots,a_n)$ истинно.

Понятие предиката – весьма широкое понятие. Это видно уже из приведенных выше примеров. Тем не менее мы это еще раз подчеркнем, показав, что n-местная функция может рассматриваться как (n+1)-местный предикат. Действительно, функции $y=f(x_1,\ldots,x_n)$, заданной на множестве M можно поставить в соответствие выражение «y равно $f(x_1,\ldots,x_n)$». Это выражение есть некоторый предикат $P(x_1,\ldots,x_n,y)$. При этом если элемент b есть значение функции в точке $(a_1,\ldots,a_n)$, то высказывание $P(a_1,\ldots,a_n,b)$ истинно, и обратно. (Подобное «превращение» функции в предикат мы уже делали выше для сложения натуральных чисел.)

На предикаты можно смотреть и более формально, причем с двух точек зрения.

Во-первых, предикат можно представить отношением следующим образом. Пусть предикат $P(x_1,\ldots,x_n)$ задан на множестве M. Рассмотрим прямую степень этого множества $M^n=MxMx\ldots xM$ и подмножество D_p множества M^n, определяемое равенством:

$D_p=\{(a_1,\ldots,a_n)$ÎM^{n}½высказывание $P(a_1,\ldots,a_n)$ истинно$\}$.

Отношение D_p можно назвать областью истинности предиката P. Во многих случаях предикат P можно отождествить с отношением D_p. При этом, правда возникают некоторые трудности при определении операций над отношениями, аналогичными операциям над предикатами.

Во-вторых, предикат $P(x_1,\ldots,x_n)$, заданный на M, можно отождествить с функцией $f_p:M^n$®{0,1}, определяемой равенством

$$f_p(a_1,\ldots,a_n)=\begin{cases}1, & \text{если высказывание } P(a_1,\ldots,a_n) \text{ истинно,}\\ 0 & \text{в противном случае.}\end{cases}$$

Мы, в основном, будем понимать термин «предикат» в смысле исходного определения, т.е. как языковое выражение. Связано это с тем, что одной из главный целей, как уже отмечалось во введении, является изучение выразительных возможностей логики первого порядка, возможности представления средствами этой логики информации, выраженного на естественном (скажем, русском) языке.

10.2. Формулы логики первого порядка

Целью параграфа является введение понятия, вынесенного в заголовок параграфа. В принципе это делается так же, как и в логике высказываний, т.е. сначала вводится понятие атомарной формулы, а затем формулы. Только с определением атомарной формулы в случае логики первого порядка ситуация несколько сложнее.

Будем исходить из следующих трех множеств: F,R,V. Элементы множества F – символы (или имена) функций, элементы R – символы (или имена) предикатов, элементы множества V – переменные. Будем считать, что каждому символу функции и предиката поставлено в соответствие натуральное число или ноль – местность (т.е. число аргументов) этого символа. Допускаются нульместные символы функций, которые называются *константами*, и нульместные символы предикатов (последние будут играть роль атомарных формул логики высказываний). Объединение F и R будем называть *сигнатурой*. Сигнатуру заранеее фиксировать не будем, она будет определяться содержанием решаемой задачи. Множество V предполагается

бесконечным, для обозначения его элементов будут использоваться, как правило, буквы x,y,z,u,y,w с индексами и без них.

В примерах, приведенных выше, мы видели, что для записи предикатов использовались арифметические выражения: 2x, x+y, – x^2. Аналогом арифметического выражения в логике служит понятие терма.

Определение. *Термом* называется

1) переменная и константа;

2) выражение вида $f(t_1,\ldots,t_n)$, где $t_1,\ldots,t_n$ – термы, а f – n-местный функциональный символ.

Можно сказать, что терм – выражение, полученное из переменных и констант с помощью символов функций.

Определение. *Атомарной формулой* называется выражение вида $r(t_1,\ldots,t_n)$, где $t_1,\ldots,t_n$ – термы, r – символ n-местного предиката.

Примерами атомарных формул являются выражения x£y^2+1, |x-y|<z, x делит нацело y-3.

Определение. *Формулой логики первого порядка* называется

1) атомарная формула,

2) выражения вида (F)&(G), (F_1)Ú(G), Ø(F), (F)®(G),(F)«(G), ("v)(F), ($v)(F), где F и G – формулы логики предикатов, v – переменная.

Формула F в двух последних выражениях называется *областью действия квантора по переменной n.*

К соглашениям о приоритете операций, принятом в логике высказываний, добавим следующее: кванторы имеют одинаковый приоритет, который больше приоритета всех связок.

Определение. Вхождение переменной в формулу называется *связанным*, если переменная стоит непосредственно за квантором или входит в область действия квантора по этой переменной. В противном случае вхождение называется *свободным*.

Например, в формуле

F=t(x)&("y)[s(x,y)®($x)(r(x,y)Út(y))]

первое и второе вхождение переменной x свободны, третье и четвертое связаны. Все вхождения переменной y связаны.

Определение. Переменная называется *свободной в формуле,* если существует хотя бы одно свободное вхождение переменной в формулу.

Формула F из предыдущего примера имеет одну свободную переменную x.

Если $x_1,\ldots,x_n$ – все свободные переменные формулы F, то эту формулу будем обозначать через $F(x_1,\ldots,x_n)$. Это обозначение будем применять и в том случае, когда не все переменные из $x_1,\ldots,x_n$ являются свободными в F.

Как и в логике высказываний. В логике первого порядка вводится понятие подформулы. Соответствующее определение получится из определения подформулы из §9.2 добавлением фразы: «Подформулами формул ("v)(F) и ($v)(F) являются они сами и все подформулы формулы F».

10.3. Интерпретация в логике первого порядка

Необходимо соотнести формулы логики предикатов первого порядка и предикаты. Как и в логике высказываний подобное соотнесение осуществляет функция, называемая интерпретацией.

Определение. *Интерпретацией* на непустом множестве M называется функция, заданная на сигнатуре FÈR, которая

1) константе ставит в соответствие элемент из M;

2) символу n-местной функции ставит в соответствие некоторую n-местную функцию, определенную на множестве M;

3) символу n-местного предиката ставит в соответствие n-местный предикат, заданный на M.

В результате любая формула F получает в соответствие предикат, местность которого равна числу свободных переменных формулы F.

Приведем примеры. Пусть сигнатура состоит из символа одноместного предиката P и двухместного предиката D, M={2,3,6,9,12,15} и

F=(P(x)&("$_y$)(P(y)®D(x,y))

Поставим в соответствие (проинтерпретируем) P(x) предикат «x – простое число», D(x,y) – предикат «x меньше или равно y». Тогда формула F получит в соответствие предикат «x=2». На этом же множестве можно рассмотреть и другую интерпретацию: P(x) ставится в соответствие «x – нечетное число», D(x,y) – предикат «x делит y». В таком случае, формула F получает в соответствие предикат «x=3». Если j – интерпретация, то предикат, соответствующий формуле F будем обозначать через j(F).

Одним из основных типов задач этой темы являются задачи, связанные с использованием выразительных возможностей языка логики предикатов. В качестве примера рассмотрим задачу перевода на язык логики предикатов следующего рассуждения. «Каждый первокурсник знаком с кем-либо из спортсменов. Никакой первокурсник не знаком ни с одним любителем подледного лова. Следовательно, никто из спортсменов не является любителем подледного лова». Для удобства ссылок это рассуждение условимся называть рассуждением о первокурсниках. Выберем следующую сигнатуру:

П(x): «x – первокурсник»,

С(x): «x – спортсмен»,

Л(x): «x – любитель подледного лова»,

З(x,y): «x знаком с y».

Тогда рассуждение запишется в виде следующей последовательности формул.

H_1=("x)[П(x)®($y)(С(y)&З(x,y))],

H_2=("x)("y)[П(x)&Л(y)®ØЗ(x,y)],

H_3=("x)(С(x)®ØЛ(x))

Мы установили, что выразительных средств логики предикатов достаточно, чтобы записать рассуждение о первокурсниках. Естественно далее поставить вопрос, логично ли оно? Будет ли третье предложение следствием первых двух? На этот вопрос мы ответим в 10.5.

10.4. Равносильность, законы логики первого порядка

Общая схема изложения материала этого и двух следующих параграфов будет напоминать изложение материала в 9.3-9.5.

Определение. Формулы $F(x_1,\ldots,x_n)$ и $G(x_1,\ldots,x_n)$ называются *равносильными*, если для любой интерпретации j с областью M высказывания $(jF)(a_1,\ldots,a_n)$ и $(jG)(a_1,\ldots,a_n)$ при любых $a_1,\ldots,a_n$ из M одновременно истинны или одновременно ложны.

Пусть F(x)=Ø("y)P(x,y), G(x)=($y)ØP(x,y), где P – символ двухместного предиката. Докажем, что формулы F(x) и G(x) равносильны. Возьмем интерпретацию j с областью M. Пусть высказывание (jF)(a) истинно для aÎM. Истинность этого высказывания одначает, что не для всякого yÎM высказывание (jP)(a,y) истинно. Следовательно, найдется bÎM, для которого высказывание (jP)(a,b) ложно. Если высказывание (jP)(a,b), ложно, то высказывание Ø(jP)(a,b) истинно. Отсюда следует, что найдется yÎM такой, для которого высказывание Ø(jP)(a,y) истинно. Это означает истинность высказывания (jG)(a). Итак, мы доказали, что если высказывание (jF)(a) истинно, то высказывание (jG)(a) тоже истинно. Обратная импликация доказывается аналогично. Значения

истинности высказываний (jF)(a) и (jG)(a) при любом aÎM совпадают. Следовательно, формулы F(x) и G(x) равносильны.

Определение. Формула $F(x_1,\ldots,x_n)$ называется *тождественно истинной*, если для любой интерпретации j с областью M высказывание $(jF)(a_1,\ldots,a_n)$ при любых $a_1,\ldots,a_n$ из M является истинным.

Как и в случае логики высказываний. Приведем список основных равносильностей – законов логики предикатов. Прежде всего, получим законы логики предикатов из законов 1–21 логики высказываний, понимая под F,G,H – произвольные формулы логики предикатов. Дополним полученный список законами, специфичными для логики предикатов

22) ("x)(F(x)&G(x) равносильна ("x)(F(x)&("x)G(x),

23) ($x)(F(x)ÚG(x) равносильна ($x)F(x)Ú($x)G(x),

24) ("x)("y)F(x,y) равносильна ("y)("x)F(x,y),

25) ($x)($y)F(x,y) равносильна ($y)($x)F(x,y),

26) Ø("x)F(x) равносильна ($x)ØF(x),

27) Ø($x)F(x) равносильна ("x)ØF(x).

Законы 21,22 утверждают, что квантор общности можно переносить через конъюнкцию, а квантор существования через – дизъюнкцию. Естественно поставить вопрос, можно ли квантор существования переносить через конъюнкцию, а квантор общности – через дизъюнкцию. Другими словами, будут ли равносильны следующие пары формул

("x)(F(x)ÚG(x) и ("x)(F(x)Ú("x)G(x)

($x)(F(x)&G(x) и ($x)(F(x)& ($x)G(x) ?

Оказывается нет. Докажем это для случая, когда F(x) и G(x) – атомарные формулы. Пусть основное множество – множество натуральных чисел N, F(x) – предикат «x – четное число» , G(x) – предикат «x – нечетное число». Обозначим эту интерпретацию буквой j. Тогда j("x)(F(x)ÚG(x))]=1, но j[("x)F(x)Ú("x)G(x)=0. Аналогично, j($x)(F(x)&G(x))]=0 и j($x)F(x)&($x)G(x)]=1.

Рассмотрим законы 23 и 24. Они утверждают, что одноименные кванторы можно менять местами. Можно ли переставлять местами разноименные кванторы, сохраняя равносильность. Другими словами, равносильны ли формулы

("x)($y)(F(x,y) и ($y)("x)F(x,y)?

Оказывается, тоже нет. В качестве основного множеств авозьмнм опять множнство натуральных чисел, F(x,y) будем считать атомарной формулой и поставим ей в соответствие предикат «x меньше y». Тогда левая формула будет истинной, правая – ложной.

Вернемся к законам 21 и 22. Мы отмечали, что квантор существования – через конъюнкцию. Тем не менее, если одна из формул F или G не содержит переменной x, то это делать можно. Запишем соответствующие законы:

28) ("x)(F(x)ÚG) равносильна ("x)(F(x)ÚG,

29) ($x)(F(x)&G) равносильна ($x)(F(x)&G, где G не содержит x.

Законы 21, 22, 28, 29 можно записать в общем виде:

30) $(Q_1x)(Q_2z)(F(x)ÚG(z))$ равносильна $(Q_1x)F(x)Ú(Q_2z)G(z)$

31) $(Q_1x)(Q_2u)(F(x)\&G(u))$ равносильна $(Q_1x)F(x)\&(Q_2u)G(u)$,

где Q_1, Q_2 кванторы V или Э, u не входит в F(x), а x не входит в G(u). Для доказательства равносильности двух формул могут оказаться полезными следующие законы:

Рассмотрим формулу F(x)=("y)P(x,y)®P(x,c), где P – символ двухместного отношения, c – константа. Докажем, что формула F(x) тоэждественно истинна. Возьмем интерпретацию j с областью M и элемент a из M. Высказывание (jF)(a) равно ("y)(jP)(a,y)®(jP)(a,j(c)). Если посылка ("y)(jP)(a,y) ложна, то вся импликация (jF)(a) истинна. Предположим, что посылка ("y)(jP)(a,y) истинна. Это означает, что для всякого yÎM высказывание (jP)(a,y) истинно, в том числе истинно это высказывание и для y=j(c). Следовательно, истинно заключение (jP)(a,j(c)) и вся импликация (jF)(a). Мы доказали, что высказывание (jF)(a) истинно для любого aÎM. Это означает, что формула F(x) является тождественно истинной.

Понятия равносильности и тождественной истинности в логике первого порядка связаны точно так же, как и в логике высказываний.

Теорема 3.1 Формулы $F(x_1,\ldots,x_n)$ и $G(x_1,\ldots,x_n)$ равносильны тогда и только тогда, когда формулы $F(x_1,\ldots,x_n)$«$G(x_1,\ldots,x_n)$ тождественно истинны.

Доказательство теоремы 3.1 аналогично доказательству теоремы 1.1 и поэтому не приводится

32) ("x)F(x) равносильна ("z)F(z),

33) ($x)F(x) равносильна ($z)F(z).

В законах 32 и 33 переменная z не входит в F(x), а переменная x не входит в F(z).

В логике высказываний мы применяли два способа доказательства равносильности формул: построение совместной таблицы истинности и переход от одной формулы к другой с помощью законов. В случае логики первого порядка остается только второй способ.

Проиллюстрируем его на примере следующей задачи: доказать равносильность формул:

F=Ø("x)($y)[S(x)&P(x,y)®($z)(T(z)&P(x,z))]

G=($x)("y)[S(x)&P(x,y)&("z)(T(z)®ØP(x,z))].

Применим к формуле F последовательно законы 26, 27 и 20, получим, что формула F равносильна формуле

F_1=($x)("y)Ø[Ø(S(x)&P(x,y))Ú($z)(T(z)&P(x,z))].

Далее, используя закрны 18,19 и 27 из F_1, получим формулу

F_2=($x)("y)[S(x)&P(x,y)&("z)Ø(T(z)&P(x,z))].

Осталось заметить, что в силу законов 17 и 20 в формуле F_2 подформулу Ø(T(z)&P(x,z)) можно заменить на T(z)®ØP(x,z).

Подчеркнем, что доказательство равносильности двух формул будем проводить с помощью законов логики первого порядка. Доказательство того, что формулы неравносильны, будем осуществлять построением интерпретации, при которой одна формула истинна, другая ложна. Например, так, как это было сделано выше для доказательства неравносильности формул ("x)(F(x)ÚG(x)) и ("x)F(x)Ú("x)G(x). Разумеется, до построения интерпретации формулы можно предварительно преобразовывать с помощью законов.

10.5. Логическое следствие

Определение. Формула $G(x_1,\ldots,x_n)$ называется логическим следствием формул $F_1(x_1,\ldots,x_n),\ldots,F_k(x_1,\ldots,x_n)$, если для любой интерпретации j с областью M и любых $a_1,\ldots,a_n$ÎM из истинности высказываний $(jF_1)(a_1,\ldots,a_n),\ldots,(jF_k)(a_1,\ldots,a_n)$ следует истинность высказывания $(jG)(a_1,\ldots,a_n)$.

Приведем примеры. Пусть F_1=("x)(P(x)®Q(x)&R(x)), F_2=P(c), G=Q(c). Покажем, что формула G является логическим следжствием формул F_1 и F_2. Возьмем интерпретацию j с областью M такую, что высказывания jF_1 и jF_2 истинны. Элемент j(c) обозначим буквой b. Истинность jF_2 означает, что высказывание (jP)(b) истинно. А истинность высказывания jF_1 означает, что для любого элемента xÎM истинно высказывание (jP)(x)®(jQ)(x)&(jR)(x). Поскольку это высказывание истинно для любого x, то, в частности, истинно для x=b. Мы видим,

что истинна импликация (jP)(b)®(jQ)(b)&(jR)(b) и ее посылка (jP)(b). Из таблицы истинности импликации следует истинность заключения (jQ)(b)&(jR)(b). Следовательно, истинно высказывание (jQ)(b). А это и есть jG. Мы доказали, что если истинны высказывания jF_1 и jF_2, то истинно высказывание jG, т.е. что G –логическое следствие F_1 и F_2.

В качестве второго примера докажем нелогичность рассуждения о первокурсниках. Мы записали это рассуждение в виде последовательности формул H_1,H_2, и H_3. Для доказательства нелогичности рассуждения надо найти интерпретацию j, при которой формулы H_1 и H_2 истинны, а формула H_3 ложна. Пусть множество M состоит из трех элементов 2,3,4. Символы С, Л и П проинтерпретируем следующим образом:

(jC)(x)=«x – простое число»,

(jЛ)(x)=«x – четное число»,

(jП)(x)=’’x>4»,

т.е. П интерпретируется как тождественно ложный предикат. Тогда формулы H_1 и H_2 истинны, поскольку посылки импликаций этих формул ложны при любом x. А формула H_3 ложна. Чтобы убедиться в этом достаточно взять x=2. Следовательно, рассуждение о первокурсниках нелогично.

Определение. Множество формул

$K=\{F_1(x_1,\ldots,x_l),\ldots,F_m(x_1,\ldots,x_l)\}$

называется *выполнимым*, если существует интерпретация j с областью M и элементы $a_1,\ldots,a_l \hat{I} M$ такие, что все высказывания $(jF_1)(a_1,\ldots,a_l),\ldots,(jF_m)(a_1,\ldots,a_l)$ истинны.

Множество формул K = { F_1=("x)($y)(P(y)&Q(x,y)), F_2=("y)Q(c,y), F_3=ØP(c) } выполнимо. Возьмем в качестве области интерпретации множество натуральных чисел **N.** Символы P,Q и C проинтерпретируем следующим образом:

(jP)(u)=«u – простое число»,

(jQ)(u,v)=«u меньше или равно v»,

(j(C)=1.

Тогда высказывание jF_1 означает, что для любого натурального числа x найдется простое число y, которое не меньше x, высказывание jF_2 означает, что 1 –наименьшее натуральное число, а высказывание jF_3 означает, что 1 – непростое число. Ясно, что все эти высказывания истинны, и поэтому множество формул K выполнимо.

Понятия логического следствия и выполнимости в логике первого порядка связаны точно так же, как и в логике высказываний.

Теорема 3.2. Формула $G(x_1,\ldots,x_n)$ является логическим следствием формул $F_1(x_1,\ldots,x_n),\ldots,F_k(x_1,\ldots,x_n)$ тогда и только тогда, когда множество формул $\{F_1(x_1,\ldots,x_n),\ldots,F_k(x_1,\ldots,x_n),ØG(x_1,\ldots,x_n)\}$ невыполнимо.

Доказательство теоремы 3.2 аналогично доказательству теоремы 1.2 и поэтому не приводится.

10.6. Нормальные формы

Как и в логике высказываний, в логике первого порядка вводятся нормальные формы. Мы рассмотрим две из них: предваренную нормальную и сколемовскую нормальную формы.

Определение. Формула G имеет *предваренную нормальную форму* (сокращенно: ПНФ), если

$G(Q_1,x_1)\ldots(Q_n,x_n)H,$

где $Q_1,\ldots,Q_n$ кванторы, а формула H не содержит кванторов.

Например, формула ("x)($y)(P(x,y)&ØQ(y)) имеет предваренную нормальную форму, а формула Ø("x)(T(x)&S(x,y)) не имеет.

Теорема 3.3. Для всякой формулы F существует формула G, равносильная F и имеющая предваренную нормальную форму.

Доказательство теоремы легко следует из анализа алгоритма приведения к ПНФ.

Алгоритм приведения к предваренной нормальной форме

Шаг 1. Используя законы 21 и 20, исключить эквиваленцию и импликацию.

Шаг 2. Занести отрицание к атомарным формулам, пользуясь законами 17-19 и 26-27.

Шаг 3. С помощью законов 22-23, 28-31 вынести кванторы вперед, используя при необходимости переименование связанных переменных (законы 32-33).

Рассмотрим пример. Пусть

F=("x)P(x)®($y)("z)(P(y)&Q(y,z)).

Выполнив шаг 1 (с помощью закона 20), получим формулу

F_1=Ø("x)P(x)Ú($y)("z)(P(y)&Q(y,z)).

С помощью закона 26 перейдем к формуле

F_2=($x)ØP(x)Ú($y)("z)(P(y)&Q(y,z)).

Тем самым, шаг 2 также выполнен. Применим закон 30 слева направо Q_1=$, Q_2=$, u=y, получим формулу

F_3=($x)($y)[ØP(x)Ú("z)(P(y)ÚQ(y,z))].

(Пользуемся тем, что ØP(x) не содержит y, а ("z)(P(y)&Q(Y,z)) не содержит x. Так как формула ØP(x) не содержит z, то применение закона 26 слева направо дает формулу

F_4=($x)($y)("z)[ØP(x)ÚÚ(P(y)&Q(y,z))].

Это и есть искомая формула, имеющая ПНФ и равносильная формуле F.

В предыдущем примере выполнение шага 3 можно организовать по-другому. В формуле F_2 связанную переменную y заменим на переменную x (закон 33), получим формулу

F_3'=($x)ØP(x)Ú($x)("z)(P(x)&Q(x,z)).

Используя закон 23, перейдем к формуле

F_4'=($x)[ØP(x)Ú("z)(P(x)&Q(x,z))].

Затем, как и в предыдущем абзаце, с помощью закона 28 вынесем квантор по z за квадратную скобку. Получим формулу

F_5'=($z)("z)[ØP(x)Ú(P(x)&Q(x,z)].

Формула F_5', как и формула F_4, имеет предваренную нормальную форму и равносильна формуле F. В некоторых ситуациях формула F_5' предпочтительнее формулы F_4, поскольку содержит меньше кванторов. (Кстати, бескванторную часть формулы F_5' можно упростить).

Перейдем к изучению сколемовской нормальной формы. Отметим вначале, что в логике первого порядка понятие конъюнктивной нормальной формы вводится точно так же, как и в логике высказываний. Сохраняется полностью алгоритм приведения к КНФ и утверждение теоремы 1.3.

Определение. Формула G имеет *сколемовскую нормальную форму* (сокращенно: СНФ), если

G=("x)…("x_n)H,

где формула H не содержит кванторов и имеет конъюнктивную нормальную форму.

Например, формула ("x)[P(x)Ú(P(y)&Q(x,y)] имеет сколемовскую нормальную форму, а формулы ("x)($y)Q(x,y), ("x)[P(x)&(P(y)ÚQ(x,y)] не имеют.

В отличие от предыдущего случая предваренной нормальной формы мы здесь вначале рассмотрим алгоритм приведения к СНФ, а затем сформулируем теорему.

Алгоритм приведения к сколемовской нормальной форме

Шаг 1 – Шаг 3 – те же, что и в предыдущем алгоритме.

Шаг 4. Бескванторную часть привести к конъюнктивной нормальной форме (алгоритм описан в §5 темы 1).

Шаг 5. Исключить кванторы существования. Этот шаг изложим на примере. Пусть после выполнения четвертого шага имеем формулу

F=($x)("y)($z)("u)($v)H(x,y,z,u,v),

где H – не содержит кванторов. Предположим, что формула не содержит константы c, символов одноместной функции f и двухместной функции g. Тогда в формуле H заменим x на c, z – на f(y), v заменим на g(y,u). Все кванторы существования вычеркнем. Получим формулу

G=("y)("u)H(c,y,f(y),u,g(g,u)).

Это и есть результат выполнения шага 5.

Приведем пример приведения к СНФ. Пусть

F=($x)("y)[P(x,y)®($z)(Q(x,z)&R(y))].

Применяя законы 20 и 23, получаем формулу

F_1=($x)("y))($z)[ØP(x,y)Ú(Q(x,z)&R(y))].

Она имеет предваренную нормальную форму. Используя закон 12 приводим бескванторную часть к КНФ:

F_2=($x)("y))($z)[ØP(x,y)Ú(Q(x,z)&(ØP(x,y)Ú(R(y))].

Сделаем подстановку x=a, z=f(y), получим искомую формулу

G=("y)[(ØP(a,y)ÚQ(a, f(y)))&(ØP(a,y))ÚR(y))].

Теорема 3.4. Для всякой формулы F существует формула G, имеющая сколемовскую нормальную форму и одновременно выполнимая (или невыполнимая) с F.

Доказательство. Пусть G – результат работы алгоритма приведения к СНФ. То, что результатом работы алгоритма является формула в сколемовской нормальной форме, ясно из описания алгоритма. Формула, которая получается после выполнения шагов 1-4 равносильна исходной, и в частности, одновременно выполнима или не выполнима.

Проанализируем шаг 5. Предположим вначале, что исключается квантор существования, впереди которого нет кванторов общности. Можно считать, что это первый квантор в записи формулы; т.е. E(u_1,…,u_n)=($y)E′(y,$u_1$,…,$u_n$). (Формула E′ может содержать кванторы.) Рассмотрим интерпретацию j с областью M, при которой формула E выполнима. Выполнимость означает, что найдутся элементы a_1,…,a_nÎM такие, что высказывание (jE)(a_1,…,a_n) или (что тоже самое) высказывание ($y)(jE′)(y,$a_1$,…,$a_n$) истинно. Отсюда следует, что существует элемент bÎM такой что высказывание (jE′)(b,a_1,…,a_n) также истинно. Исключение квантора существования по y на пятом шаге приводит к формуле D=E′(c,u_1,…,u_n), где c – константа, отсутствующая

в E′. Расмотрим интерпретацию y, которая совпадает с j на всех символах предикатов и функций, входящих в запись формулы F, и y(c)=b. Тогда $(yD)(a_1,\ldots,a_n)=(jE')(b,a_1,\ldots,a_n)$. Мы доказали, что если формула E выполнима, то выполнима и формула D.

Предположим теперь, что выполнима формула $D(u_1,\ldots,u_n)=E'(c,u_1,\ldots,u_n)$. Это означает, что существует интерпретация y с областью M и элементы $a_1,\ldots,a_n\hat{I}M$ такие, что высказывание $(yE')(y(c),a_1,\ldots,a_n)$ истинно. Но отсюда следует истинность высказывания $(\$y)(yE')(y,a_1,\ldots,a_n)$. Следовательно, формула $E(u_1,\ldots,u_n)$ выполнима. Мы доказали, что из выполнимости формулы D следует выполнимость формулы E.

Рассмотрим теперь случай, когда исключается квантор существования, впереди которого есть k кванторов общности, т.е.

$$E(u_1,\ldots,u_n)=(\forall x_1)\ldots(\forall x_k)(\$y)E'(x_1,\ldots,x_k,y,u_{1,\ldots},u_n).$$

(Формула $E^/$ может содержать кванторы). Исключение квантора по y на шаге 5 приведет к формуле

$$D(u_1,\ldots,u_n)>(\forall x_1)\ldots(\forall x_k)E(x_1,\ldots,x_k,f(x_1,\ldots,x_k),u_1,\ldots,u_n),$$

где f – символ k–местной функции, не содержащийся в E. Предположим, что формула E выполнима. Выполнимость означает существование интерпретации j областью M и элементов $a_1,\ldots,a_n\hat{I}M$ таких, что высказывание $(jE(a_1,\ldots,a_n)$ истинно. Истинность этого высказывания означает, что для любых элементов $x_1,\ldots,x_k\hat{I}M$ найдется элемент $y\hat{I}M$ такой, что высказывание $E^/(x_1,\ldots,x_k,y,a_1,\ldots,a_n)$ истинно. Если для данных элементов $x_1,\ldots,x_k$ таких элементов y несколько, то зафиксируем один. Тем самым мы определили на M функцию $i:Mx\ldots xM\to M$ такую, что высказывание

$$(jE')(x_1,\ldots,x_k,i(x_1,\ldots,x_k),a_1,\ldots,a_n$$

истинно для всех $x_1,\ldots,x_k\hat{I}M$. Рассмотрим интерпретацию y, которая совпадает с j на всех символах функций и предикатов, входящих в запись формулы E, и $(yf)(x_1,\ldots,x_n)=i(x_1,\ldots,x_n)$. Тогда

$$(yD)(a_1,\ldots,a_n)=(\forall x_1)\ldots(\forall x_k)(jE^/)(x_1,\ldots,x_k,i(x_1,\ldots,x_k),a_1,\ldots,a_n),$$

последнее высказывание, как мы видели истинно. Следовательно, формула $D(u_1,\ldots,u_n)$ выполнима. Мы показали, что из выполнимости формулы E следует выполнимость формулы D.

Пусть выполнима формула D. Это означает, что существует интерпретация y с областью M и элементы $a_1,\ldots,a_n\hat{I}M$ такие, что высказывание $(yD)(a_1,\ldots,a_n)$ или (что то же самое) высказывание $(\forall x_1)\ldots(\forall x_k)(yE^/)(x_1,\ldots,x_k,(yf)(x_1,\ldots,x_k),a_1,\ldots,a_n)$ истинно. Отсюда следует, что для любых $x_1,\ldots,x_k$ найдется y (равный $(jf)(x_1,\ldots,x_k)$) такой, что высказывание $(yE^/)(x_1,\ldots,x_k,y,a_1,\ldots,a_n)$ истинно. Следовательно, истинно высказывание $(\forall x_1)\ldots(\forall x_k)(\$y)(yE^/)(x_1,\ldots,x_k,y,a_1,\ldots,a_n)$, т.е. высказывание $(yE)(a_1,\ldots,a_n)$. Мы доказали, что из выполнимости формулы D следует выполнимость формулы E.

Теорема доказана.

10.7. Невыразимость в логике первого порядка

В примерах, рассмотренных в предыдущих параграфах, мы видели, что логика первого порядка обладает значительными выразительными возможностями. Данный параграф посвящен доказательству того, что выразительные возможности этой логики ограничены.

Рассмотрим предварительно понятие транзитивного замыкания двухместного предиката.

Определение. Пусть S(x,y) – двухместный предикат, заданный на множестве M. Предикат $S^+(x,y)$ называется *транзитивным замыканием* предиката S(x,y), если для любых $a,b\hat{I}M$ выполняется условие:

высказывание $S^+(a,b)$ истинно $\hat{U}$ существует натуральное число n и элементы $c_0,\ldots,c_n$ множества M такие, что $a=c_0,\ldots, c_n=b$ и все высказывания $S(c_0,c_1), S(c_1,c_2),\ldots, S(c_{n-1},c_n)$ истинны.

Если предикат представлять отношением, то транзитивное замыкание предиката – транзитивное замыкание бинарного отношения.

Рассмотрим пример. Пусть M – множество натуральных чисел, а S(x,y) – предикат «x непосредственно предшествует y», т.е. «y=x+1». Тогда транзитивным замыканием предиката S(x,y) будет предикат «x меньше y».

Оказывается, что не существует формулы логики первого порядка, которая выражала бы транзитивное замыкание двухместного предиката. Более точная формулировка содержится в следующем утверждении.

Теорема 3.5 Пусть P – символ двухместного предиката. Не существует формулы F(x,y) логики первого порядка такой, что для любой интерпретации j с областью M предикат (jF)(x,y) есть транзитивное замыкание предиката (jP)(x,y).

Доказательство теоремы 3.5 опирается на одно известное в логике первого порядка утверждение, которое называется теоремой компактности. В формулировке теоремы компактности используется понятие логического следствия бесконечного множества формул. Определение этого понятия легко получается из определения логического следствия множества формул $F_1,\ldots,F_k$ из §6 и поэтому не приводится.

Теорема компактности. Если формула G является логическим следствием бесконечного множества формул K, то G является логическим следствием некоторого конечного подмножества K′ множества K.

Доказательство теоремы компактности является довольно сложным, поэтому здесь не излагается.

Приведем **доказательство** теремы 3.5. Предположим противное, т.е. предположим, что существует формула F(x,y) логики первого порядка такая, что для любой интерпретации j с областью M и любых a,bÎM выполняется эквиваленция

$(jF)(a,b)\hat{U}(jP)^{+}(a,b)$.

Рассмотрим следующее множество формул

$K=\{E_0(x,y), E_1(x,y),\ldots, E_n(x,y),\ldots\}$:

$E_0=(x,y)=ØP(x,y)$

$E_1(x,y)=Ø(\$z_1)[P(x,z_1)\&P(z_1,y)]$,

$E_2(x,y)=Ø(\$z_1)(\$z_2)[P(x,z_1)\&P(z,z_2)\&P(z_2,y)]$,

$E_n(x,y)=Ø(\$z_1)\ldots(\$z_n)[p(x,z_1)\&P(x,z_2)\&\ldots\&P(z_n,y)]$,

. . .

Используя определение транзитивного замыкания и предположение о том, что формула F(x,y) определяет транзитивное замыкание, получаем, что формула ØF(x,y) есть логическое следствие множества формул K. По тереме компактности ØF(x,y) есть логическое следствие некоторого конечного подмножества K′ множества K. Можно считать, что

$K'=\{E_0(x,y), E_1(x,y),\ldots, E_d(x,y)\}$

для некоторого d.

Пусть M={0,1,…,d+1,d+2}. На множестве M определим двухместный предикат S следующим образом:

S(u,v)Ûu+1 равно v

Например, высказывания S(0,1), S(1,2) истинны, а высказывание S(0,2) ложно. Отметим, что высказывание $S^{+}(0,d+2)$ истинно. Рассмотрим интерпретацию j, для которой (jP)(u,v)=S(u,v). Все высказывания $(jE_0)(0,d+2)$, $(jE_1)(0,d+2),\ldots(jE_d)(0, d+2)$ истинны. Так как формула ØF(x,y) есть логическое следствие множества формул K′, то истинно высказывание Ø(jF)(0, d+2). С другой стороны, поскольку F(x,y) определяет транзитивное замыкание и истинно высказывание $S^{+}(0, d+2)$ (другими словами, высказывание $(jP^{+})(0, d+2)$), то истинно высказывание (jF)(0, d+2). Мы доказали, что истинно и последнее высказывание, и его отрицание. Полученное противоречие показывает,

что не существует формулы логики первого порядка, определяющей транзитивное замыкание предиката.

Теорема доказана.

10.8. Многосортная логика первого порядка

Расширим понятия формулы, введя так называемые ограниченные кванторы. Допустим, что нам надо записать на языке логики предикатов следующее утверждение «для всякого x>5 существует y>0 такое, что xy=1». Отметим, что здесь написано не «для всякого x» и «существует y», а «для всякого x>5 « и «существует y>0 «. Если на это не обратить внимание, то получается формула ("x)($y)(xy=1) имеющая другой смысл, нежели исходное утверждение. Для правильного перевода надо немного изменить исходное предложение по форме (не меняя, разумеется, смысла): «для всякого x справедливо, если x>5, то существует y такой, что y>0 и xy=1». Правильный перевод имеет вид ("x)[x>5®($y)(y>0&(xy=1)].

Если рассматривать более длинные исходные предложения, то соответствующие им формулы логики предикатов будут, вообще говоря, довольно громоздкими. Для того, чтобы частично избавиться от усложнения при переводе на язык логики предикатов, вводятся ограниченные кванторы. Пусть B(x) – формула с одной свободной переменной x. Тогда выражение "B(x) называется *ограниченным квантором общности* $B(x) – *ограниченным квантором существования.* С помощью ограниченных кванторов исходное предложение предыдущего абзаца можно записать довольно просто: ("x>5)($y>0)(xy=1).

Более формально, ограниченные кванторы вводятся следующим образом: формула ("B(x)F(x) есть сокращение формулы ("x)B(x)®F(x)), формула ($B(x)F(x) – сокращение формулы ($B(x)&F(x)). Для ограниченных кванторов справедливы аналоги законов 22-33.

33. Упомянутые выше четыре типа задач могут быть поставлены и для логики с ограниченными кванторами.

Ограниченные кванторы часто вводятся неявно. Для переменных, пробегающих множество истинности формулы B(x), вводят

специальное обозначение. Например, в геометрии довольно часто применяется следующее соглашение: A, B, C, D,… обозначаются точки, буквами a, b, c, d,… прямые, а буквами a, b, g,… – плоскости, т.е. с нашей точки зрения первые пробегают множество истинности формулы T(x), вторые – Пр(x), третьи – Пл(x).

Последовательное оформление этой идеи приводит к понятию многосортной (многоосновной) логики предикатов. Строгих определений давать не будем, укажем только отличие от обсуждавшейся ранее (одноосновной или односортной) логики предикатов. Построение формулы также исходит из множеств F, R, V. Только в этом случае переменные разбиты по сортам. В примере с геометрией таких сортов три: переменные, принимающие в качестве значений точки, прямые и плоскости. Далее, для каждого символа из F указано, какой сорт имеет первый аргумент, какой – второй и т.д., какой сорт имеет значение функции. Аналогичная информация имеется и для каждого символа предиката. Для интепретации берется не одно множество, а столько, сколько сортов переменных (эти множества называются основами). Для геометрии таких основ три: множество точек, множество прямых, множество плоскостей.

Приведем пример применения многоосновной логики предикатов.

Рассмотрим информационную систему под условным названием «Сделки». Система содержит сведения о сделках купли-продажи, произведенных некоторой фирмой. Предметом сделок служат партии товаров, определяемые номером партии, наименованием товара, единицей измерения и количеством. Используются следующие атрибуты: НОМ – номер партии товара, НАИМ – наименование товара, ЕД – единица измерения, КОЛ – количество единиц товара в партии, ДАТА – дата сделки, АГЕНТ – покупатель или продавец, СЕК – номер секции склада, СРОК – срок годности. Информация хранится в виде отношений: ПАР (НОМ, НАИМ, ЕД, КОЛ), ПОК (НОМ, ДАТА, АГЕНТ), ПРОД (НОМ, ДАТА, АГЕНТ), СКЛАД (СЕК, НОМ, СРОК). Первое отношение содержит сведения о партиях товара, которые были предметом сделок, второе – сведения о покупках, третье – о продажах партий товара. В четвертом указывается, в какой секции склада хранится купленная (но еще не проданная) партия товара и срок годности товара в партии. Система может вычислять отношение РАН(x,y)=«x раньше y», определенное на доменах атрибутов ДАТА и СРОК.

Формализуем эту информационную систему в многоосновной логике предикатов следующим образом. Введем восемь сортов переменных (по количеству атрибутов), для каждого атрибута – свой сорт. Переменные будут принимать значения в доменах соответствующих атрибутов. Другими словами, области интерпретации (основы) будут состоять из доменов атрибутов. Переменные по сортам синтаксически различать не будем. А для того, чтобы указать, что переменная x, например, изменяется по домену атрибута НАИМ, а y – по домену атрибута ЕД, будем писать: xÎНАИМ, yÎЕД. Каждому отношению поставим в соответствие предикат той же местности, что и отношения, с соответствующими типами переменных. Предикат и отношение будем обозначать одинаково. Например, отношению ПАР(НОМ, НАИМ, ЕД, КОЛ) будет соответствовать предикат ПАР(x, y, u, v), где xÎНОМ, yÎНАИМ, uÎЕД, vÎКОЛ. Сигнатура å, таким образом, будет содержать 5 символов предикатов:

å={ПАР(x, y, u, v), ПОК(x, y, z), ПРОД(x, y, z), СКЛАД(x, y, z), РАН(x,y)}.

В сигнатуру å можно добавлять константы, интепретируемые как элементы доменов атрибутов.

Эта формализация позволяет запросы к информационной системе представлять формулами логики предикатов указанной сигнатуры. Расмотрим следующий запрос:

Q_1. «Каковы номера партий товаров, купленных у фирмы b, и каково наименование товара в этих партиях?»

Запрашиваемая информация содержится в двух отношениях ПАР(НОМ, НАИМ, ЕД, КОЛ) и ПОК(НОМ, ДАТА, АГЕНТ), которые связаны номером партии товара, АГЕНТ – b. Если взять конъюнкцию предикатов

ПАР(x, y, u, v)&ПОК(x, z, фирма b),

то эта формула будет задавать пятиместный предикат, в котором, кроме запрашиваемой, будет содердаться информация о единицах, количестве единиц и дате сделок. Судя по запросу, эта дополнительная

информация пользователя не интересует, поэтому на соответствующие переменные навесим кванторы существования. Получим формулу:

$F_1(x, y)$=xÎНОМ&yÎНАИМ&($u ÎЕД)($zÎДАТА)[ПАР(x, y ,u, v)&ПОК(x, z, фирма b)]

Формула $F_1(x,y)$ представляет собой перевод запроса Q_1 на язык многоосновной логики предикатов.

Рассмотрим еще ряд примеров перевода запросов на язык логики предикатов.

Q_2. «Каковы наименования товаров, единицы измерения и количество единиц в партиях товара, срок годности, которого истекает 20.03.01?»

Q_3. «Для каких фирм срок годности товара, купленного у этих фирм, истекает 20.03.01?»

Q_4. «Какой товар хранится на складе более, чем в двух партиях?»

Q_5. «Какие из закупленных партий товаров в последствии проданы?»

Эти на язык логики предикатов будут переведены формулами F_2 – F_5:

$F_2(x,y,z)$=xÎНАИМ&yÎЕД&zÎКОЛ&($uÎСЕК)($vÎНОМ)($wÎСРОК)

[ПАР(v,x,y,z)&СКЛАД(u,v,w)&РАН(w,20.03.01)],

$F_3(x)$=xÎАГЕНТ&("yÎНОМ)("zÎДАТА)("uÎСЕК)

("vÎСРОК)[ПОК(y,z,x)&СКЛАД(u,y,v)®РАН(v,20.03.01)],

$F_4(x)$=x xÎНАИМ&($$y_1,y_2$$ÎНОМ)($$z_1,z_2$$ÎЕД)($$u_1,u_2$$ÎКОЛ)($$v_1,v_2$$ÎСЕК)

($$w_1,w_2$$ÎСРОК)[$$y_1 \neq y_2$$&ПАР($$y_1$$,x,$$z_1$$,$$u_1$$)&СКЛАД($$v_1$$,x,$$w_1$$)&

ПАР($$y_2$$,x,$$z_2$$,$$u_2$$)&СКЛАД($$v_2$$,x,$$w_2$$),

$F_5(x)$=xÎНОМ&($yÎНАИМ)($zÎЕД)($uÎКОЛ)($v_1,v_2$ÎДАТА($w_1,w_2$ÎАГЕНТ)

ПАР(x,y,z,u)&ПОК(x,v_1,w_1)&ПРОД(x,v_2,w_2) &РАН(v_1,v_2)].

Рассмотрим второй вариант выбора сигнатуры для формализации запросов $Q_1 – Q_5$. Для этого вначале к имеющимся восьми основам (доменам восьми исходных атрибутов) добавим девятую основу: множество сделок СДЕЛ. Далее, вместо предикатов ПАР(x,y,u,v), ПОК(x,y,z), ПРОД(x,y,z), СКЛАД(x,y,z) введем функции:

наим : НОМ®НАИМ,

ед : НОМ®ЕД,

кол : НОМ®КОЛ,

сдел : СДЕЛ®НОМ,

тип : СДЕЛ®{пок, прод},

дата : СДЕЛ®ДАТА,

агент : СДЕЛ®АГЕНТ,

сек : НОМ®СЕК,

срок : НОМ®СРОК,

а предикат РАН(x,y) оставим. Функции имеют естественный смысл: Например, функция *наим* номеру партиии товара ставит в соответствие наименование товара в этой партии, функция *сдел* ставит в соответствие сделке номер партии товара, относительно которого эта сделка была заключена. Новую сигнатуру будем обозначать буквой D. К сигнатуре D, как и к å, можно добавлять константы. Тогда запросы $Q_1 – Q_5$ будут переведены следующим образом:

$G_4(x,y)$=xÎНОМ&yÎНАИМ&y=*наим*(x)&(($zÎСДЕЛ)

[x=*сдел*(z)*тип*(z)=пок&*агент*(z)=фирма b,

$G_2(x,y,z)$=xÎНАИМ&yÎЕД&zÎКОЛ&

($uÎНОМ)[x=*наим*(u)&y=*ед*(u)&z=*кол*(u)&РАН(*срок*(u),20.03.01)],

$G_4(x)$=xÎНАИМ&(u_1,u_2ÎНОМ)[v_1,v_2ÎСЕК)[$u_1{}^1u_2$&*сек*(u_1)=v_1&*сек*(u_2)

=v_2],

$G_5(x)$=xÎНОМ&(v_1,u_2ÎСДЕЛ)[x=*сдел*(u_1)&x=*сдел*(u_2)&*тип*(u_1)=

пок&*тип*(u_2)=прод&РАН(*дата*(u_1), *дата*(u_2))].

Сигнатуру å можно назвать «реляционной» (или «предикатной»), а D – «функциональной». Разумеется возможны и другие варианты выбора сигнатуры.

11. Методы резолюций

Раздел посвящен рассмотрению метода доказательства того, что формула G является логическим следствием формул $F_1,F_2,...,F_k$, который называется *методом резолюций*. Отметим, что задача о логическом следствии сводится к задаче о выполнимости. Действительно, формула G есть логическое следствие формул $F_1,F_2,...,F_k$ тогда и только тогда, когда множество формул $\{F_1,F_2,...,F_k,ØG\}$ невыполнимо. Метод резолюций, если говорить более точно, устанавливает невыполнимость. Это первая особенность метода. Вторая особенность метода состоит в том, что он оперирует не с произвольными формулами, а с дизъюнктами (или элементарными дизъюнкциями).

11.1. Метод резолюций в логике высказываний

Рассмотрим вначале логику высказываний. Напомним, что литералом мы называли атомарную формулу или ее отрицание, дизъюнктом –

дизъюнкцию литералов. Дизъюнкт может состоять из одного литерала. *На дизъюнкт мы иногда будем смотреть, как на множество литералов,* т.е. не будем различать дизъюнкты, которые получаются один из другого с помощью коммутативности и ассоциативности дизъюнкции, а также идемпотентности. Последнее означает, например, что дизъюнкты XÚØYÚX и XÚØY равны. Нам понадобится особый дизъюнкт – *пустой*, т.е. дизъюнкт, не содержащий литералов. Его мы будем обозначать "квадратиком" □. Будем считать, что пустой дизъюнкт ложен при любой интерпретации. Это означает, что формула F&□ равносильна □, а формула FÚ□ равносильна F. Пустой дизъюнкт есть фактически то же самое, что и атомарная формула 0, но в контексте метода резолюций принято использовать □.

Определение. Литералы L и ØL называются *противоположными*.

Метод резолюций в логике высказываний основан на правиле резолюций.

Определение. *Правилом резолюций в логике высказываний* называется следующее правило: из дизъюнктов XÚF и ØXÚG выводим дизъюнкт FÚG.

Например, из дизъюнктов ØXÚYÚZ и XÚØY выводим дизъюнкты YÚZÚØY. Обратим внимание на то, что в первых двух дизъюнктах есть еще одна пара противоположных литаралов. Условимся, что можно применять правило резолюций не обязательно к самым левым литералам (поскольку мы не различаем дизъюнкты, отличающиеся порядком записи литералов). Тогда правило резолюций, примененное к Y и ØY, даст ØXÚZÚX. Условимся еще о следующем: в дизъюнктах не писать повторяющиеся литералы и не писать □, если есть другие литералы.

Определение. Пусть S – множество дизъюнктов. *Выводом* из S называется последовательность дизъюнктов

$D,D_2,...,D_n$

такая, что каждый дизъюнкт этой последовательности принадлежит S или следует из предыдущих по правилу резолюций. Дизъюнкт D

выводим из S, если существует вывод из S, последним дизъюнктом которого является D.

Например, если S={ØXÚYÚZ, ØYÚU, X}, то последовательность D_1=ØXÚYÚZ, D_2=ØYÚU, D_3=ØXÚZÚU, D_4=X, D_5=ZÚU – вывод из S. Дизъюнкт ZÚU выводим из S.

Применение метода резолюций основано на следующем утверждении, которое называется теоремой о полноте метода резолюций.

Теорема 4.1. Множество дизъюнктов логики высказываний S невыполнимо тогда и только тогда, когда из S выводим пустой дизъюнкт.

Доказательство. Докажем вначале достаточность.

Отметим, что правило резолюций сохраняет истинность. Это означает, что если j(ØXÚF)=1 и j(XÚG)=1 для некоторой интерпретации j, то j(FÚG)=1. Действительно, пусть j(ØXÚF)=1 и j(XÚG)=1. Тогда если j(F)=1, то и j(FÚG)=1. Если же j(F)=0, то j(ØX)=1, поскольку j(ØXÚF)=1. Но в этом случае j(X)=0 и j(G)=1, так как j(XÚG)=1. Если же j(G)=1, то и j(FÚG)=1.

Пусть из S выводим пустой дизъюнкт. Предположим противное: множество S выполнимо, т.е. существует интерпретация y, при которой все дизъюнкты из S истинны. Выводимость пустого дизъюнкта из S означает, что существует последовательность дизъюнктов $D_1,\ldots,D_n=\square$, каждый дизъюнкт которой принадлежит S или получается из предыдущих по правилу резолюций. Если дизъюнкт D_j из этой последовательности принадлежит S, то по предположению $y(D_j)=1$. Если же он получается из предыдущих по правилу резолюций, то также $y(D_j)=1$, поскольку правило резолюций сохраняет истинность. При i=n получаем, что y(□)=1. Противоречие показывает, что предположение о выполнимости множества S – ложное предположение. Следовательно, S невыполнимо. Достаточность доказана.

Докажем необходимость. Доказательство проведем индукцией по следующему параметру: d(S)=сумма числа вхождений литералов в дизъюнкты из S минус число дизъюнктов.

Пусть множество дизъюнктов S невыполнимо. Если пустой дизъюнкт принадлежит S, то он выводим из S (вывод в этом случае состоит из одного пустого дизъюнкта) и необходимость теоремы доказана. Будем считать в силу этого, что □ÏS. При этом предположении каждый дизъюнкт содержит хотя бы один литерал и поэтому d³1.

База индукции: d(S)=1. Если d(S)=1, то все дизъюнкты состоят из одного литерала. Поскольку множество S невыполнимо, то в нем должна найтись пара противоположных литералов X и ØX. В таком случае пустой дизъюнкт выводим из S, соответствующий вывод содержит три дизъюнкта: X, ØX, □.

Шаг индукции: d(S)>1. Предположим, что для любого множества дизъюнктов T такого, что d(T)<d(S) необходимость теоремы доказана. Пусть

$S=\{D_1,D_2,\ldots,D_{k-1},D_k\}$.

Так как d(S)>1, то в S существует хотя бы один неодноэлементный дизъюнкт. Будем считать, что это дизъюнкт D_k, т.е. $D_k=L\acute{U}D_k'$, где L – литерал и $D_k'^1\square$. Наряду с множеством дизъюнктов S рассмотрим еще два множества дизъюнктов

$S_1=\{D_1,D_2,\ldots,D_{k-1},L\}$,

$S_2=\{D_1,D_2,\ldots,D_{k-1},D_k^{/}\}$.

Ясно, что S_1 и S_2 невыполнимы и что $d(S_1)<a(S)$ и $d(S_2)<a(S)$. По предположению индукции из S_1 и S_2 выводим пустой дизъюнкт. Пусть

$$A_1,A_2,\ldots,A_i,\ldots,A_{l-1},A_l=\square \qquad (1)$$

вывод пустого дизъюнкта из S_1 и

$$B_1,B_2,\ldots,B_j,\ldots B_{m-1},B_m=\square \qquad (2)$$

вывод пустого дизъюнкта из S_2. Если в первом выводе не содержится дизъюнкта L, то эта последовательность дизъюнктов будет выводом из

S и необходимость теоремы доказана. Будем считать, что L содержится в первом выводе, пусть $A_i=L$. Аналогично предполагаем, что $B_j=D_k'$.

Если дизъюнкт E получается из дизъюнктов E_1 и E_2 по правилу резолюций, то будем говорить, что E *непосредственно зависит от* E_1 (и от E_2). Транзитивное замыкание отношения непосредственной зависимости назовем отношением *зависимости* (Другими словами, E зависит от $E^{/}$, если существуют дизъюнкты $E_1,\ldots,E_n$ такие, что $E=E_1,\ldots,E_n=E^{/}$ и E_1 непосредственно зависит от $E_2,\ldots,E_{n-1}$ непосредственно зависит от E_n). Преобразуем второй вывод следующим образом: к дизъюнкту B_j и всем дизъюнктам, которые от него зависят, добавим литерал L. Новая последовательность дизъюнктов

$$B_1,B_2,\ldots,B_j^{/}=D_k^{/}\acute{U}L,\ B'_{j+1},\ldots,B_m^{/} \qquad (3)$$

будет выводом из S. Если дизъюнкт B_m не зависит от B_j, то $B_m^{/}=\square$. Это означает, что из S выводим пустой дизъюнкт и необходимость теоремы доказана. Предположим, что B_m зависит от B_j. Тогда $B_m^{/}=L$. Преобразуем теперь первый вывод: на место дизъюнкта A_i (равного L) в этой последовательности подставили последовательность (3). Получим последовательность

$A_1,\ldots,A_{i-1},B_1,\ldots,B_j^{/},B'_{j+1},\ldots,B_m^{/}=L,A_{i+1},\ldots,A_l=\square$.

Эта последовательность является выводом пустого дизъюнкта из множества дизъюнктов S. Следовательно, если множество S невыполнимо, то из S выводим пустой дизъюнкт.

Теорема доказана.

Для доказательства того, что формула G является логическим следствием множества формул $F_1,\ldots,F_k$ метод резолюций применяется следующим образом. Сначала составляется множество формул $T=\{F_1,\ldots,F_k,ØG\}$. Затем каждая из этих формул приводится к конъюнктивной нормальной форме и в полученных формулах зачеркиваются знаки конъюнкции. Получается множество дизъюнктов S. И, наконец, ищется вывод пустого дизъюнкта из S. Если пустой дизъюнкт выводим из S, то формула G является логическим

следствием формул $F_1,\ldots,F_k$. Если из S нельзя вывести □, то G не является логическим следствием формул $F_1,\ldots,F_k$.

Проиллюстрируем сказанное на примере. Покажем, что формула G=Z является логическим следствием формул F_1=ØXÚY®X&Z, F_2=ØY®Z. Сформируем множество формул T={F_1,F_2,ØG}. Приведем формулы F_1 и F_2 к КНФ (формула ØG сама имеет эту форму). Мы получим, что

F_1 равносильна X&(ØYÚZ),

F_2 равносильна (YÚZ).

Тогда множество дизъюнктов S равно

{X, ØYÚZ, YÚZ, ØZ}.

Из множества S легко выводится пустой дизъюнкт:

ØYvZ, ØZ, ØY, YÚZ, У, □.

Следовательно, формула G является логическим следствием формул F_1, и F_2.

11.2. Подстановка и унификация

Перейдем к логике первого порядка. Относительно переменных в дизъюнктах будем предполагать, что они связаны кванторами общности, но сами кванторы писать не будем. Отсюда следует, что две одинаковые переменные в разных дизъюнктах можно считать различными.

Заметим, прежде всего, что в логике первого порядка правило резолюций в прежнем виде уже не годится. Действительно, множество дизъюнктов S={P(x),ØP(a)} невыполнимо, (так как предполагается, что переменная x связана квантором общности). В то же время, если использовать правило резолюций для логики высказываний, то из S пустого дизъюнкта не получить. Содержательно понятно, что в этом случае надо сделать. Поскольку дизъюнкт P(x) можно прочитать так: для любого x истинно P(x), то P(x) истинно будет и для x=a. Сделав

подстановку x=a, получим множество дизъюнктов $S^{/}$={P(a),ØP(a)}. Множество S и $S^{/}$ одновременно выполнимы (или невыполнимы). Но из $S^{/}$ с помощью прежнего правила резолюций выводится тривиальным образом. Этот пример подсказывает, что в логике первого порядка правило резолюций надо дополнить возможностью делать подстановку.

Дадим необходимые определения.

Определение. *Подстановкой* называется множество равенств

$s=\{x_1=t_1, x_2=t_2,\ldots, x_n=t_n\}$,

где $x_1,x_2,\ldots,x_n$ – различные переменные, $t_1,t_2,\ldots,t_n$ – термы, причем терм t_i не содержит переменной x_i (1£ i £ n).

Если $s = (x_1=t_1,...,x_n=t_n)$ – подстановка, а F – дизъюнкт, то через s(F) будем обозначать дизъюнкт, полученный из F *одновременной* заменой x_1 на t1; и т.д. x_n на t_n. Например, если $s=\{x_1=f(x_2), x_2=c, x_3=g(x_4)\}$, $F=R(x_1,x_2,x_3)ÚØP(f(x_2))$, то $s(F)=R(f(x_2), c, g(x_4))ÚØP(f(c))$. Аналогично определяется действие подстановки на терм.

Для удобства введем еще и *пустую подстановку* – подстановку, не содержащую равенств. Пустую подстановку будем обозначать через e.

Определение. Пусть $\{E_1,\ldots,E_k\}$ – множество литералов или множество термов. Подстановка s называется *унификатором* этого множества, если $s(E_1)=s(E_2)=\ldots=s(E_k)$. Множество *унифицируемо*, если существует унификатор этого множества.

Например, множество атомарных формул

{Q(a,x,f(x)), Q(u,y,z)}

унифицируемо подстановкой {u=a, x=y,z=f(y)}, а множество

{R(x,f(x)), R(u,u)}

неунифицируемо. Действительно, если заменить x на u, то получим множество

{R(u,f(u), R(u,u)}.

Проводить же замену u=f(u) запрещено определением подстановки, да и бесполезно, т.к. она приводит к формулам R(f(u), f(f(u))) и R(f(u), f(u)), которые тоже различны.

Если множество унифицируемо, то существует, как правило, не один унификатор этого множества, а несколько. Среди всех унификаторов данного множества выделяют так называемый наиболее общий унификатор.

Дадим необходимые определения.

Определение. Пусть $x=\{x_1=t_1, x_2=t_2,\ldots, x_k=t_k\}$ и $h=\{y_1=s_1, y_2=s_2,\ldots, y_l=s_l\}$ – две подстановки. Тогда *произведением* подстановок x и h называется подстановка, которая получается из последовательности равенств

$$\{x_1=h(t_1), x_2=h(t_2),\ldots,x_k=h(t_k), y_1=s_1, y_2=s_2,\ldots, y_l=s_l\} \qquad (4)$$

вычеркиванием равенств вида $x_i=x_i$ для $1£ i£ k$, $y_j=s_j$, если $y_j Î\{x_1,\ldots, x_k\}$, для $1 £ j £ l$.

Для обозначения результата действия подстановки на дизъюнкт мы применяем префиксную функциональную запись, поэтому произведение подстановок x и h будем обозначать через $h\circ x$, подчеркивая тем самым, что сначала действует x, а потом h.

Рассмотрим пример. Пусть $x = \{x=f(y), z=y, u=g(d)\}$, $h = \{x=c, y=z\}$. Тогда последовательность равенств (4) из определения произведения имеет вид

{x=f(y), z=z, u=g(d), x=c, y=z}.

В этой последовательности вычеркнем второе и четвертое равенство получим произведение

$h\circ x = \{x=f(y), u=g(d), y=z\}$.

Нетрудно показать, что произведение подстановок ассоциативно, т.е. для любых подстановок x,h,x выполняется равенство $x\circ(h\circ z)=(x\circ h)\circ z$, и что пустая подстановка является нейтральным элементом относительно умножения. Последнее означает выполнение равенств $s\circ e=e\circ s=s$ для любой подстановки s.

Произведение подстановок $s = \{x_1=t_1\}\circ\{x_2=t_2\}\circ\ldots\circ\{x_n=t_n\}$ мы будем иногда задавать последовательностью равенств: $s = (x_1=t_1, x_2=t_2,\ldots, x_n=t_n)$. Действие подстановки &s на дизъюнкт (и на терм) в этом случае состоит в *последовательной* (а не одновременной) замене x_1 на t_1, x_2на t_2, и т.д., x_n на t_n.

Определение. Унификатор s множества литералов или термов называется *наиболее общим унификатором* этого множества, если для любого унификатора t того же множества литералов существует подстановка x такая, что $t=x\circ s$.

Например, для множества {P(x,f(a), g(z)), P(f(b),y,v)} наиболее общим унификатором является подстановка s={x=f(b), y=f(a), v=g(z)}. Если в качестве t взять унификатор {x=f(b), y=f(a), z=c, v=g(c)}, то x={z=c}.

Если множество литералов унифицируемо, то наиболее общий унификатор существует. Это утверждение мы докажем в конце параграфа. А сейчас приведем алгоритм нахождения наиболее общего унификатора. Алгоритм называется *алгоритмом унификации*. Для изложения алгоритма потребуется понятие множества рассогласований.

Определение. Пусть M – множество литералов или термов. Выделим первую слева позицию, в к торой не для всех литералов стоит один и тот же символ. Затем в каждом литерале выпишем выражение, которое начинается символом, занимающим эту позицию. (Этими выражениями могут быть сам литерал, атомарная формула или терм). Множество полученных выражений называется *множеством рассогласований* в M.

Например, если M={P(x, f(y), a), P(x,u, g(y)), P(x, c, v)}, то первая слева позиция, в которой не все литералы имеют один и тот же символ –

пятая позиция. Множество рассогласований состоит из термов f(y), u, c. Множество рассогласований {P(x, y), ØP(a, g(z))} есть само множество. Если M={ØP(x, y),ØQ(a, v)}, то множество рассогласований равно {P(x, y), Q(a, v)}.

Алгоритм унификации

Шаг 1. Положить k=0, M_k=M, s_k=e.

Шаг 2. Если множество M_k состоит из одного литерала, то выдать s_k в качестве наиболее общего унификатора и завершить работу. В противном случае найти множество N_k рассогласований в M_k.

Шаг 3. Если в множестве N_k существует переменная v_k и терм t_k, не содержащий v_k, то перейти к шагу 4, иначе выдать сообщение о том, что множество M неунифицируемо и завершить работу.

Шаг 4. Положить s_{k+1}={v_k, t_k}○s_k, т.е. подстановка s_{k+1} получается из s_k заменой v_k на t_k и, возможно, добавлением равенства v_k=t_k. В множестве M_k выполнить замену v_k=t_k, полученное множество литералов взять в качестве M_{k+1}.

Шаг 5. Положить k=k+1 и перейти к шагу 2.

Пусть M={P(x,f(y)), P(a,u)}. Проиллюстрируем работу алгоритиа унификации на множестве M. На первом проходе алгоритма будет найдена подстановка s_1={x=a}, так как множество рассогласований N_0={x,a}. Множество M_1 будет равно {P(a,f(y)),P(a,u)}. На втором проходе алгоритма подстановка будет расширена до s_2={x=a, u=f(y)} и M_2={P(a,f(u))}. Так как M_2 состоит из одного литерала, то алгоритм закончит работу и выдаст s_2.

Рассмотрим второй пример. Пусть M={P(x,f(y)), P(a,b)}. На первом проходе алгоритма будет найдена подстановка s_1=(x=a) и M_1={P(a,f(y)), P(a,b)}. На третьем шаге второго прохода будет выдано сообщение о том, что множество M неунифицируемо, так как множество рассогласования N_1={f(y),a} не содержит переменной.

Отметим, что при выполнении шага 4 из множества M_k удаляется одна из переменных (переменная v_k), а новая переменная не возникает. Это означает, что алгоритм унификации всегда заканчивает работу, так как шаг 4 не может выполняться бесконечно. Довольно ясно, что если алгоритм заканчивает работу на шаге 3, то множество M неунифицируемое. Также понятно, что если алгоритм заканчивает работу на шаге 2, то s_k – унификатор множества M. А вот то, что s_k – наиболее общий унификатор, доказать не так то просто. Тем не менее, сделаем это.

Теорема 4.2. Пусть M – конечное непустое множество литералов. Если M унифицируемо, то алгоритм унификации заканчивает работу на шаге 2 и выдаваемая алгоритмом подстановка s_k – наиболее общий унификатор множества M.

Доказательство. Пусть t – некоторый унификатор множества M. Индукцией по k докажем существование подстановки a_k такой, что t=a_k∘s_k.

***База индукции*:** k=0. Тогда s_k=e и в качестве a_k можно взять t.

Шаг индукции: Предположим, что для всех значений k, удовлетворяющих неравенству 0£k£ l, существует подстановка a_k такая, что t=a_k○s_k.

Если s_l(M) содержит один литерал, то на следующем проходе алгоритм остановится на шаге 2. Тогда s_l будет наиболее общим унификатором, поскольку t=a_l○s_l.

Пусть s_l(M) содержит более одного литерала. Тогда алгоритм унификации найдет множество рассогласований N_l. Подстановка a_l должна унифицировать множество N_l, поскольку t=a_l○s_l – унификатор множества M. Поскольку N_l – унифицируемое множество рассогласований, то оно содержит (хотя бы одну) переменную v.

Пусть t – терм из N_l, отличный от v. Множество N_l унифицируется подстановкой a_l, поэтому a_l(v)=a_l(t). Отсюда следует, что t не содержит v. Можно считать, что на шаге 4 алгоритма для получения s_{l+1} использовано равенство v=t, т.е. s_{l+1}={v=t}○s_l. Из равенства a_l(v)=a_l(t) следует, что a_l содержит равенство v=a_l(t).

Пусть $a_{l+1}=a_l\backslash\{v=a_l(t)\}$. Тогда $a_{l+1}(t)=a_l(t)$, так как t не содержит v. Далее, имеем равенства

$a_{l+1}\circ\{v=t\}=a_{l+1}È\{v=a_{l+1}(t)\}=a_{l+1}È\{v=a_l(t)\}=a_l$.

Это означает, что $a_l=a_{l+1}\circ\{v=t\}$. Следовательно,

$t=a_l\circ s_l=a_{l+1}\circ\{v=t\}\circ s_l=a_{l+1}\circ s_{l+1}$.

Итак, для любого k существует подстановка a_k такая, что $t=a_k\circ s_k$. Так как множество M унифицируемо, то алгоритм должен закончить работу на шаге 2. Тогда последняя подстановка s_k будет унификатором множества M, поскольку множество $s_k(M)$ будет наиболее общим унификатором, так как для произвольного унификатора t существует подстановка s_k такая, что $t=a_k\circ s_k$.

Теорема доказана.

11.3. Метод резолюций для логики первого порядка

Начнем с формулировки правила резолюций.

Определение. *Правилом резолюций в логике предикатов* называется правило из дизъюнктов $ØP(t_1,...,t_n)ÚF$ и $P(s_1,...,s_n)ÚG$ выводим дизъюнкт $s(F)Ús(G)$, где s – наиболее общий унификатор множества $\{P(t_1,...,t_n), P(s_1,...,s_n)\}$. Дизъюнкт $s(F)Ús(G)$ называется *бинарной резольвентой* первых двух дизъюнктов, а литералы $ØP(t_1,…,t_n)$ и $P(s_1,…,s_n)$ *отрезаемыми* литералами.

Например, с помощью правила резолюций из дизъюнктов $ØQ(a,f(x))ÚR(x)$ и $Q(u,z)ÚØP(z)$ можно вывести дизъюнкт $R(x)ÚØP(f(x))$, используя подстановку $s=\{u=a, z=f(x)\}$.

В отличие от логики высказываний, в логике предикатов нам понадобится еще одно правило.

Определение. *Правилом склейки в логике предикатов* называется правило: из дизъюнкта $àP(t_1,...,t_n)Ú...ÚàP(s_1,...,s_n)ÚF$ выводим дизъюнкт

$s(àP(t_1,...,t_n)Ú s(F)$, где s – наиболее общий унификатор множества $\{P(t_1,...,t_n),...,P(s_1,...,s_n)\}$, à – знак отрицания или его отсутствие. Дизъюнкт $s(◊P(t_1,…,t_n)Ús(F)$ называется *склейкой* первого дизъюнкта. (Отметим, что если знак отрицания стоит перед одной из записанных выше атомарных формул, то он стоит и перед другими.)

Например, правило склейки, примененное к дизъюнкту $ØP(x,y)ÚØP(y,x)ÚØP(a,a)ÚQ(x,y,v)$, дает дизъюнкт $ØP(a,a)ÚQ(a,a,v)$.

Определение вывода в логике первого порядка немного отличается от аналогичного определения в логике высказываний.

Определение. Пусть S – множество дизъюнктов.*Выводом из множества дизъюнктов* S называется последовательность дизъюнктов

$D_1,D_2,…,D_n$,

такая, что каждый дизъюнкт D_i принадлежит S, выводим из предыдущих дизъюнктов по правилу резолюций или выводим из предыдущего по правилу склейки.

Как и в логике высказываний, дизъюнкт D выводим из S, если существует вывод из S, последним дизъюнктом которого является D.

Приведем пример. Пусть $S=\{ØB(x)ÚØC(x)ÚT(f(x)), C(y)ÚT(f(z)), B(a)\}$. Тогда последовательность

$D1=ØB(x)ÚØC(x)ÚT(f(x))$,

$D2=C(y)ÚT(f(z))$,

$D3=ØB(x)ÚT(f(x))ÚT(f(z))$,

$D4=ØB(x)ÚT(f(x))$,

$D5=B(a)$,

$D6=T(f(a))$.

является выводом из S. Отметим, что $D_1,D_2,D_5 \hat{I} S$, дизъюнкт D_3 выводим из D_1 и D_2 по правилу резолюций, дизъюнкт D_6 выводим из D_4 и D_5 по тому же правилу, а D_4 выводим из D_3 по правилу склейки.

Как и в логике высказываний, в логике первого порядка есть утверждение, называемое теоремой о полноте. Фактически это утверждение совпадает с формулировкой 4.1. Тем не менее приведем его.

Теорема 4.3. Множество дизъюнктов S логики первого порядка невыполнимо тогда и только тогда, когда из S выводим пустой дизъюнкт.

Теорема имеет довольно сложное доказательство. Оно будет приведено в 11.6.

В данном параграфе мы ограничимся примером применения метода резолюций и рядом определений, необходимых для доказательства теоремы 4.3.

Для доказательства логичности следствия формулы G из формул $F_1,\ldots,F_k$ метод резолюций в логике предикатов применяется почти так же, как и в логике высказываний. А именно, сначала составляется множество формул $T=\{F_1,\ldots,F_k,\text{Ø}G\}$. Затем каждая из формул этого множества приводится к сколемовской нормальной форме, в полученных формах вычеркиваются кванторы общности и связки конъюнкции. Получается множество дизъюнктов S. На последнем этапе находится вывод пустого дизъюнкта из множества S. Напомним, что все переменные в дизъюнктах предполагаются связанными кванторами общности. Это означает, что *метод резолюций для доказательства логичности может применяться лишь в случае, когда формулы* $F_{1,\ldots}$, F_k и G *не имеют свободных переменных.* Если все же формулы содержат свободные переменные, то их надо заменить константами (такими, которые отсутствуют в этих формулах).

Рассмотрим пример. Пусть

$$F1=(\$x)[П(x) \& (\forall y)(C(y) \to З(x,y))],$$

$$F2=(\forall x)(\forall y)[П(x) \& Л(y) \to \text{Ø} З(x,y)],$$

$$G=(\forall x)(C(x) \to \text{Ø}Л(x)).$$

Докажем, что формула G является логическим следствием множества формул F_1,F_2. Для этого достаточно доказать невыполнимость множества $T=\{F_1,F_2,\text{Ø}G\}$. Каждую из формул множества T приведем к сколемовской нормальной форме; получим формулы

$$(\forall y)[П(a) \& (\text{Ø}C(y) \text{Ú} З(a,y))],$$

$$(\forall x)(\forall y)[\text{Ø}П(x) \text{Ú} \text{Ø}Л(y) \text{Ú} \text{Ø}З(x,y)],$$

$$C(в) \& Л(в).$$

Тогда множество S будет содержать дизъюнкты: $D_1=П(a)$, $D_2=\text{Ø}C(y) \text{Ú} З(a,y)$, $D_3=\text{Ø}П(x) \text{Ú} \text{Ø}Л(y) \text{Ú} \text{Ø}З(x,y)$, $D_4=C(в)$, $D_5=Л(в)$. А последовательность дизъюнктов D_1, D_3, $\text{Ø}Л(y) \text{Ú} \text{Ø}З(a,y)$, D_5, $\text{Ø}З(a,в)$, D_2, D_4, $З(a,в)$, □ будет выводом из S. Следовательно, формула G является логическим следствием формул F_1 и F_2.

Введем, как было обещано выше, ряд определений, необходимых в следующих параграфах.

Напомним, что мы условились не писать в дизъюнктах повторяющиеся литералы. Это позволяет нам смотреть, если это необходимо, на дизъюнкт как на множество литералов. Если смотреть на дизъюнкты как на множество литералов, то результат применения правила резодюций к дизъюнктам D_1 и D_2 с отрезаемыми литералами L_1 и L_2 можно записать так

$$D=[s(D_1)-s(L_1)] \grave{E} [s(D_2)-s(L_2)],$$

где s – наиболее общий унификатор L_1 и $\text{Ø}L_2$.

Определение. *Револьвентой* дизъюнктов D_1 и D_2 называется одна из следующих бинарных револьвент:

1) бинарная резольвента дизъюнктов D_1 и D_2,

2) бинарная резольвента склейки D_1 и дизъюнкта D_2,

3) бинарная резольвента дизъюнкта D_1 и склейки D_2,

4) бинарная резольвента склейки D_1 и склейки D_2.

Приведем пример. Пусть D_1=ØP(y)ÚØP(g(x))ÚR(f(y)), D_2=P(g(a))ÚQ(b). Склейка дизъюнкта D_1 есть дизъюнкт D_1'=ØP(g(x))ÚR(f(g(x)). Бинарная резольвента D_1 и D_2' равна R(f(g(a)))ÚQ(b). Следовательно, последний дизъюнкт есть резольвента дизъюнктов D_1 и D_2.

Определение. Если D – дизъюнкт, а s – подстановка, то дизъюнкт s(D) называется *примером дизъюнкта* D.

Следующее утверждение часто называют *леммой о подъеме*.

Теорема 4.4. Если D_1' – пример дизъюнкта D_1, D_2' – пример дизъюнкта D_2 и D' – резольвента D_1' и D_2', то существует резольвента D дизъюнктов D_1 и D_2 такая, что D' – пример D.

Доказательство. Если D_1 и D_2 имеют лбщие переменные, то заменой переменных в одном из дизъюнктов можно добиться того, что переменные дизъюнкта D_1 отличны от переменных дизъюнкта D_2. Будем поэтому считать, что D_1 и D_2 не имеют общих переменных.

Так как D_1' – пример D_1 и D_2' – пример D_2, то существуют подстановки a_1 и a_2 такие, что $D_1'=a_1(D_1)$ $D_2'=a_2(D_2)$. Последовательность $a=(a_1,a_2)$ также будет подстановкой и поскольку D_1 и D_2 не имеют общих переменных, то $D_1'=a(D_1)$ и $D_2'=a(D_2)$.

Дизъюнкт D' является резольвентой дизъюнктов D_1' и D_2'. Это означает, что существуют литералы L_1'ÎD_1' и L_2'ÎD_2' и подстановка t такие, что t наиболее общий унификатор L_1' и ØL_2' и

$$D'=(t(D_1')-t(L_1'))È(t(D_2')-t(L_2')). \qquad (1)$$

(Если при получении резольвенты D' к дизъюнктам D_1' т D_2' применялись склейки, то будем считать, что они учтены подстановками a_1 и a_2.)

Пусть $L_1^1,\ldots,L_1^r$ – литералы дизъюнкта D_1, которые подстановкой a переводятся L_1', а $L_2^1,\ldots,L_2^s$ – литералы дизъюнкта D_2, которые подстановкой a переводятся в L_2'. Литералы $L_1^1,\ldots,L_1^r$, следовательно, унифицируемы, а поэтому существует наибелее общий унификатор b_1 для этого множества. Дизъюнкт $b_1(L_1^1)$ (равный $b_1(L_1^2),\ldots,b_1(L_1^r)$) обозначим через L_1. По определению наиболее общего унификатора найдется подстановка g_1, для которой выполняется равенство $a_1=g_1\circ b_1$. По аналогичным соображениям, существуют подстановки b_2 и g_2 такие, что b_2 – наиболее общий унификатор множества литералов $L_2^1,\ldots,L_2^s$ и $a_2=g_2\circ b_2$. Литерал $b_2(L_2^1)$ обозначим через L_2. Дегко видеть, что L_1 и L_2 не имеют общих переменных. Поскольку дизъюнкты D_1 и D_2 также не имеют общих переменных, то можно считать, что $b_1=b_2=b$, $g_1=g_2=g$ и $a=g\circ b$. Сказанное в этом абзаце иллюстрируется рисунками 4.1 и 4.2.

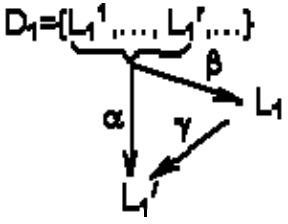


Рис. 4.1

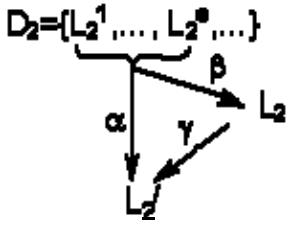


Рис. 4.2

Литералы L_1' и ØL_2', как отмечено выше, унифицируемы подстановкой t. Следовательно, литералы L_1 и ØL_2 также унифицируемы (подстановкой $t\circ g$). Отсюда следует, что существует наиболеее общий унификатор s множества $\{L_1$ Ø$L_2\}$ (см.рис.4.3). Возьмем в качестве D дизъюнкт

$$D=[s(b(D_1))–s(L_1)]È[s(b(D_2]–s(L_2)] \qquad (2)$$

Ясно, что D – резольвента дизъюнктов D_1 и D_2. Осталось показать. Что D' –пример D.

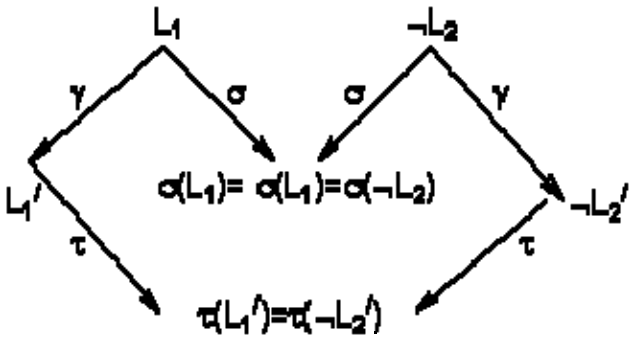


Рис. 4.3

Так как s – наиболее общий унификатор L_1 и ØL_2, то существует подстановка s такая, что t○g=d○s. В таком случае из последнего равенства, равенств (1), (2) и a=g○b следует, что

D′=(t(D$_1$′)–t(L$_1$′))È(t(D$_2$′)–t(L$_2$′))=[t(a(D$_1$))–t(a(L$_1^1$))]È[t(a(D$_2$))–(r(a(L$_2^1$))=

[(t○a)(D$_1$)–(t○a)(L$_1^1$)]È[(t○a)(D$_2$)–(t○a)(L$_2^1$)]=

[(t○g○b)(D$_1$)–(t○g○b)(L$_1^1$)]È[(t○g○b)(D$_2$)–(t○g○b)(L$_2^1$)]=

[(d○s○b)(D$_1$)–(d○s○b)(L$_1^1$)]È[(d○s○b)(D$_2$)–(d○s○b)(L$_2^1$)]=

d[s(b(D$_1$))–s(L$_1$)]Èd[s(b(D$_2$))–s(L$_2$)]=d(D).

Мы доказали, что D′ – пример D.

Теорема доказана.

11.4. Эрбрановский универсум множества дизъюнктов

По определению интерпретации ее областью может быть любое непустое множество M. Было бы удобно иметь одно множество в качестве области интерпретации. В случае, когда решается вопрос о

выполнимости множества дизъюнктов, таким множеством является так называемый эрбрановский универсум.

Пусть S – множество дизъюнктов.

Введем следующее обозначение. Через H_0 обозначим множество констант, содержащихся в S. Если S не содержит констант, то H_0 состоит из одной константы, скажем a, т.е. H_0={a}. Предположим, что введено множество H_i. Тогда H_{i+1} есть объединение множества H_i и термов вида $f(t_1,\ldots,t_n)$, где $t_1,\ldots,t_n$ÎH_i, f – символ n-местной функции, содержащейся хотя бы в одном из дизъюнктов множества S.

Определение. Множество $H_\infty=H_0ÈH_1È\ldots ÈH_nÈ\ldots$ называется *эрбрановским универсумом* множества дизъюнктов S.

Приведем три примера, которые будем использовать в дальнейшем.

Пример 1. Пусть S={P(x), ØP(x)ÚQ(f(y)), ØQ(f(a))}. Тогда

H_0={a},

H_1={a,f(a)},

. . .

H_∞={a,f(a)}, f(f(a)),…}.

Пример 2. Пусть S={P(x), Q(x)ÚØR(y)}. Множество S не содержит констант, поэтому H_0={a},. Так как дизъюнкты из S не содержат функциональных символов, то $H_0=H_1=H_2=\ldots=H_\infty$={a}.

Пример 3. Пусть S={P(x), ØP(b)ÚQ(y,f(y,a))}. Тогда H_∞={a, b, f(a,a), f(a,b), f(b,a), f(b,b), f(f(a,a),a),…}.

Эрбрановский универсум, как мы видим. Определяется не всем множеством дизъюнктов S, а только символами функций и константами, входящими в дизъюнкты из S.

Определение. Множество атомарных формул вида $P(t_1,\ldots,t_n)$, где $t_1,\ldots,t_n \in H_\infty$, а P – символ n-местного предиката, входящий в дизъюнкты из S, называется *эрбрановским базисом* множества дизъюнктов S.

Для множества дизъюнктов S из примера 1 эрбрановским базисом будет множество атомарных формул $B_1=\{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))),\ldots\}$. Эрбрановским базисом множества дизъюнктов S из примера 2 будет множество $B_2=\{P(a), Q(a), R(a)\}$.

Пусть термин «выражение» означает терм, атомарную формулу, литерал или дизъюнкт. Тогда *основным выражением* будем называть выражения, несодержащие переменных.

Определение. Пусть D – дизъюнкт из множества дизъюнктов S. *Основным примером* дизъюнкта D называется дизъюнкт, полученный из D заменой переменных на элементы эрбрановского универсума H_∞.

Пусть S – дизъюнкт из примера 1. Тогда дизъюнкт $\neg Q(f(a))$ имеет один основной пример – сам дизъюнкт, множество основных примеров дизъюнкта $\neg P(x) \vee Q(f(y))$ бесконечно $\{\neg P(a) \vee Q(f(a)), \neg P(f(a)) \vee Q(f(a)), \neg P(a) \vee Q(f(f(a))),\ldots\}$. Если S – множество дизъюнктов из примера 2, то каждый из дизъюнктов этого множества S имеет один основной пример.

Как утверждалось в начале параграфа (и будет доказано в конце), для решения вопроса о выполнимости дизъюнктов в качестве области интерпретации достаточно рассматривать только эрбрановский универсум. Оказывается, можно еще ограничить и саму интерпретацию до так называемой H-интерпретации.

Определение. Пусть H_∞ – эрбрановский универсум множества дизъюнктов S. Интерпретация j с областью H_∞ называется H – *интерпретацией множества* S, если она удовлетворяет следующим условиям:

1) для любой константы c из S выполняется равенство $j(c)=c$,

2) если f – символ n-местной функции из S, то jf – функция, определенная на H_∞ равенством

$$(jf)(t_1,\ldots,t_n)=f(t_1,\ldots,t_n)$$

для любых $t_1,\ldots,t_n \in H_\infty$.

Если $B=\{B_1,B_2,\ldots,B_n,\ldots\}$ – эрбрановский базис множества дизъюнктов S, то H-интерпретацию j удобно представлять в виде множества литералов

$$\{L_1,L_2,\ldots,L_n,\ldots\},$$

где L_i есть B_i, если $j(B_i)=1$, и $L_i=\neg B_i$, если $j(B_i)=0$.

Например, если S – множество дизъюнктов из примера 1, то эрбрановскими интерпретациями будут

$$j_1=\{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))),\ldots\}$$

$$j_2=\{P(a), Q(a), \neg P(f(a)), \neg Q(f(a)), P(f(f(a))), Q(f(f(a))), \neg P(f(f(f(a)))),\ldots\}.$$

$$j_3=\{P(a), \neg Q(a), P(f(a)), \neg Q(f(a)), P(f(f(a))), \neg Q(f(f(a))),\ldots\}.$$

Последнее, что нужно сделать, чтобы доказать основное утверждение этого параграфа (теорему 4.5), ввести понятие H-интерпретации j*, соответствующей (произвольной) интерпретации j множества S.

Предположим, что S содержит хотя бы одну константу. Если j – интерпретация множества S с областью M, то для любого элемента h эрбрановского универсума значение j(h) определено (и является элементом множества M.)

Определение. Пусть j – интерпретация множества S с областью M. Тогда H-*интерпретацией* j*, *соответствующей интерпретации* j называется H-интерпретация, удовлетворяющая следующему условию: для любых элементов $t_1,\ldots,t_n$ эрбрановского универсума выполняется эквиваленция:

$$(j^*P)(t_1,\ldots,t_n)=1 \Leftrightarrow (jP)(j(t_1),\ldots,j(t_n))=1$$

для любого символа предиката P.

Приведем пример. Пусть S – множество дизъюнктов из примера 1. Напомним, что S={P(x), ØP(x)ÚQ(f(y)), ØQ(f(a))} и эрбрановский базис множества S есть B={P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))),…}. Рассмотрим интерпертацию j с областью M={1,2}, определяемую равенством j(a)=1 и таблицей 4.1.

Таблица 4.1

	jf	jP	jQ
1	2	0	1
2	1	1	1

(В столбцах jP и jQ цифры 0 и 1 – истинностные значения.) Тогда

j*={ØP(a), Q(a), P(f(a)), Q(f(a)), ØP(f(f(a))),…}.

Рассмотрим теперь случай, когда S не содержит констант. Пусть j – интерпретация множества S с областью M и a – константа, образующая эрбрановский универсум $H_{¥}$. В этом случае значение j(a) неопределено. Для получения H-интерпретации j* расширяем функцию j на a полагая j(a) равным произвольному элементу из M. Далее поступаем так, как описано выше. Если множество M неодноэлементно, то мы можем получить не одну H-интерпретацию j*, соответствующую j. Нетрудно привести пример, когда H-интерпретаций j* столько же, сколько элементов в множестве M.

Следующее утверждение непосредственно следует из определений.

Лемма. Пусть j – интерпретация с областью M, при которой все дизъюнкты из S истинны. Тогда все дизъюнкты из S истинны при любой H-интерпретации j*, соответствующей j.

Теорема 4.5. Множество дизъюнктов S невыполнимо тогда и только тогда, когда S ложно при всех H-интерпретациях, т.е. для любой H-интерпретации множества S в S найдется дизъюнкт, который ложен при этой H-интерпретации.

Доказательство. Необходимость очевидна. Действительно, невыполнимость множества S означает, что это множество ложно при любой интерпретации. В том числе и при любой H-интерпретации. Достаточность следует из леммы, поскольку если S выполнимо, то существует хотя бы одна интепретация j, при которой все дизъюнкты из S истинны. Но тогда все дизъюнкты из S будут истинны и при H-интерпретации j*.

Теорема доказана.

11.5. Семантические деревья, теорема Эрбрана

В предыдущем параграфе мы видели, что для получения ответа на вопрос о выполнимости множества дизъюнктов можно рассматривать не все интерпертации, а только H-интерпретации. В данном параграфе мы в этом направлении продвинемся еще дальше. Мы фактически покажем, что для решения упомянутого вопроса можно ограничиться конечными подмножествами эрбрановского универсума. Основным понятием этого параграфа будет понятие семантического дерева.

На дерево будем смотреть, как на корневое ориентированное дерево. Изображать дерево будем растущим вниз, ориентацию дуг указывать не будем. Напомним, что листом дерева называется вершина, из которой не выходит не одна дуга. Путь в дереве – это последовательность дуг $e_1,e_2,\ldots,e_k,\ldots$ такая, что если дуга e_i заходит в вершину v, то дуга e_{i+1} выходит из этой вершины. Путь в дереве называется максимальным, если к нему нельзя добавить ни одной дуги. Для каждой вершины существует единственный путь от корня к этой вершине. Если вершина является листом, то этот путь максимален. Если p – путь в дереве, то через I(p) будем обозначать объединение всех литералов, принадлежащим дугам пути. В случае, когда p – путь от корня до вершины v, то вместо I(p) будем писать I(v). Мы будем использовать понятие поддерева несколько в ином смысле, нежели в теории графов. А именно, поддеревом дерева T будем называть подграф T′, удовлетворяющий следующим условиям:

1) T′ – дерево;

2) T′ содержит корень дерева T,

3) если из v в v′ идет дуга в дереве T, v и v′Î T′, то T′ содержит все вершины, в которые из v идет дуга.

Определение. Пусть S – множество дизъюнктов, B – эрбрановский базис для S. *Семантическим деревом для* S называется корневое дерево, каждой дуге которого приписано непустое множество формул из B или их отрицаний так, что выполнены следующие условия.

1. Из любой вершины выходит конечное число дуг e1,…,ek; если ci – конъюнкция литералов, приписанных дуге ei, то c1Úc2Ú…Úck – тождественно истинная формула.

2. Для любой вершины v множество I(v) не содержит противоположных литералов.

Рассмотрим примеры. Пусть S – множество дизъюнктов из примера 2 предыдущего параграфа, B – его эрбрановский базис. Напомним, что B{P(a), Q(a), R(a)}. Для простоты в примерах вместо P(a), Q(a) и R(a) будем писать просто P,Q,R. На рисунках 4.4 и 4.5 приведены примеры семантических деревьев для S. Семантическое дерево может быть бесконечным. На рисунке 4.6 приведен пример семантического дерева для множества дизъюнктов S из примера 1 §4. Напомним, что эрбрановский базис в этом случае есть B{P(a), Q(a), P(f(a)), Q(f(a)),…}.

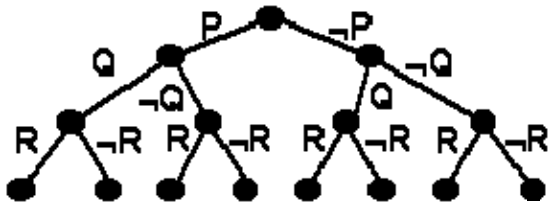


Рис. 4.4

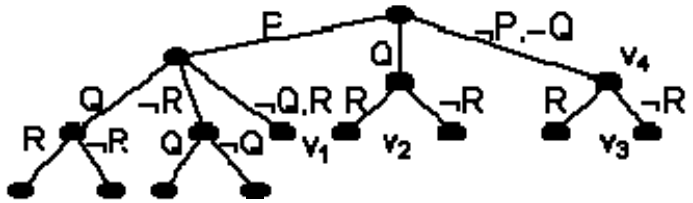


Рис. 4.5

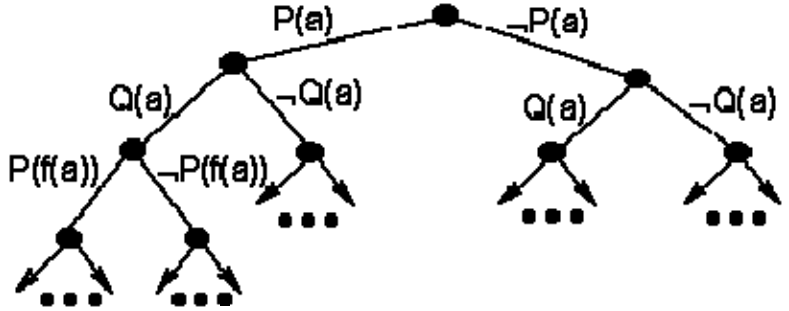


Рис. 4.6

Определение. Пусть B – эрбрановский базис множества дизъюнктов S. Семантическое дерево T называется *полным*, если для любого элемента A базиса B множество I(p) содержит либо A, либо ØA.

Семантические деревья, изображенные на рис.4.4 и 4.6 являютя полными, а семантическое дерево изображенное на рис.4.5 – неполным.

Определение. Вершина v семантического дерева называется *опровергающей*, если I(v) опровергает основной пример некоторого дизъюнкта S. Вершина v называется *максимальной опровергающей*, если вершина v′, предшествующая v, опровергающей не является.

Рассмотрим в качестве примера дерево, изображенное на рис.4.5. Напомним, что оно является семантическим деревом множества

дизъюнктов S={P(x), Q(x)ÚØR(y)}. Вершины v_1 и v_3 будут опровергающими вершинами, так как множество I(v_1), равное {ØQ(a)&R(a), P(a)}, опровергает основной пример Q(a)ÚØR(a) дизъюнкта Q(x)ÚØR(y), а множество I(v_3), равное {R(a),ØP(a)&ØQ(a)} опровергает основной пример P(a) дизъюнкта P(x). Вершина v_1 будет максимальной опровергающей, а вершина v_3 не будет максимальной, потому что опровергающей является предшествующая ей вершина v_4. Вершина v_2 опровергающей не является.

Определение. Вершина v семантического дерева называется *выводящей*, если все непосредственно следующие за ней вершины являются максимальными опровергающими.

Пусть S={P(x), ØP(x)ÚQ(f(y)), ØQ(f(a))} множество дизъюнктов примера 1 предыдущего параграфа. Дерево, изображенное на рис.4.7, является семантическим деревом для S. Вершина v этого дерева является выводящей вершиной, а никакие другие вершины выводящими не являются.

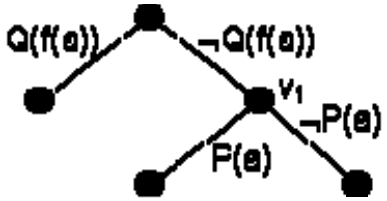


Рис. 4.7

Определение. Семантическое дерево называется *замкнутым*, если каждый его лист является максимальной опровергающей вершиной.

Дерево, изображенное на рис.4.4, замкнуто, а деревья на рис.4.5 и 4.6 незамкнуты.

Следующее утверждение – знаменитая теорема математической логики, которая является основой многих алгоритмов доказательства теорем. Она называется теоремой Эрбрана.

Теорема 4.6. Множество дизъюнктов S невыполнимо тогда и только тогда, когда любое полное семантическое дерево множества S имеет конечное замкнутое поддерево.

Доказательство теоремы 4.6 использует известное математике утверждение, которое называется леммой Кенига. Сформулируем ее.

Лемма. Если T – бесконечное дерево, из каждого узла которого выходит конечное число дуг. Тогда дерево T содержит бесконечный p путь, начинающийся от корня.

Доказательство леммы Кенига приводить не будем.

Приведем доказательство теоремы 4.6. Докажем вначале необходимость. Пусть множество дизъюнктов S невыполнимо и T – полное семантическое дерево для S. Рассмотрим максимальный путь p в дереве T. По определению полного семантического дерева для каждой атомарной формулы A эрбрановского базиса B, либо A, либо ØA принадлежит I(p). Это означает, что I(p) есть H-интерпретация множества S. Поскольку S невыполнимо, то I(p) опровергает основной пример D′ некоторого дизъюнкта D из S. Дизъюнкт D′ конечен, поэтому путь p должен проходить через максимальную опровергающую вершину дерева T. В каждом максимальном пути отметим такую вершину. Пусть T – корневое поддерево дерева T, листьями которого являются отмеченные вершины. В силу леммы Кенига. T′ – конечное поддерево дерева T. Дерево T′ по построению является замкнутым. Необходимость доказана.

Докажем достаточность. Пусть полное семантическое дерево T содержит конечное замкнутое поддерево T′. По определению поддерева (см.начало данного параграфа) отсюда следует, что каждый максимальный путь дерева T содержит опровергающую вершину. Множество всех максимальных путей полного семантического дерева исчерпывает все H-интерпретации множества S. Следовательно S ложно при всех H-интерпретациях. По теореме 4.5 S невыполнимо.

11.6. Полнота метода резолюций в логике предикатов

Параграф посвящен доказательству теоремы 4.3. Напомним ее формулировку.

Теорема. Множество дизъюнктов S невыполнимо тогда и только тогда, когда из S выводим пустой дизъюнкт.

Доказательство. Пусть множество дизъюнктов S невыполнимо и $B=\{A_1,A_2,\ldots,A_n,\ldots\}$ – эрбрановский базис для S. Рассмотрим полное семантическое дерево T, изображенное на рис.4.8.

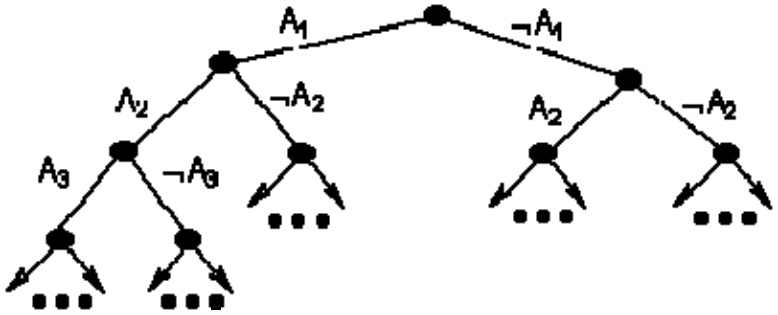


Рис. 4.8

По теореме Эрбрана T содержит конечное замкнутое семантическое поддерево $T^/$. Если $T^/$ состоит только из корня, то $\square\hat{I}S$ и поэтому $\square$ выводим из S. Предположим, что $T^/$ состоит не только из корня. Тогда $T^/$ имеет вершину v, потомки v_1 и v_2 которой являются максимальными опровергающими для множества S вершинами. Пусть

$$I(v)=\{L_1, L_2,\ldots, L_k\},$$

$$I(v_1)=\{L_1, L_{2,\ldots},L_k, A_{k+1}\},$$

$$I(v_2)=\{L_1, L_2,\ldots, L_k, \O A_{k+1}\}.$$

Существует дизъюнкт $D_1\hat{I}S$ такой, что его основной пример $D_1^/$ опровергается в $I(v_1)$, и существует дизъюнкт $D_2\hat{I}S$ такой, что его основной пример $D_2^/$ опровергается в $I(v_2)$. Так как дизъюнкты $D_1^/$ и $D_2^/$ не опровергаются в I(v), то $D_1^/$ содержит $\O A_{k+1}$, а $D_2^/ – A_{k+1}$. Применим к $D_1^/$ и $D_2^/$ правило резюлюций, отрезая литералы $\O A_{k+1}$ и A_{k+1}, получим дизъюнкт $D^/$:

$$D^/=(D_1^/ - \O A_{k+1})\grave{E}(D_2^/ – A_{k+1}).$$

Отметим, что дизъюнкт $D_1^/ – A_{k+1}$ ложен в I(v), поскольку в противном случае, $D_1^/$ был бы истинен в $I(v_1)$. Аналогично заключаем, что $D_2^/ – A_{k+1}$ ложен в той же интерпретации I(v).

Отсюда следует, что $D^/$ ложен при I(v).

По лемме о подъеме (теорема 4.4) существует дизъюнкт D, который является резольвентой дизъюнктов D_1 и D_2. Ясно, что D опровергается в I(v). Рассмотрим множество дизъюнктов $S\grave{E}\{D\}$. Замкнутое семантическое дерево $T^{//}$ для этого множества дизъюнктов можно получить вычеркиванием некоторых вершин (и идущих в них дуг) дерева $T^/$. А именно, в дереве $T^/$ вычеркиваем все дуги и вершины, которые лежат ниже первой вершины, где дизъюнкт $D^/$ ложен. Полученное таким образом дерево $T^{//}$ содержит меньше вершин, нежели дерево $T^/$, так как $v_1,v_2\ddot{I}T^{//}$.

Применим описанный выше процесс к $T^{//}$, мы получим резольвенту дизъюнктов из $S\grave{E}\{D\}$. Расширим множество $S\grave{E}\{D\}$ за счет этой резольвенты, придем к конечному замкнутому дереву с меньшим числом вершин, нежели $T^{//}$. В конце концов, получим замкнутое семантическое дерево, состоящее только из корня. Это возможно, лишь в случае, когда множество S, расширенное резольвентами, содержит пустой дизъюнкт, Следовательно, $\square$ выводим из S. Необходимость условия теоремы доказана.

Докажем достаточность. Пусть пустой дизъюнкт выводим из S, и $D_1,D_2,\ldots,D_n=\square$ – вывод из S. Предположим, что S выполнимо в некоторой интерпретации. Тогда, поскольку правила резолюций и правило склейки сохраняют истинность, то все дизъюнкты вывода, в том числе и пустой, являются истинными в этой интерпретации. Полученное противоречие доказывает, что S невыполнимо.

Теорема доказана.

11.7. Стратегии метода резолюций

В множестве дизъюнктов существует, как правило, не одна пара дизъюнктов, к которым можно применить правило резолюций. Способ выбора дизъюнктов и летералов в них, к которым применяется правило резолюций (и правило склейки) для получения резольвенты, называется *стратегией* метода. В этом параграфе будет рассмотрено три стратегии: стратегия насыщения уровней. Предпочтения более коротких дизъюнктов и вычеркивания.

Стратегия насыщения уровней. Наиболее простой с идейной точки зрения способ выбора дизъюнктов для получения резольвенты состоит в организации полного перебора возможных вариантов. Этот перебор можно организовать следующим образом. Пусть $S_0=S$ – исходное множество дизъюнктов. Будем считать, что S_0 упорядочено. Пусть D_2 пробегает по порядку множество дизъюнктов S_0, начиная со второго. В качестве D_1 берем последовательно дизъюнкты из S_0, предшествующие D_2 начиная с первого, и формируем последовательность S_1, состоящее из всевозможных резольвент дизъюнктов D_1 и D_2. (Порядок на S_1 определяется порядком добавления дизъюнктов в S_1.) Предположим, что получены последовательности дизъюнктов $S_0, S_1,\ldots,S_{n-1}$ и $n>1$. Тогда последовательность S_n получается следующим образом. В качестве D_2 берутся по порядку дизъюнкты из S_{n-1}, а в качестве D_1 – дизъюнкты из $S_0ÈS_1È\ldots ÈS_{n-1}$, предшествующие D_2. Последовательность S_n будет состоять из всевозможных резольвент дизъюнктов D_1 и D_2. Процесс порождения резольвент прекращается, как только получается пустой дизъюнкт.

Описанная в предыдущем абзаце стратегия называется *стратегией насыщения уровней*. (Уровни – это последовательности $S_0, S_1,\ldots,S_n,\ldots$)

Проследим, как она работает на примере множества дизъюнктов S={XÚY, ØXÚØY, XÚZ, ØXÚZ, ØZ}:

S_0: (1) XÚY,
(2) ØXÚØY,
(3) XÚZ,
(4) ØXÚZ,
(5) ØZ,
S_1: (6) YÚØY_2 (из (1) и (2)),
(7) XÚØX (из (1) и (2)),
(8) ØYÚZ (из (2) и (3)),
(9) YÚZ (из (1) и (3)),
(10) Z (из (3) и (4),
(11) X (из (3) и (5),
(12) ØX (из (4) и (5)
S_2: (13) XÚY (из (1) и (6),
(14) ØXÚØY (из (2) и (6),
(15) XÚY (из (1) и (7),
(16) ØXÚØY (из (2) и (7),
(17) XÚZ (из (3) и (7),
(18) ØXÚZ (из (4) и (7),
(19) XÚZ (из (1) и (8),
(20) ØYÚZ (из (6) и (8),
(21) ØXÚZ (из (2) и (9),
(22) YÚZ(из (6) и (9),
(23) Z (из (8) и (9),
(24) □ (из (5) и (10).

Мы видим, что порождено много лишних дизъюнктов. Так, 6 и 7– тождественно истинные дизъюнкты. Удаление или добавление тождественно истинного дизъюнкта не влияет на выполнтмость множества дизъюнктов, поэтому такие дизъ.нкты должны быть удалены из вывода. Далее, некоторые дизъюнкты порождаются неоднократно, например, XÚY, ØXÚØY, YÚZ. Это означает, что выбором дизъюнктов для получения резольвенты надо управлять. *Стратегия предпочтения (более коротких дизъюнктов).* Эта стратегия является следующей модификацией предыдущей: сначала в качестве D_2 берется самый короткий дизъюнкт из S_{n-1} (если таких несколько, то они перебираются по порядку), затем более длинные и т.д. Аналогичные условия налагаются и на D_1. Такая стратегия в применении к тому же множеству дизъюнктов S дает следующее:

S_0: (1) XÚY,
(2) ØXÚØY,
(3) XÚZ,
(4) ØXÚZ,
(5) ØZ
S_1: (6) X (из (3) и (5)),
(7) ØX (из (4) и (5)),
(8) YÚØY (из (1) и (2)),
(9) XÚØX (из (1) и (2)),
(10) ØYÚZ (из (2) и (3)),
(11) YÚZ (из (1) и (4)),
(12) Z (из (3) и (4)),
S_2: (13) ØY (из (2) и (6)),
(14) Z (из (2) и (6)),
(15) Y (из (1) и (7)),
(16) Z (из (3) и (7)),
(17) □ (из (6) и (7)).

Вывод оказался короче, чем в предыдущем примере, но попрежнему содержит повторяющиеся и тождественно истинные дизъюнкты. Свободным от этих недостатков является вывод, полученный в соответствие со следующей стратегией.

Стратегия вычеркивания. Введем вначале следующие понятия.

Определение. Дизъюнкт D называется *расширением* дизъюнкта C, если существует подстановка σ такая, что σ(C)ÍD.

Для логики высказываний это означает, что просто D=CÚD′ (при некоторой перестановке литералов). В случае логики предикатов ситуация не столь проста. Например, D=Q(a)ÚP(b,y)ÚR(u) есть расширение дизъюнкта C=P(x,y)ÚQ(z)ÚQ(v).

Стратегия вычеркивания, как и стратегия предпочтения является модификацией стратегии насыщения уровней. Она применяется следующим образом: после того, как получена очередная резольвента D дизъюнктов D_1 и D_2 проверяется, является ли она тождественно истинной формулой или расширением некоторого дизъюнкта C из S_0È...ÈS_{n-1}, и в случае положительного ответа D вычеркивается, т.е. не заносится в проследовательность S_n.

Применение стратегии к прежнему множеству дизъюнктов дает следующее:

S_0: (1) XÚY,
(2) ØXÚØY,
(3) XÚZ,
(4) ØXÚZ,
(5) ØZ,
S_1: (6) ØYÚZ (из (2) и (3)),
(7) YÚZ (из (1) и (4)),
(8) Z (из (3) и (4)),
(9) X (из (3) и (5)),
(10) ØX (из (4) и (5)),
(11) ØY (из (5) и (6)),
(12) Y (из (5) и (7)),
(13) □ (из (5) и (8)).

Рассмотренные стратегии являются *полными* в том смысле, что если множество дизъюнктов S невыполнимо, то из S пользуясь стратегией можно вывести пустой дизъюнкт. Для первых двух стратегий это достаточно очевидно. Полнота стратегии вычеркивания следует из

того, что если D и C–дизъюнкты из S и D–расширение C, то множество S невыполнимо в том и только в том случае, когда невыполнимо множество S|{D}.

11.8. Применение метода резолюций.

Рассмотрим применение метода резолюций **в доказательстве теорем и при планировании действий.**

Доказательство теорем. Применим метод резолюций в доказательстве одной простой теоремы из теории групп.

В качестве исходной возьмем следующую аксиоматику теории групп:

F_1: ("x,y,z)[(xy)z=x(yz)],

F_2: ("x,y)($z)(zx=y),

F_3: ("x,y)($z)(xz=y).

Предположим, что нам надо доказать теорему G: ($x)("y)(yx=y), т.е. что в группе существует правая единица.

Наша задача–установить, что формула G есть логическое следствие формул F_1, F_2, F_3. Прежде, чем решать эту задачу, перейдем к другой сигнатуре. Введем символ трехместного предика P, который интерпретируется следующим образом:

P(x,y,z) означает, что xy=z.

В новой сигнатуре формулы F_1, F_2, F_3 и G запишутся так:

F_1'=("x,y,z)[(x,y,u)&H(y,,z,v)&P(x,v,w)®P(x,z,w)],

F_2'=("x,y)($z)P(z,x,y),

F_3'=("x,y)($z)P(x,z,y),

G'=($x)("y)P(y,x,y).

Сформулируем множество T={F_1,F_2,F_3,ØG}, каждую из формул этого множества приведем к сколемовской нормальной форме и удалим кванторы общности (конъюнкция в сколемовских нормальных формах не появится). Получим множество дизъюнктов D_1,D_2,D_3,D_4:

D_1=ØP(x,y,u)ÚØP(y,z,v)ÚØP(x,v,w)ÚP(u,z,w),

D_2=P(f(x,y),x,y),

D_3=P(x,g(x,y)y),

D_4=ØP(h(x),x,h(x)).

Построим вывод пустого дизъюнкта из множества дизъюнктов D_1,...,D_4. Пусть эти дизъюнкты–первые дизъюнкты вывода. Заменим переменные в дизъюнкте D_2, получим дизъюнкт D_2'=P(f(x',y'),x',y'). Литералы P(x,y,u) и D_2' унифицируются подстановкой s_1={x=f(x',y'),y= x', u=y'}. Применим правило резолюций к D_1 и D_2 (и указанным литералам), получим дизъюнкт

D_5="P(x',,z,v)ÚØP(f(x',y')v,w)ÚP(y',z,w).

Далее, литерал P(f(x',y'),v,w) и D_2 унифицируются подстановкой s_2={x'=x,y'=y,v=x,w=y}. Правило резолюций, примененное к D_1 и D_2, дает дизъюнкт

D_6=ØP(x,z,x)ÚP(y,z,y).

Резольвентой дизъюнктов D_3 и D_6 будет дизъюнкт

D_7=P(y,g(y',y'),y).

(Для получения этой резольвенты заменим переменные в D_3, получим D_3=P(x',g(x',y'),y') и используем подстановку s_3={x=y',z=g(y',y')}. Наконец, из дизъюнктов D_4 и D_7 с помощью подстановки s_4={y=h(g(y',y')), x=g(y',y')} получаем пустой дизъюнкт.

Планирование действий. Отметим вначале одно свойство метода резюлюций. Пусть сигнатура t состоит из двух символов двухместных предикатов P и Q, которые интерпретируются следующим образом: P(x,y) означает, что x–сын y, Q(x,z) означает, что x–внук z.Рассмотрим формулы:

F_1=("x,y,z)[P(x.y)&P(y,x)®Q(x,z)],

F_2=("x)($y)P(x,y),

G=("x)($z)Q(x,z),

смысл которых достаточно ясен.

Используя метод резолюцийц, покажем, что G есть логическое следствие F_1 и F_2. Приведем формулы F_1,F_2 и ØG к сколемовской нормальной форме, получим дизъюнкты:

D_1=ØP(x,y)ÚØP(y,z)ÚQ(x,z),

D_2=P(x,f(x)),

D_3=ØQ(a,z).

Вывод пустого дизъюнкта получается довольно просто:

D_4=ØP(a,y)ÚØP(y,z) ((D_1 D_3, {x=a}),

D_5=ØP(f(a),z) (D_2 D_4 {x=a, y=f(a)}),
D_6=□ (D_2 D_5, {x=f(a),z=f(a))}.

Подстановка z=f(f(a)) означает, что дед элемента a есть отец элемента a. Таким образом, метод резолюций не только устанавливает факт логического следствия формулы G из формул F_1 и F_2, но еще и «подсказывает», как по данному x получить z такой, чтобы формула Q(x,z) была истинна.

Довольно часто интересующая нас переменная участвует не в одной подстановке, как в этом примере переменная z, а в нескольких. Для того, чтобы тследить все подстановки, в которых может изменить значение переменная, к формуле ØG добавляют литерал ответа ANS(z) и заканчивают вывод не пустым дизъюнктом, а литералом ответа.

В качестве примера использования метода резолюций в задачах планирования действий рассмотрим известную в теории искусственного интеллекта задачу об обезъяне и бананах. В задаче говорится об обезъяне, которая хочет съесть бананы, подвешенные к потолку комнаты. Рост обезъяны недостаточен, чтобы достать бананы. Однако в комнате есть стул, встав на который обезъяна может достать бананы. Какие ей надо совершить действия, чтобы достать бананы?

Задачу формализуем следующим образом. Комнату с находящимися в ней обезъяной, стулом и бананами будем называть *предметной областью*. Конкретной местонахождение в комнате обезъяны, стула и бананов будем называть *состоянием предметной области.* Рассмотрим два предиката P(x,y,z,s) R(z). Пусть

P(x,y,z,s) означает, что в состоянии s обезъяна находится в точке x, стул – в y, бананы – в z,
R(s) означает, что в состоянии s обезъяна взяла бананы.

Возможности обезъяны формализуем следующим образом. Введем три функции, которые принимают значения в множестве состояний:

ИДТИ(x,y,s) – состояние, которое получится из s, если обезъяна из точки x перешла в y,
НЕСТИ(x,y,s) – состояние, которое получится из s, если обезъяна перенесла стул из точки x в y,
БРАТЬ(s) – состояние, которое получится из s, если обезъяна взяла бананы.

Условия задачи запишутся в виде следующих формул:

F_1=("x,y,z,s))[P(x,y,z,s)®P(u,y,z, (x,,,u,s))],

F_2=("x,z,s)[P(x,x,z,s)®P(u,u,s, (x,u,s)],

F_3=("x)[P(x,x,x,s)®R((s))].

Пусть в начальном состоянии s_0 обезъяна находилась в точке a, стул–в точке b, бананы–в точке c. Следовательно, к написанным формулам надо ддобавить формулу

F_4=P(a,b,c,s_0).

Надо показать, что формула G=($s)R(s) есть логическое следствие формул F_1,F_2,F_3,F_4. Из множества формул F_1,F_2,F_3,F_4,ØG получим множество дизъюнктов D_1–D_5 (к дизъюнкту, полученному из ØG добавлен литерал ответа ANS(s)):

D_1=ØP(x,y,z,s)ÚP(u,y,z,ИДТИ(x,u,s)),

D_2=ØP(x,x,z,s,)ÚP(u,u,z,НЕСТИ(x,u,s)),

D_3=ØP(x,x,x,s)ÚR(БРАТЬ(s)),

D_4=P(a,b,c,s_0),

D_5=ØR(s)ÚANS(s).

Последовательность дизъюнктов D_1–D_5 продолжаем до вывода литерала ответа:

D6=ØP(x,x,x,s)ÚANS(БРАТЬ(s)) (из D_3 D_5),
D_7=ØP(x,x,u,s)ÚANS(БРАТЬ(НЕСТИ(x,u,s))) (из D_2 и D_6),
D_8=ØP(x,y,z,s)ÚANS(БРАТЬ(НЕСТИ((y,z,ИДТИ(x,y,s)))) (из D_1 и D_7),
D_9=ANS(БРАТЬ(НЕСТИ(b,c,ИДТИ((a,b,s_0)))) (из D_4 и D_8).

Итак, для того, чтобы обезъяне взять бананы, надо сначала из точки a идти в точку b, затем из точки b нести стул в точку c и в точке c, встав на стул, взять бананы.

11.9. Метод резолюций и логическое программирование

Наиболее распространенная технология решения задач на компьютере состоит в том, что в начале программист должен разработать алгоритм решения задачи, а затем записать его на определенном формальном языке. Эти этапы вместе с последующей отладкой требуют от программиста значительных затрат времени и достаточно высокой квалификации. На большинстве этапов решения задачи имеется сильная зависимость от внутренних механизмов компьютера, на котором это решение будет осуществлено. Отмеченные недостатки (а также и некоторые другие) алгоритмической технологии программирования стимулировали поиск новых возможностей. Осознание того, что вычисление – частный случай логического вывода, а алгоритм – формальное задание функции привело к идее **логического программирования**. **Суть этой идеи состоит в том, чтобы компьютеру предлагать не алгоритмы, а описания предметной области задачи и саму задачу в виде некоторой аксиоматической системы, а решение задачи в виде вывода такой системы**. От программиста при таком подходе требуется описать с достаточной степенью полноты предметную область и формулировку задачи на языке этой системы, а поиск вывода, приводящего к решению задачи, поручается компьютеру.

Конечно, при таком широком понимании логического программирования, любая программная система, поддерживающая ту или иную логическую модель, представляет собой фактически и систему логического программирования. Разница может возникнуть лишь тогда, когда в инструментальной системе программист определяет в существенной степени способ обработки знаний. На самом деле логическое программирование понимается, как правило, в более узком смысле.

Логическая программа представляет собой совокупность формулы логики предикатов одного из следующих видов

(1) $p(t_1,...,t_k)$.,

(2) $q(s_1,...,s_1)$:-$q_1(s_1,...,s_1),...,q_n(s_1,...,s_1)$.,

где $P(t_1,...,t_k)$, $q(s_1,...,s_l)$, $q_1(s_1,...,s_l),...$, $q_n(s_1,...,s_l)$ – атомарные формулы логики первого порядка, буквы t и s с индексами – термы. Синтаксис языка логического программирования требует, чтобы в конце каждого выражения ставилась точка. Формулы первого вида называются *фактами*, а второго *правилами.* Формула $q(s_1,...,s_l)$ называется заголовком правила (2). Выполнение программы тнтцтализируется запросом – формулой вида

(3) $r_1(u_1,...,u_m),...,r_{n0}(u_1,...,u_m)$.,

где $r_j(u_1,...,u_m)$ (1£j£n) – атомарные логики первого порядка, буквы u с индексами – термы.

Мы описали синтаксис основных конструкций логического программирования. Семантика обычно представляется в двух видах – логическая семантика и процедурная семантика.

Введем сначала логическую семантику. Каждому факту (1) поставим в соответствие формулу вида

(1/) F=("x*)$p(t_1,...,t_k)$,

где кванторы общности навешаны на все переменные атомарной формулы (1). (Кроме переменных в термах могут быть, разумеется константы.) Правилу (2) аоставим в соответствие формулу вида

(2/) G=("x*)[$q_1(s_1,...,s_l)$ &...& $q_n(s_1,...,s_l)$®$q(s_1,...,s_l)$],

где кванторы общности, как и выше, навешаны на все переменные. Запрос (3) получит в соответствие формулу

(3/) H=($x*)[$r_1(u_1,...,u_m)$ &...& $r_n(u_1,...,u_m)$],

где квантор существования связывает все переменные. Пусть $F_1,...,F_a$ – формулы, соответствующие всем фактам, $G_1,...,G_b$ – всем правилам. Тогда значение пары (программа, запрос) в логической семантике есть утверждение о том, что формула H есть логическое следствие формул $F_1,...,,F_a$, $G_1,...,G_b$.

Операционная семантика – действия компьютера при ответе на запрос. Введем ее на примере следующей программы.

(1) r(a,b).,

(2) q(b,g(c)).,

(3) p(x,f(y)):–r(x,z), q(z,f(y)).,

(4) p(x,f(y)):–r(x,z) q(z,g(y)).,

(5) r(x,z):–q(f(x),g(z)).

(Здесь a,b,c,d – константы, x,y,z – переменные.) Номера в скобках не являются синтаксической конструкцией логического программирования, они проставлены для удобства ссылок.

Предположим, что запрос есть (6)
p(u,f(v)).

При вычислении ответа на этот запрос, интерпретатор формулирует цель r(u,f(v)) и пытается достичь ее, унифицируя цель с фактами. В

нашем случае цель r(u,f(v)) не унифицируется ни с одним из фактов. Тогда интерпретатор пытается ее унифицировать с заголовком одного из правил. Это можно сделать с заголовком правила (4) с помощью подстановки s=(u=x,v=y). Запрос (6) принимает следующий вид

(6/) r(x,z),q(z,f(y))

и формируется цель r(x,z). Она достигается унификацией с первым фактом подстановкой s_1=(x=a,z=b) и интерпретатор пытается достичь цели q(b,f(y)), но эта цель не унифицируется ни с одним из фактов, ни с заголовками правил. Следовательно цель q(b,f(y)) недостижима и происходит возврат к запросу (6/) и цели r(x,z). Делается попытка достичь этой цели при помощи правила (5), но эта попытка так же неудачна. Происходит возврат к запросу (6) и цели r(u,f(v)). (См. рис.4.9, где цели подчеркнуты.)

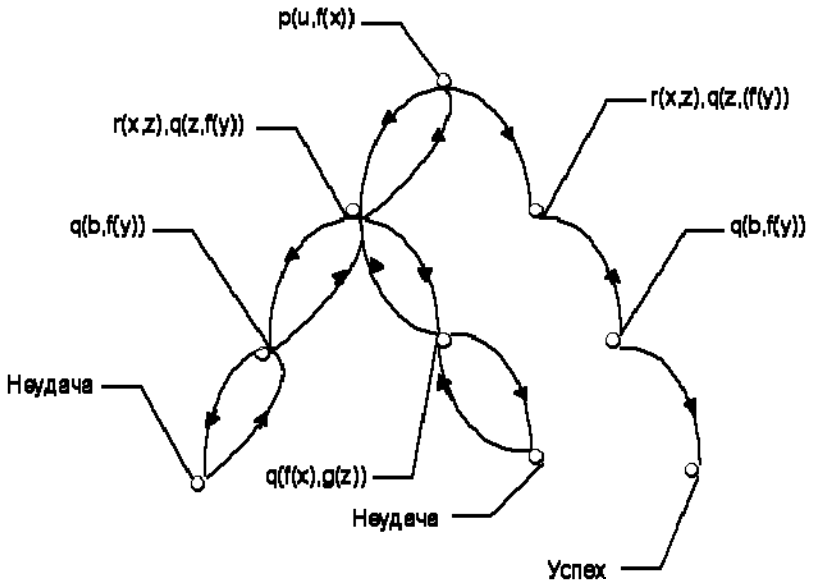


Рис. 4.9

Правило (3) «отработано». Интерпретатор унифицирует цель с заголовком правила (4) той же подстановкой s. Запрос принимает вид

(6$^{//}$) r(x,z),q(z,g(y)),

а целью становится r(x,z). Цель унифицируется с первым фактом подстановкой s_1 и ставится новая цель q(b,g(y)), которая унифицируется со вторым фактом подстановкой s_2=(y=c). Исходная цель оказалась достижимой при результирующей подстановке s_3=(u=a,v=c). Интерпретатор при этом может закончить работу и выдать подстановку s_3. Возможен и другой режим работы интерпретатора, при котором он пытается найти все подстановки, ведущие к достижению цели.

Убедимся в том, что интерпретатор при поиске ответа на запрос фактически строит вывод с помощью правила резолюций.

Логическая семантика программе (1) – 5 ставит в соответствие следующие формулы:

F_1=r(a,b),

F_2=q(b,g(c)),

G_1=("x,y,z)[r(x,z)&q(z,f(y))®p(x,f(y))],

G_2=("x,y,z)[r(x,z)&q(z,g(y))®p(x,f(y))],

G_3=("x,z)[q(f(x),g(z))®r(x,z)],

а запросу (6) – формулу

H=($u,v)p(u,f(v)).

Эта семантика, напомним, состоит в том, что H есть логическое следствие множества формул {F_1,F_2,G_1,G_2,G_3}.

Как будет применяться в этой ситуации метод резолюций? Вначале будет составлено множество формул T={F_1,F_2,G_1,G_2,G_3,ØH}. Затем каждая из формул множества T будет приведена к сколемовской нормальной форме, из которой будет получено множество дизъюнктов S. В нашем случае множество S состоит из дизъюнктов D_1 – D_6:

D_1=r(a,b),

D_2=q(b,g(c)),

D_3=Ør(x,z)&Øq(z,f(g)Úp(x,f(y),

D_4=Ør(x,z)ÚØq(z,g(y))Úp(x,f(y)),

D_5==Øq(f(x),g(z))Úr(x,z),

D_6=Øp(u,f(v)).

Отметим, что дизъюнкт D_6 получился из формулы ØH.

Вывод пустого дизъюнкта будем осуществлять в соответствии с процедурной семантикой. Правило резолюций будет применяться так, что одним из исходных (для правила) дизъюнктов будет дизъюнкт D_6 или его потомки. Попытка получить пустой дизъюнкт за один шаг, т.е. правило резолюций применить к паре {D_1,D_6} или {D_2,D_6}, не приводит к успеху. Применим тогда правило резолюций к паре {D_3,D_6}, получим резольвенту

D_7=Ør(x,z)ÚØq(z,f(y))

с помощью подстановки s=(u=x,v=y). Попробуем теперь в D_7 «уничтожить» литерал Ør(x,z). Это можно сделать с помощью дизъюнктов D_1 и D_5. Резольвентой дизъюнктов D_1 и D_7 будет дизъюнкт

D_8=Øq(b,f(y)).

Единственный литерал, которого нельзя «уничтожить» ни одним из дизъюнктов D_1 – D_5. То же самое можно сказать и о резольвенте дизъюнктов D_5 и D_7.

Мы фактически доказали, что из множества дизъюнктов $\{D_1,...,D_5,D_7\}$ пустой дизъюнкт невыполним (см. пути в графе на рис. 4.9, приводящие к неудаче). Возьмем в таком случае резольвенту дизъюнктов D_4 и D_6:

D_9=Ør(x,z)ÚØq(z,g(y)).

Литерал Ør(x,z) дизъюнкта D_9 «уничтожили» с помощью D_1 и подстановки s_1=(x=a,z=b), получили дизъюнкт

D_{10}=Øq(b,g(y)).

Резольвента дизъюнктов D_2 и D_{10} дает пустой дизъюнкт, при этом используется подстановка s_2=(y=c). Мы получили вывод пустого дизъюнкта из множества дизъюнктов $\{D_1,...,D_6\}$ (см. путь в графе на рис. 4.9, приводящий к успеху).

Итак, интерпретатор при поиске ответа на запрос строит резолютивный вывод.

Свойства и основные результаты

Логика первого порядка обладает рядом полезных свойств, которые делают её очень привлекательной в качестве основного инструмента формализации математики. Главными из них являются:

- полнота (это означает, что для любой замкнутой формулы выводима либо она сама, либо её отрицание);
- непротиворечивость (ни одна формула не может быть выведена одновременно со своим отрицанием).

При этом если непротиворечивость более или менее очевидна, то полнота — нетривиальный результат, полученный Гёделем в 1930 году

(теорема Гёделя о полноте). По сути теорема Гёделя устанавливает фундаментальную эквивалентность понятий *доказуемости* и *общезначимости*.

Логика первого порядка обладает свойством компактности: если некоторое множество формул не выполнимо, то невыполнимо также некоторое его конечное подмножество.

Согласно теореме Лёвенгейма — Скулема если множество формул имеет модель, то оно также имеет модель не более чем счётной мощности. С этой теоремой связан парадокс Скулема, который, однако, является лишь мнимым парадоксом.

Использование

Логика первого порядка как формальная модель рассуждений

Являясь формализованным аналогом обычной логики, **логика первого порядка** даёт возможность строго рассуждать об истинности и ложности утверждений и об их взаимосвязи, в частности, о логическом следовании одного утверждения из другого, или, например, об их эквивалентности. Рассмотрим классический пример формализации утверждений естественного языка в **логике первого порядка**.

Возьмем рассуждение «Каждый человек смертен. Конфуций — человек. Следовательно, Конфуций смертен». Обозначим «x есть человек» через **ЧЕЛОВЕК**(x) и «x смертен» через **СМЕРТЕН**(x). Тогда утверждение «каждый человек смертен» может быть

представлено формулой: $\forall x$(**ЧЕЛОВЕК**(x) → **СМЕРТЕН**(x))
утверждение «Конфуций — человек» формулой **ЧЕЛОВЕК**(**Конфуций**), и «Конфуций смертен» формулой **СМЕРТЕН**(**Конфуций**). Утверждение в целом теперь может быть записано формулой

($\forall x$(**ЧЕЛОВЕК**(x) → **СМЕРТЕН**(x))
ЧЕЛОВЕК(**Конфуций**)) → **СМЕРТЕН**(**Конфуций**)

12. Логика второго порядка

Логика второго порядка в математической логике — формальная система, расширяющая логику первого порядка возможностью квантификации общности и существования не только над переменными, но и над предикатами. Логика второго порядка несводима к логике первого порядка. В свою очередь, она расширяется логикой высших порядков и теорией типов.

Язык и синтаксис

Формальные языки логики второго порядка строятся на основе множества функциональных символов $\mathfrak{F}$ и множества предикатных символов $\mathfrak{P}$. С каждым функциональным и предикатным символом связана арность (число аргументов). Также используются дополнительные символы

- Символы индивидуальных переменных, обычно $x, y, z, x_1, y_1, z_1, x_2, y_2, z_2$ и т. д.
- Символы функциональных переменных F, G, H, F_1, G_1, H_1, F_2, G_2, H_2. Каждой функциональной переменной соответствует некоторое положительное число — арность функции.
- Символы предикатных переменных P, R, S, P_1, R_1, S_1, P_2, R_2, S_2. Каждой предикатной переменной соответствует некоторое положительное число — арность предиката.
- Пропозициональные связи: $\wedge$, $\vee$, $\neg$, $\rightarrow$,
- Кванторы общности $\forall$ и существования $\exists$,
- Служебные символы: скобки и запятая.

Перечисленные символы вместе с символами $\mathfrak{F}$ и $\mathfrak{P}$ образуют алфавит логики первого порядка. Более сложные конструкции определяются индуктивно.

- Терм — это символ индивидуальной переменной, либо выражение которое имеет вид $f(t_1...,t_n)$, где f — функциональный символ арности n, а $t_1...,t_n$ — термы либо выражение вида $F(t_1...,t_n)$, где F — функциональная переменная арности n, а $t_1...,t_n$ — термы.

- Атом — имеет вид $p(t_1...,t_n)$, где p — предикатный символ арности n, а $t_1...,t_n$ — термы или $P(t_1...,t_n)$, где P — предикатная переменная арности n, а $t_1...,t_n$ — термы.
- Формула — это или атом или одна из следующих конструкций: $\neg$ A, ($A_1 \vee A_2$), ($A_1 \wedge A_2$), ($A_1 \rightarrow A_2$), $\forall xA$, $\exists xA$, $\forall FA$, $\exists FA$, $\forall PA$, $\exists PA$, где A, A_1, A_2 — формулы, а x, F, P — индивидуальная, функциональная и предикатная переменные.

Семантика

В классической логике интерпретация формул логики второго порядка задаётся на модели второго порядка, которая определяется следующими данными.

- Базовое множество $\mathfrak{D}$,
- Семантическая функция σ, которая отображает
  - каждый n-арный функциональный символ f из $\mathfrak{F}$ в n-арную функцию $\sigma(f): \mathfrak{D} \times ... \times \mathfrak{D} \rightarrow \mathfrak{D}$,
  - каждый n-арный предикатный символ p из $\mathfrak{P}$ в n-арное отношение $\sigma(p) \subseteq \mathfrak{D} \times ... \times \mathfrak{D}$.

Свойства

В отличие от логики первого порядка, логика второго порядка не имеет свойств полноты и компактности. Также в этой логике является неверным утверждение теоремы Лёвенгейма — Скулема.

Представляет интерес вопрос о том, может ли логика второго порядка служить в качестве основания математики. Вопрос этот особенно интересен в той связи, что одно из самых важных направлений в основаниях математики — структурализм — напрямую связано с логикой второго порядка, и основательность претензий структурализма на то, чтобы быть адекватной философией математики, напрямую зависит от того, является ли логика второго порядка адекватным орудием исследования оснований математики.

Прежде всего следует отметить, что различие между экстенсиональными и интенсиональными сущностями не является

решающим для разделения языков на порядки. Хотя интерпретированный язык называется «второпорядковым» или «высшего порядка», если его переменные пробегают над отношениями, пропозициональными функциями, свойствами, классами или множествами сущностей, над которыми пробегают переменные логики первого порядка, природа сущностей не имеет значения для характеристики логики второго порядка.

Отличительной характеристикой порядка языка является размер области квантификации. Язык второго порядка может быть получен добавлением к дедуктивной системе первого порядка расширения аксиом с кванторами типа [Х{Ф{Х)) —> Ф(У)) и аксиомной схемы свертывания БХ Ух (Хх = Ф(х)) для каждой формулы Ф, не содержащей ^"свободно. Но основное различие языков первого и второго порядков заключается в семантике. Областью переменных первого порядка могут быть натуральные числа.

В стандартной семантике логики второго порядка переменные пробегают над совокупностью всех подмножеств области. Хотя для такой семантики все доказуемые утверждения являются истинными, в ней не проходят полнота и компактность. Все бесконечные структуры категоричны. Преимуществами логики второго порядка со стандартной семантикой являются ясность, соответствие интуиции. При фиксации области квантификации переменных первого порядка понятия «все свойства» или «все подмножества», играющие решающую роль в основаниях математики, обретают твердый смысл. По этой причине многие считают, что логика второго порядка является более подходящим кандидатом для оснований математики, чем традиционный базис — логика первого порядка плюс теория множеств. А вот сторонники логики первого порядка утверждают, что понятие переменной, пробегающей над всеми свойствами фиксированной области, в высшей степени неясно и по этой причине логика второго порядка вряд ли может считаться хорошим базисом. Для разрешения этого спора требуется рассмотрение дополнительных аргументов в пользу логики второго порядка.

Во-первых, нужно понять, в каком смысле логика второго порядка является логикой. В общепринятом смысле слова логика абстрагируется от специфики рассматриваемого предмета, и представляет собой теорию о том, что истинно и что ложно. Если руководствоваться именно таким критерием, тогда под рубрику логики

определенно попадают истинностные функции. Но вот введение кванторов усложняет критерий логики. Действительно, уже логика первого порядка требует кроме истины и лжи два дополнительных понятия — истинности предиката об объекте и самого объекта. В этом отношении логика второго порядка кажется даже более экономной, поскольку там требуется только еще одно дополнительное понятие — значение предикатной переменной. Таким образом, с точки зрения приведенного критерия логика второго порядка определенно является логикой не в меньшей степени, чем логика первого порядка. Действительно, раз уже в логике первого порядка допущено некоторое дополнение, нет никаких априорных возражений против дальнейших допущений и усложнений.

Однако в качестве главного возражения против логики второго порядка выдвигается обвинение в том, что упомянутое выше дополнение выводит эту самую логику из семейства логик вообще. Знаменитый (как и все афоризмы Куайна) афоризм «Логика второго порядка есть волк (теория множеств) в овечьей (логика) шкуре» является скорее метафоричным, нежели точным выражением ситуации, сложившейся в отношении логики второго порядка. И поэтому буквальное толкование афоризма попросту неверно. Как отмечает, среди прочих, Дж. Аззуни, логика второго порядка не является, строго говоря, теорией множеств, если под последней подразумевается теория множеств первого порядка, поскольку модели для этих систем не совпадают. Действительно, как известно, теория множеств первого порядка имеет нестандартные модели, в которых «є» не является отношением членства. Различие между логикой первого порядка и логикой второго порядка проявляется, если прибегнуть к терминологии Куайна, в онтологическом плане, имея в виду его критерий существования «Быть значит быть значением переменной». В случае одной переменной предикация Ра свойства Р сингулярному термину а не предполагает, что выражение Ра имеет в качестве онтологических допущений объем предиката Р. Онтологические допущения, как следует из критерия существования, определяются областью квантификации. Именно по этой причине отношение членства «є » не предполагается в использовании нотации «Ра». Но как только мы переходим к квантификации предиката, предикация становится равносильной отношению членства, и больше того, по сути становится логической константой. В определенном смысле логика второго порядка действительно становится равносильной теории множеств.

Дело в том, что понятие стандартной модели для языка второго порядка предполагает, что переменные для одноместного предиката должны пробегать над всеми совокупностями объектов в универсуме индивидов. Это предположение существенно, поскольку логика второго порядка не является рекурсивно аксиоматизируемой. Что касается рекурсивно аксиоматизируемых фрагментов, они будут представлять собой класс обобщенных моделей Генкина таких, что теоремы фрагмента и только они истинны для всех моделей этого класса. Но эти модели не являются стандартными. Если бы мы были ограничены рекурсивно аксиоматизируемыми системами, мы не могли бы сказать, что модели для оценки логической истины являются стандартными. Но это можно сделать, если синтаксическую операцию предикации представить в виде отношения членства (в виде логической константы) и позволить предикатной переменной пробегать над всеми подмножествами универсума индивидов. Таким образом, выражение 3F (Fa) вынуждает нас считать одним и тем же значения, которые принимает предикат, и указание сингулярным термином a. Суть аргументации Аззуни состоит в том, что логика второго порядка оказывается если не теорией множеств, то двухсортной первопорядковой теорией объектов и их классов.

Теоретико-множественные модели языков второго порядка и интерпретации этих языков приписывают значениям предикатных переменных сущности одного сорта.

Так где же лежит различие между логиками первого порядка и второго порядка? Д. Восток замечает, что сам по себе факт, что в одной теории пишется Fa, а в другой теории — a є F, означает лишь различие в нотации, что имеет место, скажем, в случае стандартной и польской нотации. Две логики могли бы различаться в отношении того, что считать правильно построенной формулой. Действительно, в логике второго порядка такое соединение символов как ab или FG не считалось бы правильно построенной формулой, в то время как в первопорядковой теории множеств b є a или G є F являются правильно построенными формулами. Правда, во избежание парадоксов такие формулы можно признать ложными, а то и вовсе отказать им в статусе правильно построенных формул. Так что различие между логикой первого порядка и логикой второго порядка лежит в их семантиках.

В частности, в двух этих случаях по-разному работает понятие общезначимости. Формальное определение общезначимости звучит

одинаково для обеих логик: формула общезначима, если и только если, она истинна при всех интерпретациях входящих в нее символов. В это определение входит намеренная интерпретация логических связок. В логике первого порядка вводится специальная интерпретация кванторов, при которой они пробегают над всеми объектами. В логике второго порядка в дополнение постулируется специальная интерпретация для кванторов второго уровня, т.е. предикатных кванторов, которая обобщает все способы, которыми предикат может быть истинен для одних объектов, и ложен для других. Именно это дополнительное условие и создает различие между двумя логиками. Оно ответственно за то, что в то время как логика первого порядка может быть снабжена полным множеством аксиом и правил, логика второго порядка лишена этого. Кроме того, этот же факт ответствен за то, что многие важные понятия в логике первого порядка не определимы, и в то же время определимы в логике второго порядка.

Приведем иллюстрацию этого обстоятельства. Известно, что тождество неопределимо в языке первого порядка, т.е. если этот язык имеет нормальные модели, в которых «=» интерпретируется как тождество, тогда он также имеет и «ненормальные» модели, в которых «=» интерпретируется другим образом. В логике второго порядка мы имеем определение a = b <-> VF (Fa <-> Fb), которое интуитивно правдоподобно. Таким образом, дополнительный вид квантификации в логике второго порядка придает ей большие выразительные возможности, но в то же время лишает ее компактности или конечной аксиоматизируемости. Как уже было сказано, те же выразительные возможности равносильны теории классов, представленной двухсортной теорией. Различие только лишь в онтологических допущениях, поскольку многие полагают логику второго порядка нейтральной в отношении онтологии, считая, что она избегает допущения классов как сущностей.

При сопоставлении логик первого порядка и второго порядка следует обратить внимание на то, в какой степени они естественны. Уже говорилось о том, что сторонники «регламентации обыденного языка» логикой первого порядка (например, Куайн) считают, что она покрывает огромную область структур обыденного языка. И только некоторые «тонкости», которые не по плечу логике первого порядка, требуют логики второго порядка. Но вот подобного рода «тонкости» играют огромную роль, скажем, в математике. Дедекинд показал, что постулаты элементарной арифметики натуральных чисел дают

категоричные структуры (все модели этих постулатов имеют одну и ту же структуру, т.е. структуру натуральных чисел). То же сделал Кантор для действительных чисел. Но оба этих доказательства опираются на второпорядковое понимание этих постулатов, потому что известно, что никакое множество первопорядковых постулатов, которые имеют бесконечные модели, не может быть категорическим. Так что приемлемость этих доказательств ведет к признанию необходимости логики второго порядка.

Другой пример необходимости логики второго порядка таков. Пусть имеется бесконечное число посылок типа

А не есть родитель В

А не есть родитель родителя В

А не есть родитель родителя родителя В

Из этих посылок следует заключение А не есть предок В.

Такой вывод вполне значим для тех, кто понимает содержание понятие «предок». В логике первого порядка этот вывод не является значимым, поскольку логика первого порядка компактна. Компактность означает, что если утверждение следует из бесконечного множества посылок, тогда оно должно следовать из конечного подмножества множества этих посылок.

Одна из главных причин спора между сторонниками логики первого порядка и логики второго порядка заключается в том, возможно ли достижение категоричности, поскольку именно это свойство является главным признаком соответствия интуиции: формальная система должна описывать объективный мир математических сущностей. Логика второго порядка категорична, но, как уже было указано, многие полагают понятие переменной над всеми свойствами формально неточным. Однако вполне возможно сделать понятие переменной над всеми свойствами формальным, поскольку возможно построение некоторой версии аксиоматической теории, достаточной для формулировки стандартной семантики логики второго порядка. В такой версии можно доказать категоричность теории и многие существенные результаты теории множеств. Но формальная

метатеория, в которой получается категоричность, сама может считаться теорией первого порядка. Следовательно, в ней проходит теорема Левенгейма —Сколема о нестандартных моделях, и категоричности больше нет. В пользу этого взгляда можно привести все те аргументы, которые традиционно приводятся при рассмотрении релятивизма в теории множеств, согласно которому одно и то же множество может иметь различную мощность в различных формальных системах.

Против этого сторонники логики второго порядка высказывают твердое убеждение, что метатеория, о которой идет речь, не может рассматриваться как неинтерпретированная теория с различными возможными моделями. Намеренная интерпретация этой метатеории — интуитивная семантика естественных языков, в которых формулируются содержательные истины математики. На первый план выступает область таких языков, а не понятие модели. Категоричность относится к естественному языку, а не к изоморфизму моделей в каждой интерпретации.

13. Комбинаторная логика

Не следует путать с комбинационной логикой — способом соединения логических элементов в цифровых устройствах.

Комбина́торная ло́гика — направление математической логики, занимающееся фундаментальными (то есть не нуждающимися в объяснении и не анализируемыми) понятиями и методами формальных логических систем или исчислений. В дискретной математике комбинаторная логика тесно связана с лямбда-исчислением, так как описывает вычислительные процессы.

С момента своего возникновения комбинаторная логика и лямбда-исчисление были отнесены к *неклассическим логикам*. Дело заключается в том, что комбинаторная логика возникла в 1920-х годах, а лямбда-исчисление — в 1940-х годах как ветвь метаматематики с достаточно очерченным предназначением — дать основания математике. Это означает, что сконструировав требуемую «прикладную» математическую теорию — предметную теорию, — которая отражает процессы или явления в реальной внешней среде, можно воспользоваться «чистой» метатеорией как оболочкой для

выяснения возможностей и свойств предметной теории. Вскоре также оказалось, что обе эти системы можно рассматривать как языки программирования (см. также комбинаторное программирование).

К настоящему времени оба эти языка не только стали основой для всей массы исследований в области информатики, но и широко используются в теории программирования. Рост вычислительной мощности компьютеров привел к автоматизации значительной части теоретического (логического и математического) знания, а комбинаторная логика вместе с лямбда-исчислением признаются основой для рассуждений в терминах объектов

13.1. Основные понятия

В комбинаторной логике в качестве основных понятий выступает *одноместная функция* и операция применения функции к аргументу (*аппликация*). Понятие функции принимается первичным по отношению к понятию множества. Функция понимается обобщённо и может оперировать с объектами равного ей уровня в качестве аргументов и значений. Так как аргументом функции может быть функция, многоместную функцию можно свести к одноместным.

Комбинатором называется функция *f*, которая удовлетворяет равенству:

$$fg_1g_2\ldots g_n=(\ldots((fg_1)g_2)\ldots g_n)=X.$$

где $g_1, g_2, \ldots, g_n$ ($n\geq 1$) — некоторые функции, X — объект, построенный из функций применением аппликации.

Любой комбинатор может быть выражен через два комбинатора S и K, определённые следующими равенствами:

$Sxyz=xz(yz)$ (*распределитель*)
$Kxy=x$ (*вычеркиватель*)

По данному лямбда-выражению можно всегда построить *аппликативное выражение*. Для этого необходимо всего два комбинатора: S и K. В виде лямбда-выражений: $K=\lambda x.\ \lambda y.\ x$,

$S=\lambda f.\ \lambda g.\lambda\ x.\ fx(gx)$.. То есть, комбинаторную логику, определённую на этих комбинаторных объектах можно рассматривать как модель лямбда-исчисления.

Другими примерами комбинаторов (в записи лямбда-исчисления) могут служить *функция тождества*, легко выражаемая через S и K:

$$I=\lambda x.\ X=SKK$$

и *комбинатор неподвижной точки* или *Y-комбинатор*:

$$Y=\lambda h.\ ((\lambda x.\ h(xx)(\ \lambda x.\ h(xx)))$$

В 1920 году комбинаторы как специальные математические сущности первоначально были введены М. Шейнфинкелем. Несколькими годами позже они независимо были переоткрыты Х. Карри, благодаря которому с тех пор были выполнены основные продвижения в комбинаторной логике (хотя другие исследователи, например, Россер, в различное время также участвовали в этой работе). Почти в то же самое время Чёрчем, Россером и Клини было начато развитие λ-конверсии.

С 1970-х гг. комбинаторы использовались в трех главных аспектах: во-первых, для построения логических систем, основанных на абстрактной записи операции; во-вторых, в теории доказательств как основа записи конструктивных функций различного вида; в-третьих, при построении и анализе некоторых языков программирования в компьютерных науках.

13.2. Категориальная комбинаторная логика

Начнем с определения категориа́льной абстра́ктной маши́ны

Категориа́льная абстра́ктная маши́на (КАМ) — это модель вычисления программы, в которой сохраняются особенности аппликативного, функционального либо композиционного стиля. Она опирается на технику аппликативного вычисления.

Один из подходов к реализации функциональных языков дается машиной, основанной на суперкомбинаторах, или SK-машиной Дэвида Тёрнера. Представление о категориальной абстрактной машине даёт альтернативный подход. Строение КАМ включает синтаксическую, семантическую и вычислительную конституэнты. Синтаксис основан на формализме де Брёйна, использование которого позволяет преодолеть трудность, вызываемые применением связанных переменных. Семантика по своим выразительным возможностям аналогична SK-машине. Вычисления выполняются по аналогии тем вычислениям, которые использованы в SECD-машине Лэндина. Занимая такие позиции, категориальная абстрактная предоставляет непротиворечивые основания для синтаксиса, семантики и теории вычислений. Такая интеграция возникает не без влияния функционального стиля программирования.

Концепция категориальной абстрактной машины возникла в середине 1980-х годов и играет роль варианта *теории вычислений для программистов*. С теоретической точки зрения, категориальная абстрактная машина представлена декартово замкнутой категорией и погружена в комбинаторную логику. Машинные инструкции являются объектами-комбинаторами, образуя в совокупности специальный вариант комбинаторной логики — *категориальную* комбинаторную логику. Категориальная абстрактная машина является ясным и математически корректным представлением языков функционального программирования. Используя равенства выражений, машинный код удаётся оптимизировать. Особенно отчётливо проявляют различные механизмы вычислений — рекурсия, ленивые вычисления, — а также механизмы передачи параметров — вызов по имени, вызов по значению и т. п. С теоретической точки зрения категориальная абстрактная машина сохраняет все преимущества объектно-ориентированного подхода к программированию .

Формализм де Брёйна

Формализм де Брёйна — техника переобозначения связанных переменных (формальных параметров), которая позволяет избежать коллизий связывания при замещении формальных параметров на фактические. Он применяется при компилировании программного кода на КАМ. Этот прием переобозначения также носит название *кодирования по де Брейну* и он позволяет, фактически, аппаратом λ-

исчисления пользоваться на тех же самых правах, что и аппаратом комбинаторной логики.

В рамках комбинаторной логики строится специальный вариант теории вычислений, называемый категориальной абстрактной машиной.

Для этого вводится в рассмотрение особый фрагмент комбинаторной логики — категориальная комбинаторная логика. Она представлена набором комбинаторов, каждый из которых имеет самостоятельное значение как инструкция системы программирования. Тем самым в комбинаторную логику встраивается ещё одно полезное приложение — система программирования, основанная на декартово замкнутой категории (д.з.к.). Это позволяет ещё раз на новом уровне переосмыслить связь операторного и аппликативного стиля программирования.

13.3. Иллативная комбинаторная логика

Пользуясь представлениями об объектах как абстрактных математических сущностях, обладающих определенными подстановочными свойствами, можно строить системы *логических рассуждений*. Наиболее известная среди таких систем основана на комбинаторах.

Логика, основанная на комбинаторах, или *иллативная комбинаторная логика*, строится из теории комбинаторов или лямбда-исчисления, расширенного дополнительными константами — *экстра-константами*, — вместе с соответствующими аксиомами и правилами вывода, которые обеспечивают средства дедуктивного вывода.

14. λ-исчисление.

Возможно, у этой системы найдутся приложения не только в роли логического исчисления. (*Алонзо Чёрч, 1932*)

Вообще говоря, лямбда-исчисление не относится к предметам, которые

«должен знать каждый уважающий себя программист». Это теоретическая наука, изучение которой необходимо, когда вы собираетесь заняться исследованием систем типов или хотите создать свой функциональный язык программирования. Тем не менее, если у вас есть желание разобраться в том, что лежит в основе Haskell, ML и им подобных, «сдвинуть точку сборки» на написание кода или просто расширить свой кругозор, то следует ознакомиться с предлагаемым изложением.

Начнём с краткого экскурса в историю. В 30-х годах XX века перед математиками встала так называемая проблема разрешения (*Entscheidungsproblem*), сформулированная Давидом Гильбертом. Суть её в том, что вот есть у нас некий формальный язык, на котором можно написать какое-либо утверждение. Существует ли алгоритм, за конечное число шагов определяющий его истинность или ложность? Ответ был найден двумя великими учёными того времени Алонзо Чёрчем и Аланом Тьюрингом. Они показали (первый — с помощью изобретённого им λ-исчисления, а второй — теории машины Тьюринга), что для арифметики такого алгоритма не существует в принципе, т.е. Entscheidungsproblem в общем случае неразрешима.

Так лямбда-исчисление впервые громко заявило о себе, но ещё пару десятков лет продолжало быть достоянием математической логики. Пока в середине 60-х Питер Ландин не отметил, что сложный язык программирования проще изучать, сформулировав его ядро в виде небольшого базового исчисления, выражающего самые существенные механизмы языка и дополненного набором удобных производных форм, поведение которых можно выразить путем перевода на язык базового исчисления. В качестве такой основы Ландин использовал лямбда-исчисление Чёрча.

14.1. λ-исчисление: основные понятия

Синтаксис

В основе лямбда-исчисления лежит понятие — анонимная функция. В нём нет встроенных констант, элементарных операторов, чисел, арифметических операций, условных выражений, циклов и т. п. —

только функции. Потому что лямбда-исчисление — это не язык программирования, а формальный аппарат, способный определить в своих терминах любую языковую конструкцию или алгоритм. В этом смысле оно созвучно машине Тьюринга, только соответствует функциональной парадигме, а не императивной.

Рассмотрим его наиболее простую форму: чистое нетипизированное лямбда-исчисление.

Термы:

переменная:	`x`
лямбда-абстракция (анонимная функция):	`λx.t`, где `x` — аргумент функции, `t` — её тело.
применение функции (аппликация):	`f x`, где `f` — функция, `x` — подставляемое в неё значение аргумента

Соглашения о приоритете операций:

- Применение функции левоассоциативно. Т.е. `s t u` — это тоже самое, что `(s t) u`
- Аппликация (применение или вызов функции по отношению к заданному значению) забирает себе всё, до чего дотянется. Т.е. `λx. λy. x y x` означает то же самое, что `λx. (λy. ((x y) x))`
- Скобки указывают группировку действий.

Может показаться, будто нам нужны какие-то специальные механизмы для функций с несколькими аргументами, но на самом деле это не так. Действительно, в мире чистого лямбда-исчисления возвращаемое функцией значение тоже может быть функцией. Следовательно, мы можем применить первоначальную функцию только к одному её аргументу, «заморозив» прочие. В результате получим новую функцию

от «хвоста» аргументов, к которой применим предыдущее рассуждение. Такая операция называется *каррированием* (в честь Хаскелла Карри). Выглядеть это так:

`f = λx.λy.t`	Функция с двумя аргументами `x` и `y` и телом `t`
`f v w`	Подставляем в `f` значения `v` и `w`
`(f v) w`	Эта запись аналогична предыдущей, но скобки явно указывают на последовательность подстановки
`((λy.[x → v]t) w)`	Подставим `v` вместо `x`. `[x → v]t` означает «тело `t`, в котором все вхождения `x` заменены на `v`»
`[y → w][x → v]t`	Подставим `w` вместо `y`. Преобразование закончено.

И напоследок несколько слов об *области видимости*. Переменная `x` называется *связанной*, если она находится в теле `t` λ-абстракции `λx.t`. Если же `x` не связана какой-либо вышележащей абстракцией, то её называют *свободной*. Например, вхождения `x` в `x y` и `λy.x y` свободны, а вхождения `x` в `λx.x` и `λz.λx.λy.x(y z)` связаны. В `(λx.x)x` первое вхождение `x` связано, а второе свободно. Если все переменные в терме связаны, то его называют *замкнутым*, или *комбинатором*. Будем использовать следующий простейший комбинатор (*функцию тождества*): `id = λx.x`. Она не выполняет никаких действий, а просто возвращает без изменений свой аргумент.

Процесс вычисления

Рассмотрим следующий терм-применение:

```
(λx.t) y
```

Его левая часть — `(λx.t)` — это функция с одним аргументом `x` и телом `t`. Каждый шаг вычисления будет заключаться в замене всех вхождений переменной `x` внутри `t` на `y`. Терм-применение такого вида носит имя *редекса* (от *reducible expression*, *redex* — «сокращаемое выражение»), а операция переписывания редекса в соответствии с указанным правилом называется *бета-редукцией*.

Существует несколько стратегий выбора редекса для очередного шага

вычисления. Рассматривать их мы будем на примере следующего терма:

```
(λx.x) ((λx.x) (λz. (λx.x) z)),
```

который для простоты можно переписать как

```
id (id (λz. id z))
```

(напомним, что `id` — это функция тождества вида `λx.x`)

В этом терме содержится три редекса:

```
id (id (λz. id z))
id (id (λz. id z))
id (id (λz. id z))
```

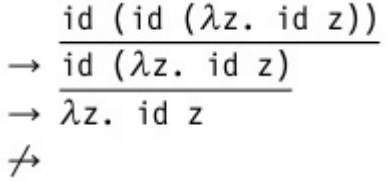
1. **Полная β-редукция**. В этом случае каждый раз редекс внутри вычисляемого терма выбирается произвольным образом. Т.е. наш пример может быть вычислен от внутреннего редекса к внешнему:

```
  id (id (λz. id z))
→ id (id (λz.z))
→ id (λz.z)
→ λz.z
↛
```

2. **Нормальный порядок вычислений**. Первым всегда сокращается самый левый, самый внешний редекс.

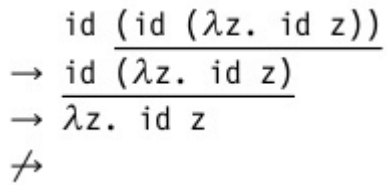
```
  id (id (λz. id z))
→ id (λz. id z)
→ λz. id z
→ λz.z
↛
```

3. **Вызов по имени**. Порядок вычислений в этой стратегии аналогичен предыдущей, но к нему добавляется запрет на проведение сокращений внутри абстракции. Т.е. в нашем примере мы останавливаемся на предпоследнем шаге:



Оптимизированная версия такой стратегии (*вызов по необходимости*) используется Haskell. Это так называемые «ленивые» вычисления.

4. **Вызов по значению**. Здесь сокращение начинается с самого левого (внешнего) редекса, у которого в правой части стоит *значение* — замкнутый терм, который нельзя вычислить далее.



Для чистого лямбда-исчисления таким термом будет λ-абстракция (функция), а в более богатых исчислениях это могут быть константы, строки, списки и т.п. Данная стратегия используется в большинстве языков программирования, когда сначала вычисляются все аргументы, а затем все вместе подставляются в функцию.

Если в терме больше нет редексов, то говорят, что он *вычислен*, или находится в *нормальной форме*. Не каждый терм имеет нормальную форму, например `(λx.xx)(λx.xx)` на каждом шаге вычисления будет порождать самоё себя (здесь первая скобка — анонимная функция, вторая — подставляемое в неё на место `x` значение).

Недостатком стратегии вызова по значению является то, что она может зациклиться и не найти существующее нормальное значение терма. Рассмотрим для примера выражение

```
(λx.λy. x) z ((λx.x x)(λx.x x))
```


Этот терм имеет нормальную форму `z` несмотря на то, что его второй аргумент такой формой не обладает. На её-то вычислении и зависнет стратегия вызова по значению, в то время как стратегия вызова по имени начнёт с самого внешнего терма и там определит, что второй аргумент не нужен в принципе. Вывод: если у редекса есть нормальная форма, то «ленивая» стратегия её обязательно найдёт.

Ещё одна тонкость связана с именованием переменных. Например, терм `(λx.λy.x)y` после подстановки вычислится в `λy.y`. Т.е. из-за совпадения имён переменных мы получим функцию тождества там, где её изначально не предполагалось. Действительно, назови мы локальную переменную не `y`, а `z` — первоначальный терм имел бы вид `(λx.λz.x)y` и после редукции выглядел бы как `λz.y`. Для исключения неоднозначностей такого рода надо чётко отслеживать, чтобы все свободные переменные из начального терма после подстановки оставались свободными. С этой целью используют *α-конверсию* — переименование переменной в абстракции с целью исключения конфликтов имён.

Так же бывает, что у нас есть абстракция `λx.t x`, причём `x` свободных вхождений в тело `t` не имеет. В этом случае данное выражение будет эквивалентно просто `t`. Такое преобразование называется *η-конверсией*.

На этом закончим вводную в лямбда-исчисление. Далее мы займёмся программированием на λ-исчислении.

14.2. Булевы константы Чёрча

Как уже говорилось ранее, в чистом бестиповом лямбда-исчислении отсутствует всё, кроме функций. Так что даже такие элементарные вещи, как числа или булевы значения необходимо реализовывать самим. Точнее, надо создать некие активные сущности, которые будут вести себя подобно необходимым нам объектам. И, естественно, процесс кодирования будет заключаться в написании соответствующих функций.

Начнём с самого простого: `True` и `False`. Два терма, воплощающие поведение этих констант, выглядят следующим образом:

`tru = λt.λf.t`	Двухаргументная функция, всегда возвращающая первый аргумент
`fls = λt.λf.f`	Двухаргументная функция, всегда возвращающая второй аргумент

Оператор `if` под такие булевы константы будет имеет вид:
`if = λb. λx. λy.b x y`
Здесь `b` — `tru` или `fls`, `x` — ветка `then`, `y` — ветка `else`.

Это будет работать следующим образом:
`if fls t e`

Поскольку условие `if` ложно (`fls`), то должно возвращаться выражение из ветки `else` (`e` в нашем случае).

`(λb. λx. λy. b x y) fls t e`	по определению `if`
`(λx. λy. fls x y) t e`	редукция подчёркнутого выражения из предыдущей строки
`(λy. fls t y) e`	редукция подчёркнутого выражения из предыдущей строки
`fls t e`	редукция подчёркнутого выражения из предыдущей строки
`(λt.λf. f) t e`	по определению `fls`
`(λf. f) e`	редукция подчёркнутого выражения из предыдущей строки
`e`	редукция подчёркнутого выражения из предыдущей строки

В определении основных булевых операторов также нет ничего сложного. Например, конъюнкция (логическое «и») будет выглядеть так:
`and = λx. λy. x y fls`

`and` получает два булевых значения `x` и `y`. Первым подставляется `x` (каррирование). Если он является `tru` (`tru y fls` после редукции), то вернётся `y`, который затем тоже «проверится» на истинность. Таким образом, итоговое `tru` мы получим только в случае, когда и `x`, и `y` «истинны». Во всех других вариантах ответом будет `fls`.

14.3.Числа Чёрча

Будем кодировать только натуральные числа, для чего вспомним аксиомы Пеано, определяющие их множество. В основе реализации по-прежнему будут лежать функции, ведущие себя в заданном контексте подобно единице, двойке и т.д. Собственно, это одна из особенностей лямбда-исчисления: сущности, записанные в его терминах, не обладают самодостаточностью, поскольку воплощают *поведение* того или иного объекта.

Итак, нам нужна функция, принимающая два аргумента: фиксированное начальное значение и функцию для определения следующего элемента (*функцию следования*). Число будет закодировано в количестве применений функции следования к начальному значению:

`0 ≡ λs.λz. z`	функция `s` применяется к начальному значению `z` нуль раз
`1 ≡ λs.λz. s z`	функция `s` применяется к начальному значению `z` один раз
`2 ≡ λs.λz. s (s z)`	функция `s` применяется к начальному значению `z` два раза
...	и так далее

Легко заметить, что нуль кодируется так же, как и логическое False. Тем не менее, не стоит делать из этого какие-либо далеко идущие выводы: это всего лишь совпадение.

Задача функции следования состоит в том, чтобы прибавить к заданному числу единицу. Т.е. в качестве аргумента она будет принимать значение, которое требуется увеличить, и которое тоже является функцией двух аргументов. Таким образом, суммарно функция `(+1)` имеет три аргумента: предшествующее число Чёрча `n`, функцию, которую надо будет применить `n+1` раз к начальному значению, и само начальное значение. Выглядит это так:
`scc = λn. λs. λz. s (n s z)`
Здесь `n s z` — `n` раз применённая к `z` функция `s`. Но нам нужно применить её `n+1` раз, откуда и берётся явное `s (n s z)`.

Допустим, нам надо получить из «единицы» one=λs.λz.sz «двойку» two = λs.λz. s (s z). Произойдёт это так:

scc one s' z'	s' и z' — чтобы не путать подставляемые значения с именами переменных
(λn. λs. λz. s (n s z)) one s' z'	по определению scc
(λs. λz. s (one s z)) s' z'	редукция подчёркутого выражения из предыдущей строки
(λz. s' (one s' z)) z'	редукция подчёркутого выражения из предыдущей строки
s' (one s' z')	редукция подчёркутого выражения из предыдущей строки
s' ((λs.λz. s z) s' z')	по определению one
s' ((λz. s' z) z')	редукция подчёркутого выражения из предыдущей строки
s' (s' z')	редукция подчёркутого выражения из предыдущей строки
two s' z'	по определению two

14.4. Арифметические операции

Сложение

Функция, осуществляющая сложение двух чисел Чёрча, будет принимать два слагаемых: x и y, которые в свою очередь тоже имеют по два аргумента — s (функцию следования) и z (начальное значение). Сложение будет состоять в том, чтобы сначала применить s к z x раз, а потом ещё y раз.

```
plus = λx. λy. λs. λz. x s (y s z)
```

В качестве примера сложим one = λs.λz. s z и two = λs.λz. s (s z). Ответ должен будет выглядеть так: three = λs.λz. s (s (s z)).

plus one two s' z'	s' и z' — чтобы не путать подставляемые значения с именами переменных
(λx. λy. λs. λz. x s (y s z)) one two s' z'	по определению plus
one s' (two s' z')	после проведения редукции
(λs.λz. s z) s' (two s' z')	по определению one
s' (two s' z')	после проведения редукции
s' ((λs.λz. s (s z) s' z')	по определению two
s' (s' (s' z'))	после проведения редукции
three s' z'	по определению three

Умножение

Функцию для умножения можно определить через функцию сложения. В конце-концов, умножить x на y означает сложить x копий y.

```
times = λx. λy. x (plus y) z
```

Есть ещё один способ определения умножения на числах Чёрча, без использования plus. Его идея заключается в том, что для получения произведения x и y нужно x раз взять y раз применённую к начальному значению функцию s:

```
times' = λx.λy.λs.λz. x (y s) z
```

Для примера умножим two = λs.λz. s (s z) на three = λs.λz. s (s (s z)). Результат должен будет иметь вид: six = λs.λz. s (s (s (s (s (s z))))).

times' two three s' z'	s' и z' — чтобы не путать подставляемые значения с именами переменных
(λx.λy.λs.λz. x (y s)	по определению times'

`z) two three s' z'`	
`two (three s') z'`	после проведения редукции
`(λs.λz. s (s z)) (three s') z'`	по определению `two`
`three s' ((three s') z')`	после проведения редукции
`(λs.λz. s (s (s z))) s' ((three s') z')`	по определению `three`
`s' (s' (s' ((three s') z')))`	после проведения редукции
`s' (s' (s' (((λs.λz. s (s (s z))) s') z')))`	по определению `three`
`s' (s' (s' (((λz. s' (s' (s' z))) z')))`	после проведения редукции
`s' (s' (s' (s' (s' (s' z')))))`	редукция подчёркнутого выражения
`six s' z'`	по определению `six`

Последней нерассмотренной операцией является вычитание — не самая тривиальная вещь на числах Чёрча. Желающие могут изучить её самостоятельно, например, по книге Бенжамина Пирса «Types and Programming Languages».
Цитата для привлечения внимания из вики-конспекта по лямбда-исчислению: «Если вы ничего не поняли, не огорчайтесь. Вычитание придумал Клини, когда ему вырывали зуб мудрости. А сейчас наркоз уже не тот».

Как видим, технически ничего сложного в лямбда-исчислении нет: всё сводится к элементарным подстановкам и редукциям. Но столь малого набора инструментов вполне хватает, чтобы при желании реализовать активные сущности, ведущие себя подобно парам, спискам, рекурсивным функциям и т.п. Они будут достаточно громоздкими, но, как уже говорилось, λ-исчисление предназначено не для написания программ, а для исследования и спецификации языков программирования и систем типов. С чем, собственно, и прекрасно справляется.

Типизированное лямбда-исчисление — это версия лямбда-исчисления, в которой лямбда-термам приписываются специальные синтаксические метки, называемые типами. Допустимы различные наборы правил конструирования и приписывания таких меток, они порождают различные системы типизации.

Типовые λ-исчисления являются фундаментальными примитивными языками программирования, которые обеспечивают основу типовым языкам функционального программирования — аппликативным языкам, — среди которых ML и Haskell, а также типовым императивным языкам программирования.

λ-исчисление с типами является языком декартово-замкнутой категории, что устанавливает прямую связь с такой моделью вычислений, как категориальная абстрактная машина. С одной точки зрения типовые λ-исчисления могут рассматриваться как специализации бестиповых λ-исчислений, а с другой — наоборот, типовые языки могут считаться более фундаментальными, из которых бестиповые получаются как частные случаи. Анализ этого явления дает теория вычислений Д. Скотта.

λ-исчисление с типами служит основой для разработки новых систем типизации для языков программирования, поскольку именно средствами типов и зависимостей между ними выражаются желаемые свойства программ.

В программировании самостоятельные вычислительные блоки (функции, процедуры, методы) языков программирования с сильной типизацией соответствуют типовым λ-выражениям.

15. Темпоральная логика

Темпоральная логика (англ. ***temporal logic***) — это логика, в высказываниях которой учитывается временной аспект. Используется для описания последовательностей явлений и их взаимосвязи по временной шкале.

В древности теории темпоральных логик изучали философы мегарской школы, в частности Диодор Крон, и стоики. Современная темпоральная логика была разработана в 1950-х Артуром Приором на

основе модальной логики и получила дальнейшее развитие в информатике благодаря трудам лауреата Тьюринговской премии Амира Пнуэли.

Есть два подхода темпоральной логики, основанные на принципах здравого смысла и диалектики: «после этого» означает «по причине этого», либо «после этого» означает «позже» в хронологическом смысле.

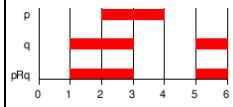
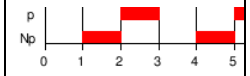
Пример

Рассмотрим утверждение: «Я голоден». Хотя смысл выражения не меняется со временем, его истинность может измениться. Утверждение в конкретный момент времени может быть истинным, либо ложным, но не одновременно. В противоположность нетемпоральным логикам, где значения утверждений не меняются со временем, в темпоральной логике значение зависит от того, когда оно проверяется. Темпоральная логика позволяет выразить утверждения типа «Я *всегда* голоден», «Я *иногда* голоден» или «Я голоден, *пока* я не поем».

15.1. Темпоральные операторы

В темпоральных логиках бывает два вида операторов: логические и модальные. В качестве логических операторов обычно используются ($\neg \wedge \vee \rightarrow$). Модальные операторы, используемые в логике линейного времени и логике деревьев вычислений, определяются следующим образом.

Текстовое обозначение	Символьное обозначение	Определение	Описание	Диаграмма
Бинарные операторы				
$\phi \bigcup \psi$	$\phi U \psi$	$(BUC)(\phi)=$ $(\exists i;C(\phi_i) \wedge$	**U**ntil (strong): ψ должно выполниться в	

		$(\forall j<i:B(\phi_j)))$	некотором состоянии в будущем (возможно, в текущем), свойство ϕ обязано выполняться во всех состояниях до обозначенного (не включительно).	
$\phi \mathrm{R} \psi$ $\phi \mathrm{V} \psi$	$\phi R \psi$ $\phi V \psi$	$(BRC)(\phi)=$ $(\forall i;C(\phi_i) \vee$ $(\exists j<i:B(\phi_j)))$	**R**elease: ϕ освобождает ψ, если ψ истинно, пока не наступит момент, когда ϕ первый раз станет истинно (или всегда, если такого момента не наступит). Иначе, ϕ должно хотя бы раз стать истинным, пока ψ не стало истинным первый раз.	
Унарные операторы				
$\mathbf{N}\phi$ $\mathbf{X}\phi$	$\circ\phi$	$N\mathrm{B}(\phi_i)=$ $B(\phi_{i+1})$	**NeXt**: ϕ должно быть истинным в состоянии, непосредственно следующим за данным.	
$\mathbf{F}\ \phi$	$\Diamond\ \phi$	$F\mathrm{B}(\phi)=$ $(trueUB)(\phi)$	**F**uture: ϕ должно стать	

			истинным хотя бы в одном состоянии в будущем.	
Gϕ	$\Box\phi$	GB(ϕ)= $\neg(B)\ \neg(\phi)$	**G**lobally: ϕ должно быть истинно во всех будущих состояниях.	
Aϕ	$A\phi$	$(AB)(\psi)$= $(\forall\ \phi: \phi_0 = \psi \rightarrow B(\phi)$	**A**ll: ϕ должно выполняться на всех ветвях, начинающихся с данной.	
Eϕ	$E\phi$	$(EB)(\psi)$= $(\exists\ \phi: \phi_0 = \psi \wedge B(\phi)$	**E**xists: существует хотя бы одна ветвь, на которой ϕ выполняется.	

Другие модальные операторы

- Оператор **W**, означающий ***W****eak until*: *fWg эквивалентно fUg* $\vee$ *Gf*

Тождества двойственности

Подобно правилам де Моргана существуют свойства двойственности для темпоральных операторов:

- $\phi\ U\psi = \neg(\neg\ \phi V \neg\psi)$
- $\neg\ \Diamond\ \phi = \Box \neg\ \phi$
- $\neg\ A\phi = E \neg\ \phi$

Приложения

Темпоральные логики часто применяются для выражения требований формальной верификации. Например, свойства типа «Если поступил запрос, то на него обязательно придёт ответ» или «Функция вызывается не более одного раза за вычисление» удобно формулировать с помощью темпоральных логик. Для проверки таких свойств используются различные автоматы, например, автоматы Бюхи для проверки свойств, выраженных логикой линейного времени LTL.

Темпоральные логики

Известны следующие темпоральные логики:

- Интервальная темпоральная логика
- μ-исчисление
  - CTL*
    - Логика линейного времени LTL
    - Логика деревьев вычислений CTL

16. Модальная логика

Модальная логика (от лат. *modus* — способ, мера) — логика, в которой кроме стандартных логических связок, переменных и/или предикатов есть **модальности** (модальные операторы). Модальности бывают разные; наиболее распространены временны́е («когда-то в будущем», «всегда в прошлом», «всегда» и т. д.) и пространственные («здесь», «где-то», «близко» и т. д.). Например, модальная логика способна оперировать утверждениями типа «Москва всегда была столицей России» или «Санкт-Петербург, когда-то в прошлом, был столицей России», которые невозможно или крайне сложно выразить в немодальном языке. Кроме временных и пространственных модальностей есть и другие, например «известно, что» (логика знания) или «можно доказать, что» (логика доказуемости).

Обычно для обозначения модального оператора используется $\Box$ и двойственный к нему $\Diamond$:

$$\Diamond A = \neg \Box \neg A.$$

Это отражает то, что сказать «Москва когда-то была столицей России» то же самое, что сказать «не верно, что Москва никогда не была столицей России».

16.1. Модальности

- Алетические (от др.-греч. ἀλήθεια — истина) модальные понятия:
  - Логические:
    - L — необходимо,
    - M — возможно,
    - C — случайно.
  - Фактические:
    - □ — необходимо,
    - ◊ — возможно,
    - Δ— случайно.

- Деонтические (др.-греч. δέοντος — должное, необходимое) модальные понятия:
  - обязательно,
  - разрешено,
  - запрещено.

Логику деонтических модальностей разработал финский философ Георг фон Вригт.

- Аксиологические (др.-греч. ἀξίᾱ — ценность) модальные понятия:
  - хорошо,
  - нейтрально,
  - плохо.

Аксиологическую логику разработал философ А. А. Ивин.

- Эпистемические (др.-греч. ἐπιστήμη — знание) модальные понятия:
  - знание,
  - полагание,
  - незнание.

Эпистемическая логика разработана Яакко Хинтикка.

- Временные:
  - прошлое,
  - настоящее,
  - будущее.

Логика знаний, *«эпистемическая логика»* (от греч. επιστήμη — *знание*)— подвид модальной логики, имеющий дело с высказываниями о знании (его состоянии). С этой темой соприкасаются философия, теоретическая информатика, искусственный интеллект, экономика и лингвистика. Её отдельные аспекты обсуждали Аристотель, такие средневековые философы, как Оккам и Дунс Скотт. Первые систематические исследования темы, развитие символики провёл Кларенс И.Льюис в 1912 г. Современную форму ей придал Сол Крипке (1963 г.), Георг Хенрик фон Вригт, Яакко Хинтикка (1962 г.) .

Логику знаний к экономике применил Роберт Ауман (Нобелевская премия по экономике 2005 года).

- Пространственные:
  - там,
  - здесь,
  - нигде.

16.2. Семантика

В математической логике и информатике наиболее распространённой является семантика Крипке, также существуют алгебраическая семантика, топологическая семантика и ряд других.

Семантика Крипке является распространенной семантикой для неклассических логик, таких как интуиционистская логика и модальная логика. Она была создана Солом Крипке в конце 1950-х — начале 1960-х годов. Это было большим достижением для развития теории моделей для неклассических логик.

Семантика для модальной логики

Рассмотрим одномодальные пропозициональные логики.

Шкалой Крипке *F* с одним отношением называется пара (W, R), где W— это произвольное множество (часто говорят множество возможных миров), а $R \subset W \times W$— отношение на W (множество стрелок или упорядоченных пар).

Моделью Крипке M называется пара (F, V), где V— это оценка на шкале, которая каждой переменной ставит в соответствие множество миров, в которых эта переменная считается истинной. Формально оценку представляют, как функцию из множества переменных PL в множество всех подмножеств W. Истинность в точке в модели Крипке обозначается с помощью знака $\models$ и определяется индукцией по длине формулы:

$M,x \models p$, если $x \in V(p)$

$M,x \not\models \bot$

$M,x \models AB$, если $M,x \not\models A$ или $M,x \models B$

$M,x \models \Box A$, если $\forall y:(xRy \Rightarrow M, y \models A)$

Другие логические связки, такие как $\wedge, \vee$ и $\neg$ можно выразить через $\rightarrow$ и $\bot$. Дуальный модальный оператор $\Diamond$ выражается так $\Diamond A \underset{def}{=} \neg \Box \neg A$..

Аналогично можно определить семантику для многомодальных логик, для этого в шкале Крипке должно быть столько отношений, сколько есть модальностей в логике.

16.3. Синтаксис

Модальная формула определяется рекурсивно как слово в алфавите состоящем из счетного множества пропозициональных переменных PL , классических связок $\rightarrow$, $\bot$, скобок (,), и модального оператора $\Box$. А именно, формулой является

1. *p* для любого $p \in PL$.
2. $\bot$.
3. $(A \rightarrow B)$, если A и B— формулы.
4. $(\Box A)$, если *A*— формула.

Нормальной модальной логикой называется множество модальных формул, содержащее все классические тавтологии, аксиому нормальности

$$\Box(p \rightarrow q) \rightarrow (p \Box \rightarrow \Box q)$$

и замкнутое относительно правил Modus ponens $\frac{A, A \rightarrow B}{B}$,

подстановки $\frac{A(p)}{A(B)}$ и введение модальности $\frac{A}{\Box A}$.

Минимальная нормальная модальная логика обозначается K .

Логика в информатике

Логика в информатике — это направления исследований и отраслей знания, где логика применяется в информатике и искусственном интеллекте. Логика очень эффективна в этих областях.

Область применения

Включаются следующие основные применения:

- исследования в логике, вызванные развитием компьютерных наук. Например, аппликативные вычислительные системы, теория вычислений и модели вычислений;
- формальные методы и логика рассуждения о понятиях. Например, семантическая сеть, семантическая паутина;
- булева логика и алгебра для разработки аппаратного обеспечения компьютеров;

- решение задач и структурное программирование для разработки прикладных программ и создания сложных систем программного обеспечения
- доказательное программирование — технология разработки алгоритмов и программ с доказательствами правильности алгоритмов;
- фундаментальные понятия и представления для компьютерных наук, которые являются естественной областью для формальной логики. Например, семантика языков программирования;
- логика знания и предположения. Например, искусственный интеллект;
- язык Пролог и логическое программирование для создания баз знаний и экспертных систем и исследований в сфере искусственного интеллекта;
- логика для описания пространственного положения и перемещения;
- логика в информационных технологиях. Например, реляционная модель данных, реляционные СУБД, реляционная алгебра, реляционное исчисление;
- логика вычислений с объектами. Например, комбинаторная логика, суперкомбинаторы;
- логика для компилирования программного кода и его оптимизации. Например, категориальная абстрактная машина;
- логика для эквивалентного преобразования объектов. Например, λ-исчисление;
- переизложение логики и математики в терминах, понятных специалистам в компьютерных науках.

Этот список продолжает пополняться.

16.4 Логическое программирование

Парадигмы программирования

• Императивная
(контрастирует с декларативной)

Процедурная
Структурная
Аспектно-ориентированная
Объектно-ориентированная
Агентно-ориентированная
Компонентно-ориентированная
Прототипно-ориентированная
Обобщённое программирование

• Декларативная
(контрастирует с императивной)

Чистота языка
Чистота функции
Функциональная
В терминах Рефал-машины
Аппликативная
Комбинаторная
Бесточечная
(чистая конкатенативная)
Логическая

Ограничениями

- Конкатенативная
- Векторная[en]
- Метапрограммирование

Языково-ориентированная
Предметно-ориентированная
Пользователями[en]
Автоматизация процесса программирования
Рефлексивность
Гомоиконность

- Связанные темы

Программирование в крупном и мелком масштабе[en]
Модульность
Полиморфизм
Продолжения и CPS
Параллелизм и конкурентность

- Методы и алгоритмы

Автоматное
Динамическое
Потоков данных
Событийно-ориентированное
Реактивное

Сервис-ориентированное

Логи́ческое программи́рование — парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода. Логическое программирование основано на теории и аппарате математической логики с использованием математических принципов резолюций.

Самым известным языком логического программирования является Prolog.

Первым языком логического программирования был язык Planner, в котором была заложена возможность автоматического вывода результата из данных и заданных правил перебора вариантов (совокупность которых называлась планом). Planner использовался для того, чтобы понизить требования к вычислительным ресурсам (с помощью бэктрекинга — поиска с возвратом) и обеспечить возможность вывода фактов, без активного использования стека. Затем был разработан язык Prolog, который не требовал плана перебора вариантов и был, в этом смысле, упрощением языка Planner.

От языка Planner также произошли логические языки программирования QA-4, Popler, Conniver и QLISP. Языки программирования Mercury, Visual Prolog, Oz и Fril произошли уже от языка Prolog. На базе языка Planner было разработано также несколько альтернативных языков логического программирования, не основанных на методе поиска с возвратами), например, Ether).

Часть вторая

Логика Лукасевича

Пионеры многозначных логик Э. Пост, Я. Лукасевич и Д. А. Бочвар создавали свои системы, имея разные цели. *N*-значные функционально полные системы Э. Поста P_n являются обобщением (с циклическим отрицанием) двузначной логики, не имеющими сколь-нибудь семантически содержательной интерпретации. Трехзначная логика Д. А. Бочвара $\mathbf{B}_3$ с промежуточным истинностным значением "бессмысленно" предназначена для формализации логических и семантических парадоксов (ее семантическое истолкование очевидно - потеря смысла высказываний в формально корректном языке). Трехзначная логика Я. Лукасевича, созданная им в 1920 г., имела философскую мотивацию и была связана с его идеей опровергнуть аристотелевскую доктрину логического фатализма, основанную на двузначной логике.

Однако *n*-значные обобщения логики Я. Лукасевича оказались интересным логико-математическим формализмом, который, не имея ясного семантического истолкования (имеются в виду истинностные значения, отличные от "истины" и "лжи"), породил многочисленные исследования логического и алгебраического характера.(Следует упомянуть в связи с этим польских логиков А. Тарского, А. Линденбаума, М. Вайсберга, Е. Слупецкого, Р. Вуйцицкого и других, а также американских логиков Б. Россера и А. Тюркетта, Ч. Чэна, А. Роуза и других исследователей и грузинского логика Р. Григолия.)

Многочисленные попытки построить семантическое истолкование *n*-значных логик Я. Лукасевича $Ł_n$ $(n \geq 3)$ бывали остроумными, но не исчерпывающе убедительными (например, интерпретация Д. Скотта посредством идеи "степени ошибки").
Первым глубоким результатом, устанавливающим связь между логиками Я. Лукасевича и арифметическими фактами, была теорема Р. Мак-Нотона о $Ł_\infty$ и ее конечных фрагментах $Ł_n$ (для каждого набора истинностных значений $\frac{S}{n-1}, \ldots, \frac{S_m}{n-1}$ всякий общий делитель чисел $S_1, \ldots, S_m, (n\text{-}1)$ является делителем *S,* где

$F(\frac{S_1}{n-1}, \ldots, \frac{S_m}{n-1}) = \frac{S}{n-1}$, а $F(x_1,\ldots, x_m)$ — функция, выразимая в $Ł_n$)
Затем были обнаружены факты связи $Ł_n$ и простых чисел, а именно, было показано, что логика Я. Лукасевича $Ł_n$ функционально предполна тогда и только тогда (т.т.т.), когда (*n*-1) -простое число, а также было обнаружено, что $Ł_n$ имеет *I-J*-совершенную дизъюнктивную нормальную форму т.т.т., когда (*n*-1) есть степень простого числа. (*I-J* -с.д.н.ф. есть дизъюнкция конъюнкций элементарных формул вида $J_\nu p$ и $I_{ij} p$.

$$J_\nu p = \begin{cases} 1, & \text{если } p = \nu \\ 0, & \text{иначе} \end{cases}, \qquad I_{ij} p = \begin{cases} \frac{j}{n-1}, & \text{если } p = \frac{i}{n-1} \\ 0, & \text{иначе} \end{cases}.$$

Напомним, что множество функций *X n*-значной логики называется предполным в P_n, если $X \neq P_n$ и для всякой *F* такой, что $F \overline{\in} X$, $[X \cup \{F\}] = P_n$, где $[X \cup \{F\}]$ - множество всех суперпозиций функций из $X \bigcup \{F\}$.
Автор приводимого ниже материала по логике Лукасевича А. С. Карпенко в многочисленных работах получил ряд глубоких и трудно доказуемых результатов, устанавливающих интересные связи между логиками $Ł_n$ и логиками функционально эквивалентными $Ł_n$, с одной стороны, и арифметическими фактами, с другой стороны. Он построил характеризации четных чисел и нечетных чисел, соответственно, посредством логических исчислений, функционально эквивалентных логикам Я. Лукасевича $Ł_n$.
А. С. Карпенко получил также аналоги результатов, характеризующие простые числа посредством специально построенных исчислений функционально эквивалентных $Ł_n$.
Весьма эффектными и труднодоказуемыми результатами А. С. Карпенко являются теоремы о характеризации простых чисел через аналоги штриха Шеффера для соответствующих *n*-значных логик: для любого $n \geq 3$ *n* есть простое число т.т.т., когда $S_{n+1} = K_{n+1}$, где K_{n+1} - логика функционально эквивалентная $Ł_{n+1}$, а S_{n+1} -множество всех суперпозиций аналога штриха Шеффера для $Ł_{n+1}$.
Таким образом, важным итогом многолетних и плодотворных исследований А. С. Карпенко являются характеризации простых чисел, степеней простых чисел, четных чисел и нечетных чисел как посредством специально построенных логических исчислений, так и

посредством аналогов штриха Шеффера для соответствующих логик Я Лукасевича.
Методологический смысл результатов А. С. Карпенко состоит в том, что обнаружена связь между фактами арифметики и конечнозначными логиками Я. Лукасевича. По-видимому, "арифметическая природа" логик Я. Лукасевича не только обнаружена, но и систематически изучена. Это обстоятельство является некоторым аргументом против общепринятого понимания логик Я. Лукасевича (бесконечнозначной и конечнозначных) как логических оснований формализации нечеткости в смысле Л. Заде (хотя, разумеется, для широкого класса прикладных задач несомненно требуются соответствующие недвузначные логики как аппарат формализации правдоподобных рассуждений).
В этой книге А. С. Карпенком представлены также результаты относительно порождений классов простых чисел, полученные посредством компьютерных программ, ставших эффективным подспорьем современных исследований комбинаторных проблем.
В 1979 г. Д. А. Бочвар высказал идею о том, что многозначные логики следует рассматривать как фрагменты формализованных семантик. Если принять эту идею, то для трех трехзначных логик B_3 (логика Бочвара), E_3 (логика Эббингхауза) и $Ł_3$ (логика Лукасевича) имеют место следующие включения (относительно множеств тавтологий и множества функций в них выразимых): $B_3 \subset E_3 \subset Ł_3$. Для этих логик допустима следующая интерпретация : B_3 - логика "математической бессмыслицы" (парадоксальность высказываний, деление на нуль и т. п.), E_3 - логика "лингвистической бессмыслицы", $Ł_3$ - логика неопределенности (понимаемой в том смысле, что промежуточное истинностное значение может быть либо истинным, либо ложным, но этот факт не установлен).
Аналогичные включения имеют место и для n-значных обобщений B_3, E_3 и L_3: $B_n \subset E_n \subset Ł_n \subseteq T_n$ для $n \geq 3$, где $T_n = Ł_n$, если и только если (n-1) - простое число, а T_n содержит все функции $I_{ij}(x)$

где $1 \leq i, j \leq n\text{-}2$, а $I_{ij}(x) = \begin{cases} j, \text{ если } x = \frac{i}{n-1} \\ 0, \text{ если } x = \frac{i}{n-1} \end{cases}$.

Логики T_n являются предполными (в P_n) классами функций, сохраняющими "истину" (1) и "ложь" (0). В этом смысле они являются максимально не-постовскими, содержащими изоморф двузначной логики.
Идея максимальности $Ł_3$ была высказана В. И. Шестаковым и формализована им, где было показано, что класс функций, соответствующий $Ł_3$, является предполным в P_3. Этот простой

результат привел к неочевидному обобщению: $Ł_n$ предполна т.т.т., когда (n-1) - простое число. Следовательно, простые числа характеризуются предполными классами логик Я. Лукасевича $Ł_n$. Впоследствии (в 1983 г. этот факт был обнаружен Г. Е. Хендри, а в 1986 г. - А. Урквартом).
Однако систематическое исследование характеризации различных множеств натуральных чисел (простых чисел, степеней простых чисел, четных и нечетных чисел) посредством логик $Ł_n$ или функционально им эквивалентных были осуществлены лишь А. С. Карпенко, результаты которого и представлены ниже.

1. Классическая логика высказываний

1.1. Логические связки. Истинностные таблицы

Логика высказываний (пропозициональная логика) является разделом символической логики, изучающим сложные высказывания, образованные из простых, и их взаимоотношения. В отличие от логики предикатов простые высказывания при этом выступают как целостные образования, внутренняя структура которых не рассматривается, а учитывается лишь то, с помощью каких союзов и в каком порядке простые высказывания сочленяются в сложные. **Под <u>высказыванием</u> понимается то, что выражается повествовательным предложением**.
В естественном языке существует много способов образования сложных высказываний из простых. Мы выберем пять общеизвестных грамматических связок (союзов) «не», «и», «или», «если , то» и «если и только если». <u>Процесс символизации естественного языка средствами логики высказываний состоит в следующем</u>:
Элементарные высказывания замещаются *пропозициональными переменными p, q, r,...* с индексами или без них, указанные выше грамматические связки называются *логическими связками* (пропозициональными связками), которые получили следующие обозначения и названия $\neg$ (отрицание), $\wedge$ или & (конъюнкция), $\vee$ (дизъюнкция), $\supset$ (импликация) и $\equiv$ (эквиваленция), и, наконец, используются скобки (,) для того, чтобы можно было по-разному группировать высказывания и тем самым определять порядок выполнения операций.
<u>Отрицание</u> является <u>одноместной связкой</u>, а остальные <u>четыре - двухместными</u>. Выражением языка логики высказываний будем называть любую последовательность указанных выше символов. Некоторые из этих выражений являются правильно построенными.

Такие выражения называются *формулами,* определение которых задается следующими правилами, где буквы *A, B...* используются как метапеременные: (1) всякая пропозициональная переменная есть формула; (2) Если *A* и *B* - формулы, то
$(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \supset B)$, $(A \equiv B)$ тоже формулы;
(3). Никакие другие выражения не являются формулами. Примерами формул являются p, $\neg q$, $\neg(p \vee q)$. Внешние скобки при записи формул будем опускать. Таким образом, **правила задают эффективный способ распознавания, является ли выражение логики высказываний формулой.** Множество всех формул обозначим посредством *For.*
Теперь сделаем два основных допущения, на которых основывается семантика классической логики высказываний:
(I) Каждое простое высказывание является или только истинным, или только ложным (принцип двузначности). «Истина» и «ложь» называются *истинностными значениями* высказывания и обозначаются соответственно И и Л или 1 и 0.
(II) Истинностное значение сложного высказывания определяется только истинностными значениями составляющих его простых высказываний (принцип экстенсиональности). Это означает, что пропозициональные связки являются знаками *истинностных функций.*
Возникает вопрос, какие истинностные функции соответствуют нашим связкам?
Удобным способом задания истинностных функций является табличный, где слева указываются все возможные приписывания значений аргументам (пропозициональным переменным), а справа - значения самой функции:

p	¬p
1	0
0	1

p	q	p ⊃ q	p ∨ q	p ∧ q	p ≡ q
1	1	1	1	1	1
1	0	0	1	0	0
0	1	1	1	0	0
0	0	1	0	0	1

Отсюда, например, следует, что высказывание $p \supset q$ ложно тогда и только тогда (т.т.т.), когда *p* истинно и *q* ложно. Приведенные выше таблицы называются *истинностными таблицами,* а определенные посредством их пропозициональные связки называются *классическими связками.*
Легко определить, сколько имеется различных классических связок. Число различных строк в таблице для истинностной функции с *m* аргументами равно 2^m и на каждой из них значение функции можно задать двумя способами: 1 или 0. Поэтому число таких функций составляет 2 в степени 2^m. Отсюда, например, число одноместных связок равно 4, а число двухместных связок равно 16.

1.2. Законы логики

Каждая формула определяет некоторую истинностную функцию, которая графически может быть представлена истинностной таблицей. Другими словами, каждая формула может быть представлена как функция от пропозициональных переменных, пробегающих по множеству {0, 1}. Посредством истинностных таблиц эта функция единственным образом расширяется на всё множество формул *For.*
Функцию $v : For \to \{0, 1\}$ будем называть *логической оценкой* множества формул *For* для любых $A, B \in For$:

$v(\neg A) = 1$ т.т.т., когда $v(A) = 0$

$v(A \supset B) = 0$ т.т.т., когда $v(A) = 1$ и $v(B) = 0$

$v(A \vee B) = 0$ т.т.т., когда $v(A) = v(B) = 0$

$v(A \wedge B) = 1$ т.т.т., когда $v(A) = v(B) = 1$

$v(A \equiv B) = 1$ т.т.т., когда $v(A) = v(B)$.

Среди всего множества формул выделяются формулы, которые на каждой строке истинностной таблицы принимают только значение, равное 1, т.е. $v(A) = 1$ при любом приписывании значений пропозициональным переменным, входящим в *A.* Такие формулы называются *тавтологиями* (тождественно истинными высказываниями). В формальной логике тавтологии играют важную роль. Они служат для записи её законов, так как тавтологии являются всегда истинными высказываниями только в силу своей символической формы, независимо от содержания входящих в них исходных высказываний. Легко установить, что формулы

(1) $p \supset p$,

(2) $p \vee \neg p$,

(3) $\neg(p \wedge \neg p)$

являются тавтологиями. Законы, выражаемые этими формулами, называются соответственно *законом тождества, законом исключенного третьего* и *законом непротиворечия* и были сформулированы уже Аристотелем. Использование этих законов в качестве способов рассуждения привело к тому, что они были названы ***основными законами мышления***. Наиболее распространенной формулировкой закона исключенного третьего является следующая: *одно из утверждений p или не-p должно быть истинным.* Эта формулировка получила в схоластической логике название *tertium non datur.* **Закон непротиворечия формулируется следующим образом**: *два взаимно противоречащих высказывания не могут быть одновременно истинными, т.е. одно из них должно быть ложным.* **Последний закон формулируется у Аристотеля прежде всего как универсальный принцип бытия, наиболее достоверный из всех начал**. Однако уже на заре XX в. еще до того, как окончательно оформилась классическая логика, оба эти закона подверглись серьезной критике, что положило начало развитию неклассических логик. В связи с трехзначной логикой Лукасевича мы к этим законам ещё вернемся, а сейчас дополним список законов классической логики:

(4) $\neg\neg p \equiv p$ (*закон двойного отрицания*)

(5) $(p \supset q) \supset (\neg q \supset \neg p)$ (*закон контрапозиции*)

(6) $(\neg p \supset \neg q) \supset (q \supset p)$ (*обратный закон контрапозиции*).

Особое место среди законов занимают чисто импликативные тавтологии:

(7) $p \supset (q \supset p)$ (*закон утверждения консеквента*)

(8) $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$ (*закон самодистрибутивности*)

(9) $(p \supset q) \supset ((q \supset r) \supset (p \supset r))$ (*закон транзитивности*)

(10) $(p \supset (p \supset q)) \supset (p \supset q)$ (*закон сокращения*).

Обратим внимание на исключительно важное свойство **истинностных таблиц: они дают нам эффективную процедуру для решения вопроса о том, является ли данная пропозициональная формула тавтологией. Указанная процедура называется *разрешающей***

процедурой* и отсюда следует, что данная логика высказываний является *разрешимой логикой.

Приведем некоторые общие факты о тавтологиях, настолько общие, что они называются *правилами логики высказываний.*

1. *Правило отделения* (modus ponens). Если *A* и $A \supset B$ тавтологии, то *B* тавтология (сокращенно *MP).*

2. *Правило подстановки.* Если *A(p)* есть тавтология, то *A(B)* тоже тавтология, где *B* замещает каждое вхождение *p,* т.е. подстановка в тавтологию приводит к тавтологии (сокращенно *Subst).* Уже отсюда следует бесконечное множество тавтологий.

1.3. Функциональная полнота

Будем называть формулы *A* и *B логически эквивалентными,* если формула $A \equiv B$ есть тавтология. Очевидно, что если формулы *A* и *B* эквивалентны, то они равны как истинностные таблицы, т.е. принимают одинаковые истинностные значения.

Назовем систему пропозициональных связок $\mathcal{M}$ *полной,* если всякая истинностная функция представима некоторой формулой, в которую входят только связки из системы $\mathcal{M}$, т.е. посредством такой системы можно выразить все истинностные функции. Используя свойства логической эквивалентности, можно показать, что каждая логическая связка может быть определена в терминах $\neg$, $\wedge$, $\vee$ в классической логике, т.е. система пропозициональных связок $\{\neg, \wedge, \vee\}$ является функционально полной. Более точно, для каждой истинностной функции * можно найти такую формулу D, использующую только связки $\neg$, $\wedge$, $\vee$, что истинностные таблицы для * и D равны.

Теорема о функциональной полноте. *В классической логике каждая истинностно-функциональная связка может быть определена в терминах* $\neg$, $\wedge$, $\vee$*[Мендельсон* 1976].

Впервые подобная теорема была доказана Э.Постом *[Post* 1921].

Отметим некоторые эквивалентности, показывающие взаимовыразимость одних связок через другие.

$p \vee q \equiv \neg p \supset q$, $p \vee q \equiv (p \supset q) \supset q$, $p \vee q \equiv \neg(\neg p \wedge \neg q)$;

$p \wedge q \equiv \neg(p \supset \neg q)$, $p \wedge q \equiv \neg(\neg p \vee \neg q)$;

$p \supset q \equiv \neg p \vee q$, $p \supset q \equiv \neg(p \wedge \neg q)$,

$(p \equiv q) \equiv (p \supset q) \wedge (q \supset p)$

Тогда системы связок $\{\neg, \supset\}$, $\{\neg, \vee\}$ и $\{\neg, \wedge\}$ являются

функционально полными. Это значит, что мы можем строить логику высказываний, взяв в качестве исходной любую из указанных систем связок.

1.3.1. Штрих Шеффера

В классической логике существуют две истинностно-функциональные связки, каждая из которых образует функционально полную систему. Первая из них называется *штрих Шеффера* (1913 г): высказывание $p|q$ истинно т.т.т., когда не истинно p и не истинно q, т.е. $p|q \equiv \neg p \wedge \neg q$. Достаточность связки $|$ следует из тавтологий $\neg p \equiv p|p$, $p \wedge q \equiv (p|p) \,|\, (q|q)$. Другая связка называется *стрелка Пирса:* высказывание $p \downarrow q$ истинно т.т.т., когда неверно, что p и q оба истинны, т.е. $p \downarrow q \equiv \neg(p \wedge q)$. Достаточность связки $\downarrow$ следует из тавтологий $\neg p \equiv p \downarrow p$, $p \vee q \equiv (p|p) \,|\, (q|q)$.

Таким образом, для того чтобы показать, что какая-то связка является штрихом Шеффера, надо (i) определить её посредством исходных связок, а затем (ii) посредством её определить исходные. Некоторые аналогии штриха Шеффера нам понадобятся в последующем.

1.4. Аксиоматизация. Адекватность

Наряду с понятием тавтологии фундаментальным для логики является понятие *логического следования.* **Одна из главных задач логики заключается в том, чтобы устанавливать, *что из чего следует,* и тем самым определять, какие высказывания являются *теоремами* при заданных условиях.** Говорят *«B* логически следует из *A* или является логическим следствием из *A»* и пишут $A \models B$, если в совместной таблице истинности для *A и B* формула *B* имеет значение. И во всех тех строках, где *A* имеет значение И. Отсюда следует, что $A \models B$ ттт., когда $A \supset B$ есть тавтология. Если формула *A* является тавтологией, то иногда пишут $\models A$. Приведенное определение логического следования без труда может быть расширено на некоторую систему формул *Г,* и тогда пишут $\Gamma \models B$. Примером логического следования (вывода) из посылок является правило modus ponens. Отметим также, что в силу табличного определения импликации получаем, что тождественно истинная формула *A* логически следует из любой системы формул. А из того, что имеется разрешающая процедура для тавтологий, получаем, что проблема

выводимости произвольной формулы *B* из заданной системы посылок также разрешима.

Если определено понятие тавтологии и определено семантическое понятие логического следования (как это сделано выше), то говорят, что дано *семантическое представление* логики высказываний, а сама логика высказываний зачастую отождествляется с множеством тавтологий или с самим отношением логического следования. Однако при этом возникает следующая серьезная проблема: как обозреть все тавтологии, которых бесконечное множество? **Для решения этой проблемы переходят к синтаксическому представлению логики высказываний**.

В рамках синтаксического подхода формальный (символический) язык логики высказываний и понятие формулы остаются прежними, а из всего множества тавтологий выбирается некоторое их конечное подмножество, элементы которого называются *аксиомами.* Например:

1. $p \supset (q \supset p)$
2. $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$
3. $p \supset (p \vee q)$
4. $q \supset (p \vee q)$
5. $(p \supset r) \supset ((q \supset r) \supset ((p \vee q) \supset r))$
6. $(p \wedge q) \supset p$
7. $(p \wedge q) \supset q$
8. $(p \supset q) \supset ((p \supset r) \supset (p \supset (q \wedge r)))$
9. $(p \supset \neg q) \supset (q \supset \neg p)$
10. $p \supset (\neg p \supset q)$
11. $p \vee \neg p$.

Таким образом, мы задали аксиоматическое определение логических связок $\neg$, $\wedge$, $\vee$, $\supset$ в отличие от табличного при семантическом описании логики высказываний. Переход от формулы или системы формул к формуле осуществляется с помощью уже известных правил, которые записываются следующим образом:

R1. Из A и $A \supset B$ следует B (modus ponens)

R2. Из $\vdash A(p)$ следует $\vdash A(B)$ (подстановка).

Так заданную логику высказываний обозначим посредством $\mathbf{C}_2$ и назовем *классической логикой высказываний.*

Из раздела (1.3) следует, что логику высказываний можно развивать на основе системы связок $\{\neg, \supset\}$. Именно так *впервые* и была представлена аксиоматизация $\mathbf{C}_2$ в работе Г.Фреге *[Frege* 1879]. Следующая аксиоматизация $\mathbf{C}_2$ принадлежит Лукасевичу *[Lukasiewicz & Tarski* 1930], который значительно упростил аксиоматизацию, предложенную Фреге:

1. $p \supset (q \supset p)$
2. $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$
3. $(\neg p \supset \neg q) \supset (q \supset p)$.

Правила вывода: modus ponens и подстановка.
Детально эта аксиоматизация $\mathbf{C}_2$ исследуется А.Чёрчем *[Чёрч* 1960].
Логическое исчисление, заданное посредством некоторого множества аксиом и некоторого множества правил вывода, называется *гильбертовским исчислением*. *Доказуемыми формулами* (или теоремами) рассматриваемого исчисления называются любые формулы, которые могут быть получены из аксиом исчисления в результате применения (возможно, многократного) указанных правил.
Запись $\vdash A$ служит сокращением утверждения «*A* есть теорема».
Если формула *A* выводима из некоторого множества Г исходных формул (посылок), то запись принимает вид $\Gamma \vdash A$.
В качестве «вспомогательного» правила весьма полезной является ***теорема дедукции***, когда какое-нибудь утверждение *B* доказывают в предположении верности другого утверждения *A,* после чего заключают, что верно утверждение «если *A,* то B»: Теорема дедукции. *Если* Г - *множество формул, A и B - формулы и* $\Gamma, A \vdash B$, *то* $\Gamma \vdash A \supset B$. *В частности, если* $A \vdash B$, *то* $\vdash A \supset B$ *[Herbrand* 1930].
Исходя из синтаксического представления логики высказываний, последняя зачастую отождествляется с множеством теорем или, что более принято, с **отношением выводимости**. Итак, **при семантическом подходе формулы интерпретируются как функции на множестве из двух элементов {0, 1}, а при синтаксическом -как определенный набор символов, и различаются только теоремы и не теоремы**. Однако несмотря на такое различие, оба подхода к построению логики высказываний, по существу, эквивалентны и, как говорят, являются *адекватными.* Это значит, что **понятия логического следования и понятия вывода эквивалентны.** Рассмотрим в связи с этим весьма примечательную теорему
Теорема адекватности. *Для всякой формулы A,* $\vdash A$ т.т.т., когда $\models A$.
Доказательство в одну сторону, а именно: для всех *A,* если $\vdash A$, то $\models A$ носит название ***теоремы о корректности***. Это

минимальное условие, которое мы требуем от логического исчисления и которое состоит в том, что представленная нами семантика корректна для выбранной аксиоматизации. Для доказательства теоремы нужно проверить, что все наши аксиомы (1) - (11) являются тавтологиями, что легко устанавливается непосредственной проверкой с помощью истинностных таблиц. А наши правила вывода выбраны таким образом, что они сохраняют тавтологичность. Поэтому **все формулы последовательности, образующей доказательство какой-либо теоремы исчисления $\mathbf{C}_2$, в том числе и сама доказанная теорема, являются тавтологиями.** Из этой теоремы следует важнейшее свойство нашего исчисления высказываний $\mathbf{C}_2$: в $\mathbf{C}_2$ **формулы *A* и $\neg A$ не могут быть одновременно доказуемыми, т.е. исчисление высказываний $\mathbf{C}_2$ *непротиворечиво*.** Если бы это было не так, то используя аксиому (10) и применяя дважды modus ponens, получаем, что в $\mathbf{C}_2$ доказуема любая формула *B.* В силу этого **противоречивая логика высказываний никакой ценности не представляет. В ней истина и ложь неразличимы и поэтому любая теорема одновременно истинна и ложна.**
Имеет место и обратное утверждение о том, **что каждая тавтология доказуема, т.е. *для всякой формулы A, если* $\models A$, *то* $\vdash A$.**
Доказательство этой теоремы не столь тривиально и носит название ***теоремы о полноте*** **(дедуктивной) исчисления высказываний относительно предложенной семантики.** По существу здесь утверждается, что логических средств, т.е. аксиом и правил вывода исчисления высказываний $\mathbf{C}$ 2 вполне достаточно для доказательства всех тавтологий. Таким образом, главная цель достигнута: используя минимальные средства, можно обозреть все множество тавтологий.
Как уже говорилось, первая аксиоматизация классической логики $\mathbf{C}_2$ была предпринята Г.Фреге. Однако в терминах современного символического языка аксиоматизация $\mathbf{C}_2$ появилась в «Principia Mathematica» А.Уайтхеда и Б.Рассела *[Whitehead & Russell* 1910-1913]. В обеих работах вопрос о полноте просто не возникал. Их целью было показать, что вся логика, а в действительности вся математика, может быть развита внутри их системы, основанной на классической логике. Первая публикация доказательства полноты принадлежит Посту *[Post* 1921], который исходил из системы Уайтхеда и Рассела. Для доказательства теоремы адекватности Пост использовал двузначные истинностные таблицы (приведенные выше).

1.5. Алгебраизация

Обратим внимание на то, что некоторые эквивалентности логики высказываний выражают основные свойства пропозициональных связок. Например, эквивалентности $(A \vee B) \equiv (B \vee A)$ и $(A \wedge B) \equiv (B \wedge A)$ выражают *коммутативный закон* связок конъюнкции и дизъюнкции. Это позволяет представить логику высказываний в виде своеобразной алгебраической структуры.

Непустое множество L с двумя бинарными операциями $\vee$ и $\wedge$ на L называется *решеткой,* если L удовлетворяет следующим тождествам *[Гретцер* 1982]:

I. (a) $x \vee x = x$

(b) $x \wedge x = x$ **(идемпотентность)**

II. (a) $x \vee y = y \vee x$

(b) $x \wedge y = y \wedge x$ **(коммутативность)**

III. (a) $x \vee (y \vee z) = (x \vee y) \vee z$

(b) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ **(ассоциативность)**

IV. (a) $x \vee (x \wedge y) = x$

(b) $x \wedge (x \vee y) = x$ **(поглощение)**

Решетка L называется *дистрибутивной,* если выполняются законы дистрибутивности:

V. (a) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

(b) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

Дистрибутивные решетки лежат в основе большинства хорошо известных многозначных логик. Специально дистрибутивным решеткам посвящена монография *[Balbes & Dwinger* 1974].

Дистрибутивная решетка L называется *решеткой де Моргана [Moisil* 1935], если для одноместного оператора ~ (инволюция) выполняются тождества

VI. $\sim\sim x = x$

VII. $\sim(x \vee y) = \sim x \wedge \sim y$

VIII. $\sim(x \wedge y) = \sim x \vee \sim y$.

Решетка L с двумя нульарными операциями 0 и 1 называется *ограниченной:*

IX. (a) $x \vee 0 = x$

(b) $x \wedge 1 = x$.

X. (a) $x \vee 1 = 1$

(b) $x \wedge 0 = 0$.

Ограниченные решетки обычно называются алгебрами. Соответственно ограниченная дистрибутивная решетка $\langle L, \vee, \wedge, \sim, 0, 1\rangle$ с отрицанием де Моргана (тождества (VI) - (VIII)) называется *алгеброй де Моргана.* Алгебра де Моргана, в которой операция ~ выполняет условие

(K). $x \wedge \sim x \leq y \vee \sim y$ **для всех** $x, y \in L$,

называется *алгеброй Клини [Kalman* 1958].

В *[Mukaidano* 1981] найдено характеристическое тождество, превращающее алгебру де Моргана в алгебру Клини:

(K'). $(x \wedge \sim x) \vee (y \vee \sim y) = y \vee \sim y$.

В ограниченной решетке L элемент y называется *дополнением* x, если $x \wedge y = 0$ **и** $x \vee y = 1$. В этом случае элемент y обозначают $\sim x$.

Булевой алгеброй называется дистрибутивная решетка с дополнениями. Имеется большое число различных (эквивалентных) систем тождеств, определяющих класс булевых алгебр.

Например, алгебра $\mathcal{B} = \langle L, \vee, \wedge, \sim, 0, 1\rangle$ называется *булевой алгеброй,* если $\langle L, \vee, \wedge, 0, 1\rangle$ есть ограниченная дистрибутивная решетка и выполняются следующие два тождества:

(B1). $x \vee \sim x = 1$

(B2). $x \wedge \sim x = 0$

Понятно, что булева алгебра является также алгеброй де Моргана и Клини, поскольку все условия для последних выполняются в булевой алгебре.

Примеры. 1) *Алгебра множеств.* Пусть $\mathcal{P}(U)$ есть множество всех подмножеств некоторого множества U. Для $X \in \mathcal{P}(U)$ определим -X как дополнение $U \backslash X$ множества X, а для X и Y из $\mathcal{P}(U)$ пусть $X \cup Y$ обозначает обычное теоретико-множественное объединение множеств X и Y, а $X \cap Y$ обозначает теоретико-множественное пересечение множеств X и Y. Тогда $\langle \mathcal{P}(U), \cup, \cap, - \rangle$ оказывается булевой алгеброй. Роль 0 играет пустое множество $\varnothing$, а 1 есть U.

2) *Алгебра классических истинностных значений.* Двухэлементная структура $\mathcal{B}_2 = < \{0, 1\}, \vee, \wedge, \neg, 0, 1 >$ с операциями, определенными посредством истинностных таблиц, является простейшей булевой алгеброй.

Классическим и одним из наиболее важных результатов в теории булевых алгебр стала

Теорема представления Стоуна *[Stone* 1936]: *каждая булева алгебра изоморфна алгебре множеств.*

Другая теорема представления утверждает: *каждая булева алгебра изоморфна подалгебре прямого произведения двухэлементной булевой алгебры* $\mathcal{B}_2$; или в другой формулировке: *каждая булева алгебра изоморфна подпрямому произведению двухэлементной булевой алгебры* $\mathcal{B}_2$.

Таким образом, благодаря теоремам представления абстрактные элементы булевых алгебр приобретают конкретный смысл.

Приведем еще один примечательный пример булевой алгебры:

3) *Алгебра Линденбаума.* Рассмотрим бинарное отношение ≈ на множестве формул *For*: $\mathbf{A} \approx \mathbf{B}$ т.т.т., когда $\mathbf{A} \equiv \mathbf{B}$ есть тавтология. Легко убедиться, что ≈ есть отношение эквивалентности на множестве формул *For.* Для произвольных классов эквивалентности |A| и | B | из $For/\approx$ пусть $|\mathbf{A}| \cup |\mathbf{B}| = |\mathbf{A} \vee \mathbf{B}|, |\mathbf{A}| \cap |\mathbf{B}| = |\mathbf{A} \wedge \mathbf{B}|$ и $-|\mathbf{A}| = |\neg\mathbf{A}|$. Тогда алгебра $L^* = < For/\approx, \cup, \cap, -, \mathbf{0}, \mathbf{1}>$ называется *алгеброй Линденбаума* (классической логики) и есть не что иное как булева алгебра. Нулевым элементом 0 здесь является класс всех противоречий $|\mathbf{A} \wedge \neg\mathbf{A}|$, а единицей 1 - класс эквивалентности, состоящий из всех тавтологий $|\mathbf{A} \vee \neg\mathbf{A}|$.

Легко видеть, что между эквивалентностями классической логики высказываний $\mathbf{C}_2$ и тождествами булевой алгебры существует соответствие. Например, между формулой $(A \vee B) \equiv (A \vee B)$ и первым тождеством в (II). Более того, $\vdash A$ *в* C_2 *т.т.т., когда* $A^* = 1$ *в* L^*, где A^* есть аналог A на языке алгебры L^*. Таким образом, возникают средства для *алгебраического* доказательства дедуктивной полноты логических исчислений.

Алгебры Буля, явившись результатом исследований Г. Буля в области законов правильных рассуждений *[Boole* 1847], нашли самое широкое применение в логико-математических исследованиях, в области инженерии контактно-релейных схем, компьютерных наук, аксиоматической теории множеств, теории моделей, и в других областях науки и математики. Хорошее введение в теорию булевых алгебр имеется в *[Burns & Sankappanctvar* 1981]. См. также *[Владимиров* 1969]. Популярное изложение имеется в *[Яглом* 1980], где

рассматриваются также конечные булевы алгебры. Имеется трехтомный справочник по булевым алгебрам *[Monk* (ed) 1989].

2. Трехзначная логика Лукасевича Ł$_3$

Логическое опровержение фаталистического аргумента Аристотеля привело Я.Лукасевича к открытию одной из самых необычных логических систем в мире. Хотя метатеория для трехзначной логики Лукасевича **Ł$_3$** *строится по аналогии с классической логикой* **C$_2$** *и в ней сохраняются свойства непротиворечивости, полноты и разрешимости, все равно отличие* **Ł$_3$** *от* **C$_2$** *оказывается фундаментальным. Самое главное заключается в том, что* **Ł$_3$** *не есть ограничение* **C$_2$**, *а наоборот, есть расширение последней (хотя в первой и не проходят такие основные законы классической логики, как закон исключенного третьего и закон непротиворечия). Значит, на самом деле появляются новые операции с совершенно странными свойствами (таковой является импликация Лукасевича) и тогда встает серьезный вопрос об их логическом статусе и о том, что такое логическая система вообще.*

2.1. Ян Лукасевич

Ян Лукасевич (Jan Lukasiewicz) родился 21 декабря 1878 г. во Львове и умер 13 февраля 1956 г. в Дублине. Выдающийся польский логик и философ, один из главных представителей *Львовско-Варшавской Школы,* зачинатель исследований по математической логике в Польше. Философское образование получил во Львове под руководством К.Твардовского - основателя Л.-В. Школы, затем в Берлине и Лувене (Бельгия). В 1902 г. защитил докторскую диссертацию. Профессор Варшавского (1915-1939) университета, ректор Варшавского университета (1922/23 и 1931/1932), академик Польской АН с 1937 г. После Второй мировой войны в 1946 г. получает кафедру математической логики в Королевской Ирландской академии в Дублине.

В 20-е годы в Варшаве Лукасевич вел семинар, на котором разработаны основные понятия металогики. Среди его студентов были А.Тарский, А.Линденбаум, Б.Собочиньский, М.Вайсберг и др. После войны его учениками по существу стали К.А.Мередит и А.Н.Прайор. Последний писал в предисловии к своей книге «Время и модальность» *[Prior* 1957], что она целиком обязана Лукасевичу.

Целью логических исследований Лукасевич считал прежде всего разработку точных методов анализа философских рассуждений. В основание философии может быть положена «научная метафизика», или общая теория предметов, но не эпистемология в духе Декарта или Канта, ибо такой путь, по мнению Лукасевича, ведет в тупик. Выход из тупика - в **применении логической методологии, позволяющей свести к минимуму число исходных философских понятий, обладающих очевидностью и интуитивной ясностью, чтобы затем через них строго определять философские понятия «пространственно-временной структуры мира», «причинности», «детерминизма», «индетерминизма» и др. Таким образом, логика дает методологический образец для философии (в частности, аксиоматико-дедуктивный метод).** Лукасевич весьма скептически относился к попыткам построения всеобъемлющих философских систем. Критикуя психологизм и априоризм в логике, он **выдвинул идею логического плюрализма: различные логические системы способны эксплицировать различные онтологические теории**
Например, **классическая двузначная логика эксплицирует принцип «жесткого» детерминизма в философском и научном мышлении, тогда как переход к *многозначной логике* позволят проводить корректные «индетерминистские» рассуждения.**
Основные результаты Лукасевича лежат в **области математической логики**. Ему принадлежат важные результаты **в области классической (теория дедукции и аксиоматизация), интуиционистской, модальной, импликативной и вероятностной логики.** Он провел ряд исследований по проблемам аксиоматизации формализованной силлогистики, по истории логики (силлогистика Аристотеля, логика древних стоиков); им введена оригинальная бесскобочная запись логико-математических формул. Впервые в мире в 1910 г. (одновременно с Н.А.Васильевым) им был подвергнут критике *закон непротиворечия* (см. *[Lukasiewicz* 1971]).
Однако главным его результатом было создание *новой логики,* с которой он связывал «борьбу за освобождение человеческого духа» *[Lukasiewicz* 1918]. Как отмечает Е.Слупецкий в предисловии к тому избранных работ Лукасевича *[Lukasiewicz* 1970], «...Проблемой, которой Лукасевич посвятил почти всю свою жизнь и которую он стремился разрешить, прилагая необычайные усилия и проявляя огромный энтузиазм, была проблема детерминизма. Именно эта проблема вдохновила его на изумительную идею многозначных логик» (p. *vii).* См. также монографию Я.Воленского *[Wolenski* 1989].
Широкую известность принесла Лукасевичу публикация в 1920 г. первой системы многозначной логики, а именно трехзначной

[Lukasiewicz 1920]. Непосредственным философским основанием для этого явилось опровержениие Лукасевичем аристотелевской доктрины *логического фатализма.*

2.2. Логический фатализм

Философская доктрина, утверждающая, что из одних законов (принципов) логики следует, что всё в мире предопределено и поэтому человек не имеет свободы воли, получила название доктрины логического фатализма. Аргумент логического фатализма с целью его опровержения впервые был изобретен Аристотелем (IV в. до н.э.) в его знаменитой 9-й главе трактата «Об истолковании» *[Аристотель* 1978].
Сам аргумент можно представить в следующем виде. Предположим, сейчас истинно, что завтра будет морское сражение. Из этого следует, что не может быть, чтобы завтра не было морского сражения. Следовательно, необходимо, что завтра морское сражение произойдет *(принцип необходимости).* Подобно этому, если сейчас ложно, что завтра будет морское сражение, то необходимо, что морское сражение завтра не произойдет. Но само высказывание о том, что завтра произойдет морское сражение, сейчас либо истинно либо ложно *(логический принцип двузначности).* Следовательно, или необходимо, что морское сражение завтра произойдет, или необходимо, что морское сражение завтра не произойдет. Обобщив этот аргумент, получаем, что всё происходит по необходимости и нет ни случайных событий, ни свободы воли.
Логическая структура данного аргумента выглядит следующим образом. Пусть *«p»* есть высказывание о будущем случайном событии; *~p* - высказывание, противоречащее *p,* и читается как «не-*p*»; *T(p)* и *F(p)* обозначают соответственно «истинно, что *p»* и «ложно, что *p*»; N(*p*) обозначает «необходимо, что *p».* Тогда имеем:

(1) $T(p) \to N(p)$ - принцип необходимости,

(2) $F(p) \to N(\sim p)$ - по аналогии с (1),

(3) $T(p) \vee F(p)$ - принцип двузначности,

(4) $N(p) \vee N(\sim p)$ - из (1), (2) и (3) по правилу классической логики, которое называется «сложная конструктивная дилемма»: из $A \to C$, $B \to D$ и $A \vee B$ следует $C \vee D$.

В основе приведенного фаталистического аргумента лежат две посылки: (1) и (3). Первую посылку, утверждающую, что *если истинно, то необходимо,* Лукасевич называет «принципом необ-

ходимости» *[Лукасевич* 1959], который безоговорочно принимался всеми элинистическими философскими школами. Посылка (3) является «принципом двузначности» (бивалентности), утверждающим, что *каждое высказывание или истинно или ложно*. Принятие принципа двузначности в античности было тесно связано с доктриной детерминизма (фатализма). Эпикурейцы, которые были индетерминистами, отрицали принцип двузначности, в то время как стоики, и прежде всего Хризипп, являющиеся последовательными детерминистами, учили, что все высказывания, в том числе и высказывания о будущих случайностях, должны быть истинными или ложными, и считали это утверждение направленным против Аристотеля *[Mates* 1953]. Большинство комментаторов и исследователей считают, что Аристотель ограничивает применимость принципа двузначности относительно высказываний о будущих случайных событиях и этим разрушает фаталистический аргумент. Одновременно с этим возникает проблема истинностного статуса высказываний о будущих случайных событиях.

Лукасевич выразил суть логического фатализма (детерминизма) у Аристотеля весьма однозначно: «Аристотель верит, что детерминизм был бы неизбежным следствием из этого закона (закона двузначности. - *А.К.),* следствием, которого он не может принять. Поэтому он вынужден ограничить этот закон. Однако он не сделал этого достаточно убедительно и по этой причине его способ разрешения вопроса не совсем ясен» *[Lukasiewicz* 1930]. Зато совершенно ясно это сделано самим Лукасевичем, что и привело к созданию первой системы многозначной логики.

2.3. Введение в логику третьего истинностного значения

Суть новаторской идеи Лукасевича заключается в том, что в логику вводится третье истинностное значение, промежуточное между «истиной» и «ложью» и интерпретируемое им как «безразлично». В напутсвенной речи, произнесенной в Варшавском университете 7 марта 1918 г., Лукасевич скажет, что уже в 1910 г. он пытался сконструировать не-аристотелеву логику, но безрезультатно. Однако теперь (летом 1917 г.) ему удалось *доказать,* что «кроме истинных и ложных высказываний существуют возможные высказывания, к которым объективная возможность относится как нечто третье в добавление к существованию и несуществованию. **Это позволяет установить систему трехзначной логики**...» *[Lukasiewicz* 1918].

В своей статье «О детерминизме» Лукасевич даёт философское обоснование введения в логику третьего истинностного значения. Здесь Лукасевич обосновывает, что **существуют будущие факты, для которых еще нет соответствующих фактов в настоящем, т.е. нет ничего, что с необходимостью заставило бы нас принять высказывание о таком будущем факте как истинное. Но, с другой стороны, мы не можем утверждать, что такое высказывание ложно, если в настоящее время не существует факта, являющегося причиной того, что будущий факт не произойдет. Такие высказывания Лукасевич называет в этой статье «безразличными» и делает важное заключение, что альтернатива, составленная из двух подобных высказываний,** например, «Ян будет завтра в полдень дома, либо Яна завтра не будет в полдень дома», **должна быть истинна согласно закону исключенного третьего** (см.выше 1.3). Лукасевич утверждает, что аристотелевское решение проблемы, по-видимому, состоит в том, что альтернатива «завтра произойдет морское сражение или завтра не произойдет морское сражение» уже сегодня истинна, но ни высказывание «завтра будет морское сражение», ни высказывание «завтра не будет морское сражение» сегодня не являются истинными. Эти высказывания касаются будущих случайных событий и, как таковые, они *ни истинны и ни ложны.*

Предложив такую интерпретацию Аристотеля, Лукасевич, однако, заключает, что доводы Аристотеля подрывают не столько закон исключенного третьего, сколько один из глубочайших принципов всей нашей логики, который им же впервые и установлен, а именно, что каждое высказывание *либо истинно, либо ложно.* **Этот принцип Лукасевич называет *принципом бивалентности*.** Поскольку принцип бивалентности лежит в самих основах логики, он не может быть доказан. «Ему можно лишь доверять, а доверяет ему тот, кому он кажется очевидным. Поэтому мне ничто не препятствует этот принцип не признать и принять, что кроме истинности и ложности существуют еще другие логические значения, по крайней мере еще одно, *третье* логическое значение. [...] Вводя это третье значение в логику, мы изменяем её до основания. Трехзначная система логики... отличается от обычной до сих пор известной двузначной логики в не меньшей степени, нежели системы неэвклидовой геометрии отличаются от эвклидовой геометрии» *[Лукасевич* 1993].

Сейчас можно с уверенностью сказать, что подобная высокая оценка относительно создания новой логики вполне соответствует действительности.

2.4. Истинностные таблицы. Аксиоматизация

К основным проблемам при построении многозначных логик относятся, во-первых, определение логических связок, во-вторых, их содержательная интерпретация и, в-третьих, самое сложное, интерпретация самих истинностных значений. Последнюю мы уже рассмотрели (и еще к ней вернемся), вторая проблема не нашла своего решения у Лукасевича, а решение первой предпринято им в статье «О трехзначной логике» *[Lukasiewicz* 1920].

Оставляя классические значения для импликации → и отрицания ~, когда аргументы принимают значения из множества {0, 1}, Лукасевич следующим образом доопределяет логические связки:

$(1 \to {}^1/_2) = ({}^1/_2 \to 0) = {}^1/_2,$

$(0 \to {}^1/_2) = ({}^1/_2 \to {}^1/_2) = ({}^1/_2 \to 1) = 1,$

$\sim {}^1/_2 = {}^1/_2.$

Посредством исходных связок определяются другие логические связки.

$p \vee q = (p \to q) \to q$ **(дизъюнкция),**

$p \wedge q = \sim(\sim p \vee \sim q)$ **(конъюнкция),**

$p \equiv q = (p \to q) \wedge (q \to p)$ **(эквивалентность).**

Тогда истинностные таблицы для логических связок выглядят так:

p	$\sim p$
1	0
${}^1/_2$	${}^1/_2$
0	1

$\to$	1	${}^1/_2$	0
1	1	${}^1/_2$	0
${}^1/_2$	1	1	${}^1/_2$
0	1	1	1

$\wedge$	1	${}^1/_2$	0
1	1	${}^1/_2$	0
${}^1/_2$	${}^1/_2$	${}^1/_2$	0
0	0	0	0

$\vee$	1	${}^1/_2$	0
1	1	1	1
${}^1/_2$	1	${}^1/_2$	${}^1/_2$
0	1	${}^1/_2$	0

$\equiv$	1	${}^1/_2$	0
1	1	${}^1/_2$	0
${}^1/_2$	${}^1/_2$	1	${}^1/_2$
0	0	${}^1/_2$	1

Оценка множества формул *For* в трехзначной логике Лукасевича есть функция v: $For \to \{0, {}^1/_2, 1\}$, «совместимая» с приведенными выше таблицами. Формула A называется *тавтологией,* если при любой оценке v принимает *выделенное значение* 1. Множество данных тавтологий называется трехзначной (матричной) логикой Лукасевича и обозначается посредством $Ł_3$.

Первая аксиоматизация множества тавтологий $Ł_3$ принадлежит ученику Лукасевича М.Вайсбергу *[Wajsberg* 1931]:

1. $(p \to q) \to ((q \to r) \to (p \to r))$
2. $p \to (q \to p)$
3. $(\sim p \to \sim q) \to (q \to p)$
4. $((p \to \sim p) \to p) \to p.$

Правила вывода такие же, как и для классичекой логики:

Rl. Modus ponens.

R2. Подстановка.

Аксиоматизация Вайсберга означает, что для $Ł_3$, как и для C_2, имеет место

Теорема адекватности. *Для всякой формулы* A, $\vdash A$ в $Ł_3$ *т.т.т., когда* $\models A$ *в* $Ł_3$.

(В *[Epstein* 1990] имеется также доказательство теоремы адекватности для $Ł_3$ в виде $\Gamma \vdash A$ *т.т.т., когда* $\Gamma \models A$.*)*

Таким образом, как и классическая логика, исчисление $Ł_3$ непротиворечиво и дедуктивно полно. На этом фундаментальные сходства между C_2 и $Ł_3$ заканчиваются.

2.5. Отличия трехзначной логики Лукасевича $Ł_3$ от классической

Обратим внимание на одно весьма важное свойство истинностных таблиц для $Ł_3$, а именно: на классическом можестве истинностных значений, т.е. на множестве {1, 0} определение логических связок $Ł_3$ совпадает с определением связок классической двузначной логики C_2. Отсюда следует, что любая тавтология $Ł_3$ есть тавтология C_2, *но не наоборот.*

Например, легко проверить, что закон сокращения

$(p \to (p \to q)) \to (p \to q)$

не есть тавтология в $Ł_3$.

(Таким образом, трехзначная логика Лукасевича является исторически первым примером логик без сокращения, которые привлекли к себе большое внимание. См. *[Ono & Komori* 1985], где также обсуждаются работы Лукасевича.)

Обратим внимание, что если в аксиоматизации Вайсберга аксиому (4) заменить на закон сокращения, то получим аксиоматизацию $\mathbf{C}_2$. Это следует из того факта, что из аксиом Вайсберга (1), (2) и *закона сокращения* выводима *самодистрибутивность.* Таким образом, аксиоматизацию $\mathbf{Ł}_3$ можно представить как замену в аксиоматизации $\mathbf{C}_2$ закона сокращения на аксиому Вайсберга (4).

На самом деле **введение Лукасевичем в логику третьего истинностного значения, промежуточного между истиной и ложью, имело весьма радикальные последствия для самой логики, самым важным из которых оказалось то, что *ни закон исключенного третьего* $p \vee \sim p$, ни закон непротиворечия $\sim(p \wedge \sim p)$ не являются законами $\mathbf{Ł}_3$; эти формулы принимают значение $^1/_2$, когда p имеет значение $^1/_2$.**

Однако если мы отождествим третье истинностное значение ½ *с* 1, т. е. будем рассматривать $\mathbf{Ł}_3$ с двумя выделенными значениями, то формулы $p \vee \sim p$ и $\sim(p \wedge \sim p)$ станут тавтологиями. Долгое время считалось, что при таком рассмотрении множество тавтологий $\mathbf{Ł}_3$ совпадает с множеством тавтологий $\boldsymbol{C_2}$, пока А.Тюркетт (см *[Rescher* 1969]) не нашел контрпример

$$\sim(p \to \sim p) \vee \sim(\sim p \to p)$$

Эта формула является классической тавтологией, но в $\mathbf{Ł}_3$, когда p принимает значение 1/2, вся формула принимает значение 0. Заметим, что данная формула эквивалентна формуле $\sim(p \equiv \sim p)$.

Реакция на подобную ревизию классической логики была весьма неоднозначной, наиболее важные содержательные аспекты которой мы рассмотрим ниже.

Наиболее *существенное* отличие $\mathbf{Ł}_3$ от $\mathbf{C}_2$ состоит в следующем. Как явствует из раздела (1.3), классическая двузначная логика является функционально полной. В $\mathbf{Ł}_3$ это не так, однако если к последней добавить оператор Слупецкого Tp, который переводит любое значение p в 1/2

p	Tp
1	$^1/_2$
$^1/_2$	$^1/_2$
0	$^1/_2$

то получим функционально полную трехзначную логику, которую обозначим посредством $\mathbf{Ł}_3^T$.

Теперь, если к аксиомам Вайсберга для $\mathbf{Ł}_3$ добавить две аксиомы, содержащие оператор Слупецкого Tp:

5. $\mathrm{T}p \to \sim \mathrm{T}p$

6. $\sim \mathrm{T}p \to \mathrm{T}p$,

то получим аксиоматизацию трехзначной логики Слупецкого $\mathbf{Ł}_3^T$ *[Slupecki* 1936]. Заметим, что в классической логике никакие формулы вида $A \supset \neg A$ и $\neg A \supset A$ не являются тавтологиями.

В дальнейшем мы уточним *функциональные свойства* функционально не полной $\mathbf{Ł}_3$. При обобщении $\mathbf{Ł}_3$ на произвольный конечный случай, т.е. на $\mathbf{Ł}_n$ $(n \leq 2,\ n \in \mathrm{N})$, функциональные свойства последней окажутся решающим моментом для всего нашего исследования.

2.6. Трехзначная модальная логика Лукасевича

Обратим внимание еще на одну особенность $\mathbf{Ł}_3$, которая состоит в том, что теперь мы можем конструировать новые логические связки, не существующие в $\mathbf{C}_2$. Этот факт для Лукасевича оказался весьма существенным, поскольку он показал, что в рамках двузначной логики нельзя построить модальную логику, но теперь, введя в логику третье истинностное значение, Лукасевич ставит задачу дать такое определение оператора возможности Mp, чтобы для всех теорем о модальных предложениях, начиная от Аристотеля и вплоть до Лейбница, существовала бы по крайней мере одна интерпретация в трехзначной логике $\mathbf{Ł}_3$, посредством которой каждая такая теорема была бы истинной *[Lukasiewicz* 1930]. Все эти теоремы были в итоге сведены Лукасевичем к трем группам:

(I) $\sim \mathrm{M}p \to \sim p$

(II) $\sim p \to (\sim p \to \sim \mathrm{M}p)$

(III) $\mathrm{M}p \wedge \mathrm{M}\sim p$ для некоторого p.

Тщательно анализируя эти три утверждения, Лукасевич показывает, что в рамках классической двузначной логики мы приходим к противоречию и что ответственность за противоречие несет логический *принцип двузначности* (бивалентности). Размышления о статусе этого принципа, а также анализ высказываний о будущих случайных событиях опять приводят Лукасевича к идее введения в логику третьего истинностного значения, которое он окончательно интерпретирует как «возможность» *[Lukasiewicz* 1930].

Теперь остается только найти подходящее определение модального оператора возможности M*p* в рамках трехзначной логики **Ł3,** что и было сделано в 1921 г. учеником Лукасевича А.Тарским *[Lukasiewicz* 1930].

$$Mp = \sim p \rightarrow p,$$

т.е. «возможно, что *p*» означает «если не-*p*, то *p*». Оператор необходимости L*p* определяется через исходный оператор M*p* обычным образом: $Lp = \sim M \sim p$. На основе этих определений строятся истинностные таблицы для M*p* и L*p:*

p	M*p*	L*p*
1	1	1
1/2	1	0
0	0	0

Таким образом, построение модальной логики явилось еще одним источником появления трехзначной логики Лукасевича.

Обратим внимание, что можно ввести и другие модальные операторы, наиболее интересным из которых является оператор *случайности:*

$$Qp = Mp \wedge M \sim p,$$

который «выделяет» третье истинностное значение (см. выше утверждение III):

p	Q*p*
1	0
1/2	1
0	0

В *[Epstein* 1990] этот оператор интерпретируется как «неопределенно, что... », обозначается посредством «I*p*» и определяется так $Ip = p \equiv \sim p$.

Это позволяет, как отмечает Г.Малиновский *[Malinovski* 1993], сформулировать в **Ł3** аналоги закона исключенного третьего и закона непротиворечия:

$$p \vee \sim p \vee Qp$$

$$\sim(p \wedge \sim p \wedge \sim Qp).$$

Между свойствми модальных операторов **Ł3** и модальных операторов системы Льюиса **S5** имеется некоторое сходство, которое нашло свое точное выражение в работе Р.Вудруффа *[Woodruff* 1974], где дан перевод **Ł3** в **S5**. Таким образом, **Ł3** можно проинтерпретировать посредством **S5.**

2.7. Трудности интуитивной интерпретации Ł3

С формальной точки зрения трехзначная логика Лукасевича выглядит безупречной: показана её непротиворечивость, т.е. в **Ł3** недоказуема некоторая формула *A* вместе со своим отрицанием *~A,* доказана дедуктивная полнота **Ł3** и, как и классическая логика, **Ł3** является разрешимой. Но поскольку построение **Ł3,** т.е. введение в логику третьего истинностного значения, имело сугубо содержательные предпосылки, а именно идею отразить в логической форме индетерминистский статус высказываний о будущих случайных событиях и таким образом опровергнуть фаталистический аргумент Аристотеля, то встает нетривиальный вопрос**: насколько формальные свойства Ł3 оказались адекватными для выражения этой идеи. И вот здесь как раз возникают весьма серьезные затруднения.**

Приданию глубокого философского смысла трехзначной модальной логике Лукасевича посвящена статья создателя **временной логики** А.Н.Прайора *[Prior* 1953], который задает следующий вопрос: является ли этот модальный язык действительно подходящим для экспликации тех проблем, которые имел в виду Лукасевич. Ответ на этот вопрос, подчеркиват Прайор, зависит от интерпретации, которую мы придаем истинностным значениям **Ł3**. В этой работе **Прайор впервые обращается к идее овремененных высказываний, истинностные значения которых могут изменяться во времени, что было общепринято в античности и в средние века, как он отмечает, но совсем забыто в наше время**. По мнению Прайора, Аристотель в девятой главе трактата «Об истолковании» пытается преодолеть истинную трудность - возможность использовать высказывания во

вневременном смысле для описания событий типа «завтрашнее морское сражение». И Прайор делает вывод, что Аристотель говорит о некоторых высказываниях о будущем, как не являющихся ни истинными, ни ложными, поскольку еще нет определенного факта, с которым эти высказывания можно соотнести; однако как утверждение, так и отрицание подобных высказываний потенциально истинно или потенциально ложно, но не актуально истинно или ложно. Когда же эта потенциальность исчезает со временем, тогда значение «1» приписывается высказываниям определенно истинным, т.е. при описании будущих событий как предопределенных или событий, которые уже стали настоящими или прошлыми. **Такие высказывания и являются «необходимыми»**. Таким образом, утверждение высказываний о состоянии дел в настоящем и прошлом и утверждение их «необходимости» являются эквивалентными в $Ł_3$.

Что же касается вообще свойства модальных операторов в $Ł_3$ не принимать третьего истинностного значения, замечает Прайор, то такая особенность достаточно хорошо согласуется с нашим интуитивным понятием «возможности» как того, что каким-то образом оказывается реальным даже тогда, когда того, возможностью чего она является, еще нет. Следствием этого и является двузначный характер модальной части $Ł_3$. В итоге Прайор очень высоко оценивает создание Лукасевичем модальной логики $Ł_3$ и считает, что Лукасевич сделал для аристотелевской проблемы логического фатализма то же, что он сделал для аристотелевской теории силлогистики.

Казалось бы, все трудности преодолены, но дело в том, и это признается всеми сторонниками традиционной интерпретации (в том числе и Лукасевичем), что Аристотель явно утверждал, что альтернатива $p \vee \sim p$ в любом случае является всегда истинной. Однако в $Ł_3$, как уже отмечалось, закон исключенного третьего не имеет места. Как предполагает Прайор, Аристотель сказал бы, что обычно $(p \vee q) = {}^1/_2$ при $p = {}^1/_2$ и $q = {}^1/_2$, но если q в $p \vee q$ становится $\sim p$, тогда альтернатива принимает значение не 1/2, а 1. Это позволяет Прайору заключить, что в предполагаемой трехзначной логике Аристотеля дизъюнкция не была бы истинностно-функциональной *[Prior* 1953]. В целом ряде более поздних работ, рассматривающих проблему логической экспликации высказываний о будущих случайных событиях посредством $Ł_3$, расхождение между Лукасевичем и Аристотелем по поводу $p \vee \sim p$ служит основным возражением против адекватности $Ł_3$ для решения аристотелевской проблемы.

Таким образом, хотя Лукасевич и ввел строгое различие между принципом бивалентности и законом исключенного третьего, но в его $Ł_3$ не принимается ни то, ни другое, что привело к неадекватной экспликации аристотелевской проблемы. Как замечает Прайор *[Prior* 1955], а затем С. Мак-Колл *[McCall* 1966], положение можно было бы исправить, если определить дизъюнкцию не так, как это сделал Лукасевич: $p \vee q = (p \to q) \to q$, а по-другому, как это обычно делается в классической логике: $p \vee q = \sim p \to q$. Тогда $p \vee q \neq \max(v(p), v(q))$, поскольку ${}^1/_2 \vee {}^1/_2 = 1$, но теперь $p \vee \sim p$ будет законом в $Ł_3$, поскольку в этом случае ${}^1/_2 \vee \sim {}^1/_2 = 1$.

(Это как раз один из тех случаев, когда принцип двузначности нарушен, а закон исключенного третьего имеет место. Таким образом, разница между ними является принципиальной.)

Но тогда не является законом $(p \vee p) \to p$, и это, предполагает Мак-Колл, послужило причиной, по которой Лукасевич выбрал первое определение. И правда, каким образом можно обосновать дизъюнкцию со свойством ${}^1/_2 \vee {}^1/_2 = 1$? Однако эти примеры указывают на другую особенность $Ł_3$, которая заключается в том, что уже в трехзначной логике можно обобщать свойства классических связок по-разному, в результате чего возможны, например, различные дизъюнкции, в то время как в $\mathbf{C}_2$ (как мы видели):

$$p \vee q = \neg p \supset q = (p \supset q) \supset q.$$

Неожиданно выяснилось, и самым очевидным образом, что $Ł_3$ имеет свойство, которое уже безотносительно к аристотелевской проблеме делает ее уязвимой. Т.Сугихара *[Sugihara* 1954] в рецензии на статью Прайора *[Prior* 1953] замечает, что не только формула $p \vee \sim p = {}^1/_2$, когда $p = {}^1/_2$, но и дуальная ей формула, а именно $p \wedge \sim p$ тоже принимает значение 1/2, когда p =1/2, т.е. закон непротиворечия $\sim(p \wedge \sim p)$ не является законом $Ł_3$. Это говорит о том, продолжает Сугихара, что невозможно проинтерпретировать $Ł_3$ в терминах случайности. При этом Сугихара ссылается на возражение против $Ł_3$, сделанное Ф.Гонсетом и А.Мостовским. Возражение Гонсета, считающееся неопровержимым доводом против содержательной интерпретации $Ł_3$, состоит в следующем. На конференции в Цюрихе (1938 г.) «Проблема обоснования и метод математических наук» при обсуждении доклада Лукасевича (см. *[Lukasiewicz* 1941]) Гонсет обратил внимание, что следующее конъюнктивное высказывание, приведенное Лукасевичем: «Через год я буду в Варшаве и через год я не буду в Варшаве» - имеет истинностное значение 1/2, поскольку само высказывание «Через год я буду в

Варшаве» имеет истинностное значение 1/2 в интерпретации Лукасевича, а операции отрицания и конъюнкции это значение не меняют. Однако совершенно ясно, замечает Гонсет, что такое конъюнктивное утверждение должно быть ложным сейчас. Х.Карри *[Curry* 1941] в рецензии на доклад Лукасевича отмечает, что остаются неясными те «интуитивные модальные основания», которые, по утверждению Лукасевича, должны представлять $Ł_3$. При этом Карри ссылается на уже упомянутое обсуждение доклада. Это же возражение приводит М.Бохеньский *[Bochehski* 1949] в рецензии на две работы К.Клозака. Клозак утверждает, что закон исключенного третьего всегда истинен, на что Бохеньский возражает, говоря, что данное утверждение не признается сторонниками многозначных логик. А более сильным аргументом против таких логик Бохеньский как раз считает замечание Гонсета о том, что закон непротиворечия тоже не имеет в них места.
Годом позже Мостовский *[Mostowski* 1950] в рецензии на статью Е.Расевой о многозначных логиках Лукасевича отмечает, что хотя автор дает некоторые иллюстративные примеры, соответствующие интерпретации 1/2 как «возможности», но обходит молчанием весьма важное замечание Гонсета.
(Здесь Мостовский предлагает интерпретировать $^1/_2$ более подходящим образом, а именно как «случайность». Такую же интерпретацию принимает и *А.Черч[Чёрч* 1960].
Как считает Мостовский, «это замечание разрушает всякую надежду, что будет когда-либо возможно найти разумную интерпретацию трехзначной логики Лукасевича в терминах обыденного языка».
После Сугихары резкой критике содержательную интерпретацию L_3 подверг Мо Шо-Куэй *[Mox Shaw-Kwei* 1954]. Отмечая, что под возможными высказываниями Лукасевич подразумевает (следуя Аристотелю) высказывания, относящиеся к будущему времени, Шо-Куэй делает такое заключение: «Мы видим, что следующие высказывания не соответствуют нашей интуиции: $^1/_2 \equiv {\sim}^1/_2$, $^1/_2 \equiv {}^1/_2 \wedge {\sim}^1/_2$, $^1/_2 \equiv {}^1/_2 \vee {\sim}^1/_2$. Ибо мы рассматриваем высказывание, независимо от того, относится ли оно к будущему или нет, как никогда не эквивалентное своему отрицанию; и мы считаем конъюнкцию высказывания и его отрицания всегда ложной, а их альтернативу всегда истинной, а не возможностью».
Интересно, что Прайор несколько позже *[Prior* 1957] в очерке на страницах иллюстрированного литературно-политического еженедельника «Listener» отмечает, что никто еще не дал удовлетворительного формального представления связей между временем и модальностями, хотя ближе всех к этому подошел

Лукасевич. Прайор считает, что достижением Лукасевича, имеющим большие последствия для логики, является его $Ł_3$. Как раз здесь Прайор подробно разбирает возражение Гонсета (без ссылки на него), замечая, что по обычным двузначным допущениям ни одна логическая связка не является более очевидной и истинностно-функциональной, чем $\wedge$, но трудно сохранить этот истинностно-функциональный характер $\wedge$ в $Ł_3$.
Позднее Т.Чэпмен *[Chapman* 1972] возражает против $Ł_3$ как средства для преодоления явной несовместимости между индетерминизмом и двузначным характером любых высказываний на том основании, что в $Ł_3$ закон непротиворечия не имеет места. На то же самое, но десятью годами раньше, указывают известные историки логики В. и М. Ниль *[Kneale* 1962], в силу чего они вообще считают логику $Ł_3$ неприемлемой.
Таким образом, основная трудность при содержательной интерпретации $Ł_3$ заключается в строго истинностно-функциональном характере логических связок $\wedge$ и $\vee$, а попытки их истолкования в не истинностно-функциональном смысле, что является следствием рассмотрения формально-логических свойств $Ł_3$ в контексте фундаментального философского понятия «возможности» («случайности»), требуют построения иных логик.
На другую трудность, вызванную этим смыслом нового истинностного значения, обратил внимание Т.Котарбиньский *[Котарбиньский* 1963]. Отметив, что вопрос о роли многозначной логики для защиты индетерминизма является спорным, Котарбиньский продолжает: «Довольно загадочной остается и проблема интерпретации как знака половинчатости (а также других знаков логических значений в *n*-значных системах), так и существующего в системе Лукасевича функтора М, читающегося "возможно, что... ". Ведь именно он должен вводить понятие возможности в исчисление высказываний, но, с другой стороны, и знак половинчатости тоже должен говорить о какой-то "возможности" высказывания, логическим значением которого он является». Таким образом, Котарбиньский указывает на несовместимость в одной логической системе двух разных видов возможности (оба встречающихся у Аристотеля), один из которых есть «билатеральная возможность» (двусторонняя), относящаяся к будущим случайным событиям, а другой вид возможности - «унитеральная возможность» (односторонняя), описываемая обычными свойствами оператора М, например, $p \rightarrow Mp$ и не верно, что $p \rightarrow {\sim}M{\sim}p$, т.е. не верно, что $p \rightarrow Lp$. Однако в L_3 имеет место $p \rightarrow (p \rightarrow Lp)$.

Наиболее интересной попыткой дать интуитивную интерпретацию L_3 является статья Е.Слупецкого, Е.Брыля и Т.Пруцналя *[Slupecki, Bryll, Prucnal* 1967] (см. также *[Слупецкий* 1974]). Слупецкий исходит из работы Лукасевича «О детерминизме», которая не была известна Прайору. Кратко суть этой интерпретации состоит в следующем. Все события **Z** разделены на прошлые, настоящие и будущие и предполагается, что операции над событиями, а именно сложение $\cup$, умножение $\cap$ и дополнение $\bar{}$ удовлетворяют аксиомам булевой алгебры, как это принято в теории вероятностей, т.е. структура $\mathbb{Z} = \langle Z, \cup, \cap, \bar{} \rangle$ есть булева алгебра. Событие является фактически детерминированным, если существует в прошлом или в настоящем факт, являющийся его причиной; и событие недетерминировано, если неверно, что оно детерминировано и в то же время неверно, что противоположное событие является детерминированным. Отсюда утверждение об истинности высказывания *p*, описывающего событие E, эквивалентно утверждению о детерминированности этого события и т.д. Слупецкий отмечает, что такое понимание логических значений совпадает с намерениями Лукасевича. Предполагается, например, что дизъюнкция двух высказываний описывает сумму событий, описанных ее аргументами, и т.д. Исходя из этого, а также приняв некоторые естественные утверждения о причинных связях, обосновываются таблицы истинности, которые в точности совпадают с матричными определениями дизъюнкции, конъюнкции и отрицания в $Ł_3$. Однако более тщательный анализ, проведенный М.Новаком *[Nowak* 1988], показывает, что допущения относительно $\mathbb{Z}$ должны быть модифицированы так, чтобы $\mathbb{Z}$ выглядела как решетка де Моргана (см. выше 1.5), а не как алгебра Буля.

Как уже говорилось, **исходными операциями в $Ł_3$ являются отрицание и импликация, через которые и определяются дизъюнкция, конъюнкция и модальные операторы.** Но легко видеть, что посредством отрицания, дизъюнкции и конъюнкции нельзя определить импликацию в $Ł_3$. Более того, при выделенном значении «1» множество тавтологий в системе с исходными связками $\{\sim, \vee, \wedge\}$ будет пусто и поэтому, считает Слупецкий, эта система не представляет какого-либо интереса. Однако если к этой системе добавить модальные операторы Тарского, то получим логику **S3**, функционально эквивалентную $Ł_3$. Чтобы это показать, достаточно через систему связок логики **S3** выразить импликацию $p \to q$ из $Ł_3$. Слупецкий это делает следующим образом:

$$p \to q = (\sim p \vee q) \vee \mathbf{M}(\sim p \wedge q)$$

замечая по этому поводу, что смысл данного выражения, т.е. смысл импликации Лукасевича, довольно-таки неуловим. И поэтому в указанной работе решается проблема, поставленная Слупецким еше в начале 60-х годов, об аксиоматизации трехзначной логики Лукасевича $Ł_3$ со множествами исходных связок $\{\vee, \sim, \mathbf{L}\}$, $\{\vee, \sim, \mathbf{M}\}$, $\{\wedge, \sim, \mathbf{L}\}$, $\{\wedge, \sim, \mathbf{M},\}$. В итоге, как отмечалось выше, дается аксиоматизация $Ł_3$ в сигнатуре $\{\vee, \sim, \mathbf{N}\}$.

Но Слупецкий обращает внимание и на более существенную трудность. Операции, которые соответствовали бы операторам M и L тем же самым образом, как операции сложения, умножения и дополнения соответствуют $\sim$, $\vee$ и $\wedge$, не существуют в булевой алгебре. Поэтому включая модальные высказывания в трехзначную логику мы должны расширить ранг пропозициональных переменных, до сих пор ограниченный высказываниями, описывающими только события, а это является более сложным и менее интуитивным, замечает Слупецкий. В заключение он отмечает, что хотя интуиции Лукасевича в обосновании трехзначной логики носят общий характер, тем не менее они чрезвычайно глубоки и представляют огромный интерес. Детальный же анализ требует обширных и трудоемких исследований.

Однако вернемся к импликации Лукасевича. Вопрос этот не праздный уже потому, что эта связка является единственной исходной бинарной связкой в $Ł_3$. К сожалению, можно только предполагать, из каких мотивов исходил Лукасевич при определении свойств $p \to q$. Как отмечает Хао Ван *[Wang Hao* 1961], при введении в логику третьего истинностного значения имеются две альтернативы приписывания значения всей импликации $p \to q$, когда $p = {}^1/_2$ и $q = {}^1/_2$ (Хао Ван третье истинное значение интерпретирует как «неопределенно» и обозначает как «u»): в одном случае ${}^1/_2 \to {}^1/_2 = {}^1/_2$, в другом случае ${}^1/_2 \to {}^1/_2 = 1$. В первом случае, $p \to p$ не является больше универсальным логическим законом. Во втором случае $p \to \sim p = 1$, когда $p = 1/2$. Однако эти последствия, замечает Хао Ван, не столько являются основанием для отрицания одной из альтернатив, сколько служат иллюстрацией того, что или мы еще не владеем достаточно хорошим пониманием импликации, примененной к неопределенным высказываниям, или у нас нет надежного руководящего принципа, позволяющего выбирать между двумя этими альтернативами. Лукасевич выбирает ${}^1/_2 \to {}^1/_2 = 1$, продолжает Хао Ван, и в качестве преимущества сохраняет закон $p \to p$. Однако такая интерпретация не позволяет идентифицировать $p \to q$ с $\sim p \vee q$, поскольку $p \vee \sim p$ не является больше универсальным логическим законом. Другая

альтернатива принята С.К.Клини *[Клини* 1957] в его трехзначной логике $\mathbf{K}_3$, где $\sim, \vee$ и $\wedge$ есть в точности логические связки из $\mathbf{Ł_3}$. Посредством этой интерпретации $p \to q = \sim p \vee q$, и поэтому можно развивать трехзначную логику, не включая $\to$ в качестве исходной связки. Но тогда не имеет места $p \to p$.

Обратим внимание, что матричная трехзначная логика $\mathbf{K}_3$ является важным примером алгебры Клини (см. выше 1.5). Поэтому алгебраическая структура $\mathbb{Z}$ должна быть скорректирована от решетки де Моргана до алгебры Клини. Что же касается алгебраической структуры $\mathbf{Ł_3}$, то она намного сложнее (см. раздел 2.10).

Интересно, что в дискуссию о логическом статусе высказываний о будущих случайных событиях, об интерпретации промежуточного (третьего) истинностного значения и об интерпретации самой $\mathbf{Ł_3}$ были втянуты многие виднейшие логики того времени. Общий итог дискуссии оказался весьма критическим относительно возможности дать какую-либо *интуитивно* приемлемую интерпретацию трехзначной логики Лукасевича $\mathbf{Ł_3}$. И для этого, как мы увидим далее, есть серьезные основания. Тем не менее, попытки проинтерпретировать $\mathbf{Ł_3}$ продолжаются и одна из последних принадлежит С.А.Павлову [Павлов 1998], который предложил рассматривать $\mathbf{Ł_3}$ в рамках разработанного им языка логики ложности (с оператором ложности).

2.8. Погружение классической логики в $\mathbf{Ł_3}$

Как говорилось выше, $\mathbf{Ł_3} \subset \mathbf{C_2}$, но тем не менее можно показать, что $\mathbf{Ł_3}$ *богаче* C_2 (!) Покажем, что $\mathbf{Ł_3}$ содержит *трехзначный изоморф* классической логики $\mathbf{C}_2$. Для этого посредством исходных связок $\mathbf{Ł_3}$ определим две новые связки:

$$\lceil p = \sim \mathrm{L} p,$$

$$p \to_1 q = p \to (p \to q),$$

истинностные таблицы для которых выглядят так

p	$\lceil p$
1	0
$^1/_2$	1
0	1

$\to_1$	1	$^1/_2$	0
1	1	$^1/_2$	0
$^1/_2$	1	1	1
0	1	1	1

Обозначим логику со связками $\lceil$ и $\to_1$ как $\mathbf{Ł_3^*}$. Покажем, что множество формул, доказуемых в $\mathbf{Ł_3^*}$, есть в точности множество формул, доказуемых в $\mathbf{C}_2$, т.е. $\mathbf{Ł_3^*}$ есть трехзначный изоморф $\mathbf{C}_2$. Возьмем аксиоматизацию $\mathbf{C}_2$, предложенную Лукасевичем (см. раздел 1.4). Заменим в аксиомах вхождения $\supset$ и $\neg$ на $\to_1$ и $\lceil$ соответственно. Нетрудно проверить табличным способом, что, с одной стороны, эти аксиомы так же имеют место в $\mathbf{Ł_3^*}$, как и правила вывода. С другой стороны, всякая тавтология $\mathbf{Ł_3^*}$ является тавтологией $\mathbf{C}_2$, поскольку истинностные таблицы для $\mathbf{Ł_3^*}$ совпадают с истинностными таблицами для $\mathbf{C}_2$ на множестве {0,1} Таким образом, множества тавтологий $\mathbf{Ł_3^*}$ и $\mathbf{C}_2$ совпадают.

Отсюда следует, что $\mathbf{Ł_3}$ содержит C_2 и, значит, богаче последней.

На самом деле мы показали, что существует *перевод* (погружение) $\mathbf{C}_2$ в $\mathbf{Ł_3}$, т.е. указано отображение * языка C_2 в язык $\mathbf{Ł_3}$:

$$(p)^* = p,$$

$$(A \supset B)^* = A^* \to_1 B^*,$$

$$(\neg \mathrm{A})^* = \lceil (\mathrm{A})^*.$$

Тогда имеет место следующая

Теорема. $\vdash A$ *в* C_2 *т.т.т., когда* A^* *в* $\vdash \mathbf{Ł_3}$.

Истинностная таблица для $\to_1$ впервые и независимо друг от друга была приведена в *[Siupecki, Bryll, Prucnal* 1967] и *[Monteiro A.* 1967]. В первом случае, как мы уже видели, $p \to_1 q$ определяется как $\sim \mathrm{L} p \vee q$ и приводится аксиоматизация $\mathbf{Ł_3}$ с исходными связками $\sim$, L и $\vee$, во втором случае, $p \to_1 q$ определяется как $\mathrm{M} \sim p \vee q$ и отмечается, что в качестве исходных связок для $\mathbf{Ł_3}$ можно взять $\sim, \wedge$ и $\to_1$, определив $p \to q$ как

$$(p \to_1 q) \wedge (\sim q \to_1 \sim p).$$

Приведенное выше определение $p \to_1 q$ как $p \to (p \to q)$ принадлежит Р.Вуйцицкому *[Wojcicki* 1988] и позволяет легко перейти к n-значной логике Лукасевича посредством итерации $"p \to"$ в определении $p \to_1 q$. Вуйцицкий исходит из более общей теоремы М.Токажа *Tokarz* 1971] при доказательстве перевода $\mathbf{C}_2$ в $\mathbf{Ł_3}$. Более простое доказательство имеется в *[Epstein* 1990]. В последней работе отмечается, что для формулировки теоремы дедукции импликация Лукасевича $\to$ не подходит, поскольку $A \wedge \sim A \vDash \sim(A \to B)$ в

$\mathbf{Ł_3}$, но $\nvdash (A \wedge \sim A) \to \sim(A \to B)$ в $\mathbf{Ł_3}$. Но для этого подходит связка $\to_1$:

Если $\Gamma, A \vdash B$, *то* $\Gamma \vdash A \to_1 B$.

Конечно, рассмотренный нами изоморф $\mathbf{C}_2$ не является единственным в $\mathbf{Ł_3}$. Впервые на то, что трехзначная логика может иметь $\mathbf{C}_2$ в качестве изоморфа, было указано Д.А.Бочваром при построении трехзначной логики бессмысленности $\mathbf{B}_3$ *[Бочвар* 1938], предназначенной для разрешения некоторых парадоксов классической математики.

(Заметим, что по своим функциональным свойствам $\mathbf{B_3} \subset \mathbf{Ł_3}$.)

В импликативно-негативной форме этот изоморф выглядит следующим образом:

p	$\lceil p$
1	0
$^1/_2$	1
0	1

$\to^{\mathrm{L}}$	1	$^1/_2$	0
1	1	0	0
$^1/_2$	1	1	1
0	1	1	1

Понятно, что этот изоморф $\mathbf{C}_2$ содержтися в $\mathbf{Ł_3}$: $p \to^{\mathrm{L}} q = \mathrm{L}p \to \mathrm{L}q$. Н.Решер, имея ввиду, что данный изоморф содержтися в $\mathbf{B}_3$, строит (в другой терминологии) также изоморф $\mathbf{C}_2$, содержащийся в $\mathbf{Ł_3}$: $\rceil p = \sim \mathrm{M}p$ *и* $p \to^{\mathrm{M}} q = \mathrm{M}p \to \mathrm{M}q$ *[Rescher* 1969].

(Этот изоморф является также изоморфом, содержащимся в $\mathbf{B}_3$.

Интересно, что комбинация обоих указанных изоморфов, т е. $\lceil$ *и* $\to^{\mathrm{M}}$, дает известную паранепротиворечивую логику Сетте $\mathbf{P}_1$ (см. *[Карпенко* 2000]).

Г.Малиновский *[Malinowski* 1997] приводит еще один изоморф $\mathbf{C}_2$, содержащийся в $\mathbf{Ł_3}$.

(На самом деле, как следует из работы В.Е.Комендантского [Комендантский 2000], $\mathbf{Ł_3}$ содержит 65 (!) изоморфов $\boldsymbol{C_2}$, для построения которых им создана специальная компьютерная программа.)

Из наличия в $\mathbf{Ł_3}$ изоморфа $\boldsymbol{C_2}$ следует, что можно дать аксиоматизацию $\mathbf{Ł_3}$ на основе $\mathbf{C}_2$, т.е. берется аксиоматика $\mathbf{C}_2$ в соответствующем переводе, к ней добавляются аксиомы для дополнительных связок и аксиомы, определяющие взаимоотношение первой группы аксиом со второй. В неявном виде для $\mathbf{Ł_3}$ это и было сделано в работах *[Siupecki, Bryll, Prucnal* 1967] и *[Финн* 1974]. Такой подход положен в основу единого метода аксиоматизации широкого класса

многозначных логик, в том числе и n-значных логик Лукасевича *[Аншаков & Рынков* 1982, 1984] и *[Anshakov & Rychkov* 1984].

2.9. Импликация Лукасевича и трехзначная интуиционистская логика G3

Трехзначная интуиционистская логика $\mathbf{G}_3$ появилась в работе А.Гейтинга *[Heyting* 1930], где впервые было сформулировано пропозициональное (и предикатное) интуиционистское исчисление **H**. Аксиоматизация последнего получается посредством удаления закона исключенного третьего $\vee \neg p$ из аксиоматики классической логики $\mathbf{C}_2$ (см. выше 1.4.).

Матрицы для $\mathbf{G}_3$, появившиеся в результате доказательства независимости **H**, выглядят следующим образом:

p	$\rceil p$
1	0
$^1/_2$	0
0	1

$\Rightarrow$	1	$^1/_2$	0
1	1	$^1/_2$	0
$^1/_2$	1	1	0
0	1	1	1

Истинностные таблицы для $\vee$ *и* $\wedge$ в $\mathbf{G}_3$ в точности совпадают с таблицами для этих связок в $\mathbf{Ł_3}$ и $\mathbf{K}_3$, однако разница между системами связок весьма существенна, поскольку в $\mathbf{G}_3$ через $\rceil p$ и $p \Rightarrow q$ нельзя выразить $p \vee q$ *и* $p \wedge q$. Но

$$p \vee q = ((p \Rightarrow q) \Rightarrow q) \wedge ((q \Rightarrow p) \Rightarrow p).$$

Отсюда следует, что в качестве исходных связок в $\mathbf{G}_3$ можно взять связки $\neg$, $\wedge$ *и* $\Rightarrow$. Легко убедиться, что ни $\neg\neg p \Rightarrow p$, ни $p \vee \neg p$ не являются здесь тавтологиями, хотя первая есть тавтология в $\mathbf{Ł_3}$.

Впервые $\mathbf{G}_3$ была аксиоматизирована Я. Лукасевичем *[Lukasiewicz* 1941]. Она получается за счет добавления к аксиомам интуиционистского пропозиционального исчисления **H** аксиомы

$$(\rceil p \Rightarrow q) \Rightarrow (((q \Rightarrow p) \Rightarrow q) \Rightarrow q)$$

Нетрудно показать, что логические связки $\mathbf{G}_3$ выразимы посредством связок из $\mathbf{Ł_3}$:

$$\rceil p = \sim(\sim p \to p),$$

$$p \Rightarrow q = \rceil(\sim(p \to q)) \vee q.$$

Впервые выразимость связок из $\mathbf{G}_3$ посредством $\mathbf{Ł_3}$ была представлена Г.Мойсилом *[Moisil* 1963], но формула для выразимости $p \Rightarrow q$ гораздо сложнее. У Л.Монтейро *[Monteiro L.* 1970] это выглядит следующим образом:

$$p \Rightarrow q = \mathbf{L}{\sim}p \vee q \vee (\mathbf{M}{\sim}p \wedge \mathbf{M}q).$$

Заметим, что результат Мойсила позволяет дать аксиоматизацию $\mathbf{Ł_3}$ на основе интуиционистской импликации $\Rightarrow$, что и было сделано Л. Итурриоз *[Iturrioz* 1977].

Очевидно, что $\mathbf{G}_3$ не эквивалентна $\mathbf{Ł_3}$, поскольку $\sim p$ нельзя выразить связками из $\mathbf{G}_3$. Таким образом, $\mathbf{G_3} \subset \mathbf{Ł_3}$. Однако если добавить связку ~ к $\mathbf{G}_3$, то, как показал Мойсил *[Moisil* 1963], получим $\mathbf{Ł_3}$:

$$p \to q = (p \Rightarrow q) \vee (\sim q \Rightarrow \sim p)$$

В *[Cignoli* 1982] имеется упрощение:

$$p \to q = (p \Rightarrow q) \vee \sim p.$$

Можно дать другое определение трехзначной импликации Лукасевича, используя наравне с интуиционистской импликацией $\Rightarrow$ *дуальную* ей, которая обозначается посредством $\Leftarrow$ и называется «брауэровой» (см. следующий раздел). Последняя определяется следующим образом *[Monteiro* Л. 1980]:

$$p \Leftarrow q = \sim(\sim p \Rightarrow \sim q).$$

Отрицание $\lceil$, дуальное к $\rceil$, определяется как $\lceil p = p \Leftarrow 1$.

Оказывается, посредством связок $\rceil$, $\lceil$, $\vee$ и $\wedge$ можно определить отрицание Лукасевича *[Cignoli &Monteiro* 1965]:

$$\sim p = \rceil p \vee (p \wedge \lceil p).$$

Более того, посредством этих связок можно определить также импликацию Рейтинга *[Varlet* 1969]:

$$p \Rightarrow q = (\rceil p \vee \rceil\rceil q) \wedge (\lceil p \vee q).$$

Отсюда следует, что трехзначная логика Лукасевича $\mathbf{Ł_3}$ есть в точности *трехзначная* **H-B**-*логика* (см. следующий раздел). Заметим, что в ней импликацию Лукасевича можно определить следующим образом:

$$p \to q = (p \Rightarrow q) \vee \sim(q \Leftarrow p).$$

2.10. Алгебраизация

Для того чтобы перейти к алгебраизации $\mathbf{Ł_3}$, сначала рассмотрим исключительно важный класс алгебр, а именно *алгебры Гейтинга* и

связанные с ними другие алгебраические структуры. Алгебры Гейтинга являются алгебраическим примером *интуиционистской логики [Heyting* 1930] (см. также *[vanDalen* 1986]).

Пусть $\mathbf{x}, \mathbf{y} \in L$. Элемент $\mathbf{z} \in L$ называется *псевдодополнением* элемента x *относительно* y, если z - наибольший элемент со свойством $\mathbf{x} \cap \mathbf{z} \leq \mathbf{y}$. Относительное псевдодополнение обозначается посредством $\mathbf{x} \Rightarrow \mathbf{y}$. Решетка L называется *импликативной* (см. *[Расёва & Сикорскип* 1972]), если $\mathbf{x} \Rightarrow \mathbf{y}$ существует для всех элементов $\mathbf{x}, \mathbf{y} \in L$. Заметим, что решетка с $\Rightarrow$ обладает наибольшим элементом 1, так как для любого $\mathbf{x}$, $\mathbf{x} \Rightarrow \mathbf{x} = \mathbf{1}$; и, главное, решетка с $\Rightarrow$ является дистрибутивной. Каждая импликативная решетка с наименьшим элементом 0 есть *алгебра Рейтинга.* Или, по-другому, алгебры Рейтинга являются решетками с 0, резидуальными относительно пересечения *[Bfyth & Janowith* 1972], где «резидуалом» относительно $\wedge$ является как раз операция $\Rightarrow$, определяемая следующим образом:

$$x \leq y \Rightarrow z \ m.m.m., \ когда \ x \wedge y \leq z.$$

Как эквациональный класс $\langle L, \vee, \wedge, \Rightarrow, 0, 1\rangle$ есть алгебра Рейтинга, если $\langle L, \vee, \wedge, 0, 1\rangle$ есть ограниченная дистрибутивная решетка и для бинарной операции $\Rightarrow$ выполняются следующие три тождества *[Эсакиа* 1985]:

(H1). $\mathbf{x \wedge (x \Rightarrow y) = x \wedge y}$

(H2). $\mathbf{x \wedge (y \Rightarrow z) = x \wedge (x \wedge y \Rightarrow x \wedge z)}$.

(H3). $\mathbf{(x \wedge y \Rightarrow x) \wedge z = z}$.

Очевидно, любая булева алгебра есть алгебра Рейтинга.

Дистрибутивные решетки с операцией $\Rightarrow$ (но в других обозначениях), а также с дуальной к ней операцией $\Leftarrow$ впервые исследовались Т.Сколемом, начиная с 1919 г. (см. *[Карри* 1969]). Такие алгебры Х. Карри называет *сколемовскими структурами.* Алгебры Рейтинга под названием *брауэровы алгебры* были введены Г.Биркгофом в 1940 г. (см. *[Биркгоф* 1984]).

Алгебра $\langle L, \vee, \wedge, \Rightarrow, \Leftarrow, 0, 1\rangle$ называется *дважды* (double) *гейтинговой алгеброй,* или *дважды брауэровой алгеброй,* или *полубулевой алгеброй* (под этим названием она аксиоматизируется в работе *[Rauszer* 1974]) или *алгеброй Сколема [Григолия* 1987], если $\langle L, \vee, \wedge, \Rightarrow, 0, 1\rangle$ есть алгебра Рейтинга, а $\Leftarrow$ есть бинарная операция, дуальная к $\Rightarrow$, т. е. элемент $\mathbf{z}$ $(= \mathbf{x} \Leftarrow \mathbf{y})$ является наименьшим элементом со свойством $\mathbf{x} \cup \mathbf{z} \geq \mathbf{y}$. Операция

$x \Leftarrow y$ в *[Расёва & Сикорский* 1972] называется «псевдоразностью». В *[McKinsey & Tarski* 1946] алгебра $\langle L, \vee, \wedge, \Leftarrow, 0, 1\rangle$ изучается под названием *брауэровой алгебры.* Или, по-другому, алгебры Брауэра являются решетками с 1, резидуальными относительно объединения:

$x \geq y \Rightarrow z$ *т.т.т., когда* $x \vee y \geq z$.

Дважды алгебры были введены, чтобы восстановить принцип дуальности булевой алгебры.

Алгебра $\langle L, \vee, \wedge, \Rightarrow, \sim, 0, 1\rangle$ называется *симметрической алгеброй Рейтинга [Monteiro A.* 1969], если $\langle L, \vee, \wedge, \Rightarrow, 0, 1\rangle$ есть алгебра Гейтинга и $\langle L, \vee, \wedge, \sim, 0, 1\rangle$ есть алгебра де Моргана. Операция ~ на решетке *L* позволяет рассмотреть принцип дуальности: каждое утверждение, доказанное для $\vee$, $\wedge$ и ~, остается истинным, если $\vee$ **и** $\wedge$ заменить соответственно на $\wedge$ **и** $\vee$.Более того, здесь $x \Leftarrow y = \sim(\sim x \Rightarrow \sim y)$. Таким образом, симметрическая алгебра Гейтинга есть дважды алгебра Гейтинга.

Заметим, что в алгебре Гейтинга имеет место $\neg x = x \Rightarrow 0$, где унарная операция $\neg$, называется *псевдодополнением* (интуиционистское отрицание), и $\lceil x = x \Leftarrow 1$, где унарная операция $\lceil$ называется *дуальным псевдодополнением.*

Изучение (дистрибутивных) решеток с псевдодополнением $\neg$, точно так же, как и (дистрибутивных) решеток с инволюцией ~, стало отдельным направлением в области алгебраических структур.

Алгебра $\langle L, \vee, \wedge, \neg, 0, 1\rangle$ называется *p-алгеброй,* если $\langle L, \vee, \wedge, 0, 1\rangle$ есть ограниченная решетка и для любого $x \in L$ элемент $\neg x$ является псевдодополнением элемента x.

Алгебра $\langle L, \vee, \wedge, \neg, \lceil, 0, 1\rangle$ называется *дважды p*-алгеброй, если $\langle L, \vee, \wedge, \neg, 0, 1\rangle$ есть *p*-алгебра и $\langle L, \vee, \wedge, \lceil, 0, 1\rangle$ - дуальная *p*-алгебра.

Дистрибутивная *p*-алгебра называется *стоуновой* (см. *[Гретцер* 1982,]; здесь же приведена соответствующая литература по p-алгебрам), если она удовлетворяет *стоунову тождеству*

$$\neg x \vee \neg\neg x = 1,$$

и *дважды стоунова,* если выполняется также тождество

$$\lceil x \wedge \lceil\lceil x = 0.$$

Если к алгебре Гейтинга $\langle L, \vee, \wedge, \Rightarrow, 0, 1\rangle$ добавить, например, закон исключенного третьего $x \vee \neg x = 1$ или закон двойного отрицания $\neg\neg x = x$, то получим аксиоматизацию булевой алгебры.

Теперь рассмотрим алгебраические примеры трехзначной логики Лукасевича $Ł_3$. В предыдущем разделе было показано, что логики со множествами связок $\{\rightarrow, \sim\}$ **и** $\{\vee, \wedge, \sim, M\}$ эквивалентны. Именно в этой сигнатуре Г.Мойсилом *[Moisil* 1940] было введено понятие трехэлементной алгебры Лукасевича, аксиоматизация которой была значительно упрощена А.Монтейро *[MonteiroA.* 1963]: Алгебра

$$\mathcal{L}_3 = \langle\{1, \tfrac{1}{2}, 0\}, \vee, \wedge, \sim, M, 1\rangle$$

есть трехэлементная алгебра Лукасевича, где $\langle\{1, \frac{1}{2}, 0\}, \vee, \wedge, 1\rangle$ есть дистрибутивная решетка с 1 и для унарных операторов ~ и M выполняются следующие тождества:

1. $\sim\sim x = x$,
2. $\sim(x \wedge y) = \sim x \vee \sim y$,
3. $\sim x \vee Mx = 1$,
4. $x \wedge \sim x = \sim x \wedge Mx$,
5. $M(x \wedge y) = Mx \wedge My$.

Или, по-другому, трехэлементная алгебра Лукасевича $\mathcal{L}_3$ есть алгебра де Моргана, снабженная операцией M, удовлетворяющей условиям (3), (4), (5). Или, по-другому, $\mathcal{L}_3$ есть алгебра Клини, снабженная операцией M, удовлетворяющей условиям (3), (4). Заметим, что существует большое число алгебраических построений для $Ł_3$, эквивалентных между собой. Чтобы все их систематизировать, обратим внимание на следующий факт: все сформулированные выше алгебры, начиная с алгебры де Моргана, на двухэлементном множестве {0, 1} превращаются в булеву двухэлементную алгебру.

С введением третьего элемента проблема становится не столь тривиальной, однако все указанные дважды алгебры, а также симметрическая алгебра Гейтинга и некоторые объединения их сигнатур являются трехэлементными алгебрами Лукасевича $\mathcal{L}_3$, поскольку трехзначные логики со множествами связок $\{\vee, \wedge, \neg, \lceil\}$, $\{\vee, \wedge, \Rightarrow, \Leftarrow\}$, $\{\vee, \wedge, \Rightarrow, \sim\}$, $\{\vee, \wedge, \Rightarrow, \lceil\}$, $\{\vee, \wedge, \Rightarrow, \lceil\}$, $\{\vee, \wedge, \Leftarrow, \neg\}$, $\{\vee, \wedge, \sim, \neg\}$ эквивалентны.

Поэтому неудивительно, что в *[Varlet* 1968, 1969] дана характеризация $\mathcal{L}_3$ в терминах дважды *p*-алгебр, а точнее, дважды алгебр Стоуна; в *[Monteiro L.* 1970] - в терминах симметрической алгебры Рейтинга; в *[Iturrioz* 1976] - в терминах дважды алгебры Рейтинга, и в *[Bechio* 1978] - в терминах алгебры Рейтинга с дуальным псевдодополнением и в терминах дуальной алгебры Рейтинга с псевдодополнением.

(По-другому, $\mathcal{L}_3$ есть трехэлементная решетка с 0 и 1, резидуальная относительно пересечения и объединения)
Заметим, что приведенную выше аксиоматизацию $\mathcal{L}_3$ можно рассматривать как аксиоматизацию в терминах алгебры де Моргана с псевдодополнением $\neg$, если заменить всюду оператор M на $\neg\neg$.
Имеются и другие аксиоматизации $\mathcal{L}_3$ *[Monteiro A.* 1980], например, характеризация $\mathcal{L}_3$ в терминах алгебры Нельсона. См также *[Abad & Figalla* 1984].
Интересно проследить изменение алгебраических структур с увеличением числа элементов. Например, известно, что если $n > 3$, то отрицание де Моргана нельзя определить посредством псевдодополнения и дуального псевдодополнения вместе с решеточными операциями. Поэтому характеризация $Ł_4$ (см. ниже) способом Дж. Варле *[Varlet* 1968, 1969] непригодна. Все дело в том, что функциональные свойства $Ł_3$ настолько «богаты» (точный смысл этого слова будет определен ниже) и обладают таким «критическим» свойством, что допускают столь много различных алгебраических характеризаций.

3. Конечнозначные логики Лукасевича $Ł_n$

Уже к 1930 г. были получены основные результаты относительно матричных логик Лукасевича $Ł_n$ и введено само понятие логической матрицы. Все объявленные результаты (без доказательств) оказались справедливыми и одним из наиболее интересных является сформулированная А.Тарским теорема о кардинальной полноте $Ł_n$ (число расширений). Здесь впервые с помощью компьютерной программы обсуждаются некоторые свойства этих расширений, а в Таблицах чисел (таблица 1) приводятся значения для $n \leq 2000$. Отметим также исключительно важный результат Р. МакНотона о критерии выразимости логических связок в бесконечнозначной логике Лукасевича и, как следствие, в $Ł_n$. Ни для каких других логик подобного свойства не обнаружено. Этот результат окажется существенным при выявлении истинной сущности логик $Ł_n$ (см. гл. 8).

3.1. Логические матрицы

Понятие многозначной логической матрицы для фиксированного пропозиционального языка *S* введено Я.Лукасевичем и А.Тарским в

ставшей классической *работе [Lukasiewicz & Tarski* 1930], подводящей итог исследованиям Львовско-Варшавской школы в области многозначной логики. См. также *[Wojcicki* 1988].
Логическая матрица представляет собой систему $\mathfrak{M} = < V, f_1, \ldots, f_k, D >$, где V есть непустое множество истинностных значений, элементы которого обозначаются x, y, z с индексами или без них; $f_1, \ldots, f_k$ - множество матричных операций, определенных на множестве V, и D - множество выделенных значений, такое, что $D \subset V$. После этого вводится функция оценки *v*, приписывающая пропозициональным переменным значения из множества V. Формула *A* называется общезначимой в $\mathfrak{M}$, если при любых значениях переменных в множестве V значение *A* принадлежит D. **Матричная логика есть не что иное, как множество общезначимых формул в данной матрице.**
Обратим внимание на то, что логическая матрица $\mathfrak{M}$ представляет собой систему $<\mathcal{A}, D>$, где $\mathcal{A}$ - некоторая универсальная алгебра, а множество матричных операций $f_1, \ldots, f_k$ образует ее сигнатуру. Часто именно в алгебраических терминах дается определение самой многозначной логики.
Пусть *S* есть пропозициональный язык произвольной логики, который, как обычно, состоит из множества пропозициональных переменных, множества логических связок $\{\supset, \vee, \wedge, \neg\}$ и вспомогательных символов, т.е. *S* есть множество всех формул этого языка. Тогда *S* можно рассматривать как **алгебру формул**. Под логической матрицей для *S* подразумевается любая матрица $\mathfrak{M} = <\mathcal{A}, D>$ с алгеброй $\mathcal{A}$ подобной алгебре *S,* т.е. операции обеих алгебр имеют одну и ту же *арность.* Это позволяет определить *оценку* языка *S* в $\mathfrak{M}$, как гомоморфизм h: $S \to \mathcal{A}$. Пусть $A \in S$.
Тогда формула *A* истинна, если $h(A) \in D$, и *A* является тавтологией, если $h(A) \in D$ для каждого гомоморфизма *h* языка *S* в $\mathcal{A}$.
Множество всех тавтологий обозначается посредством $E(\mathfrak{M})$.
Под *правилом* над множеством *S* обычно понимается отношение $r \subseteq \mathcal{P}(S) \times S$, где $\mathcal{P}(S)$ есть множество всех подмножеств *S* и × есть операция декартова произведения; при этом, естественно, правила должны сохранять тавтологичность. Для правила modus ponens (из *A* и $A \supset B$ следует *B)* это свойство содержится в следующем определении матрицы: логическая матрица называется *нормальной,* если формулы $A \in D$ и $B \in V$ всегда влекут $A \supset B \in V$, где D и V два непересекающихся множества *[Lukasiewicz & Tarsfci* 1930]. Таким образом, логическая матрица называется нормальной, если она верифицирует правило modus ponens.

Предположим, что $\mathcal{R}$ есть некоторое множество правил над S и пусть $X \subseteq S$. Каждая такая пара $(X, \mathcal{R})$ называется *пропозициональным исчислением* **L** над S. Говорят, что матрица $\mathfrak{M}$ *адекватна* для исчисления $(X, \mathcal{R})$, если замыкание X относительно всех правил из $\mathcal{R}$ равно $E(\mathfrak{M})$.

Особый интерес представляют исчисления $\mathbf{L} = (X, \mathcal{R})$, где $\mathcal{R}$ есть множество правил, которое содержит по крайней мере два правила: modus ponens и подстановку. Понятие вывода формулы A определяется стандартно, как, например, у Э.Мендельсона *[Мендельсон* 1976]:

Выводом в **L** называется всякая последовательность $A_1, \ldots, A_n$ формул, такая, что для любого i формула A_i, есть либо аксиома **L**, либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода. Формула A логики **L** называется теоремой в L, если существует вывод в L, в котором последней формулой является A; такой вывод называется *выводом формулы A*. Запись $\vdash A$, как и ранее, служит сокращением утверждения *«A* есть теорема». Если формула A выводима из некоторого множества Γ исходных формул, то тогда запись принимает вид $\Gamma \vdash A$.

Поскольку вывод во всех исчислениях определяется по приведенной выше схеме, то пара $\mathbf{L} = (X, \mathcal{R})$ полностью определяет множество доказуемых в **L** формул. Заметим, что исчисления рассматриваемого вида принято называть исчислениями *гильбертовского типа,* а множество X - множеством *аксиом* исчисления $\mathbf{L} = (X, \mathcal{R})$.

Предположим, что **L** абсолютно непротиворечиво. Тогда матрица $\mathfrak{M}$ называется *моделью* **L**, если каждая доказуемая формула в **L** общезначима в $\mathfrak{M}$. Если же верно и обратное, т. е. что каждая общезначимая формула в $\mathfrak{M}$ доказуема в **L**, то модель $\mathfrak{M}$ называется *точной моделью* или, по-другому, $\mathfrak{M}$ есть *характеристическая матрица* для **L**.

Поскольку понятие матрицы подпадает под более общее понятие алгебраической структуры (или модели), то все модельно-теоретические операции, которые используются на алгебраических структурах, применимы и к логическим матрицам. Некоторые понятия окажутся нам полезными. Так, $\mathfrak{N} = \langle \mathcal{A}^*, D^* \rangle$ является *подматрицей* $\mathfrak{M} = \langle \mathcal{A}, D \rangle$, если $\mathcal{A}^*$ есть подалгебра $\mathcal{A}$ (это значит, что операции из $\mathcal{A}$ замкнуты на некотором подмножестве $V^* \subset V$) и $D^* = V^* \cap D$. Имеет место следующий важный факт: если $\mathfrak{N} \subseteq \mathfrak{M}$, то $E(\mathfrak{M}) \subseteq E(\mathfrak{N})$.

Пусть **J** есть любое множество индексов. Для каждого $j \in J$ пусть $\mathfrak{M}_j = \langle \mathcal{A}, D \rangle$ есть определенная матрица для языка $\mathcal{L}$. Мы можем образовать произведение алгебры $\mathcal{A} = \prod_{j \in J} \mathcal{A}_j$ и ее подмножества $D = \prod_{j \in J} D_j$. В результате матрица $\mathfrak{M} = (\mathcal{A}, D)$ называется *произведением матриц* $\mathfrak{M}_j$ и обозначается посредством $\prod_{j \in J} \mathfrak{M}_j$. Имеет место следующая теорема *[Jaskowski* 1936]:

Если $\mathfrak{M} = \prod_{j \in J} \mathfrak{M}_j$, **тогда** $E(\mathfrak{M}) = \prod_{j \in J} E(\mathfrak{M}_j)$.

Отсюда следует, что операция произведения матриц сохраняет класс тавтологий исходной матрицы.

Заметим, что в общем случае операция прямого произведения алгебр не сохраняет свойств исходной алгебры. Поэтому вводится понятие *подпрямого произведения* (см. *[Burns & Sankappanavar* 1981]): алгебра $\mathcal{A}$ является подпрямым произведением индексированного семейства $(\mathcal{A}_j)_{j \in J}$ алгебр, если (i) $\mathcal{A} \leq \prod_{j \in J} \mathcal{A}_j$ и (ii) $\pi_j(\mathcal{A}) = \mathcal{A}_j$ для всех $j \in J$, где π_j есть *проективное отображение.*

3.2. N-значная матричная логика Лукасевича

Наиболее известными и, как мы увидим, обладающими удивительными свойствами являются конечнозначные логики Лукасевича $\mathbf{L}_n$, матричное определение которых впервые появилось в *[Lukasiewicz* 1922/1923]. Эти логики являются обобщением трехзначной логики $\mathbf{L}_3$ и имеют следующий пропозициональный язык $\mathcal{L}$.

Пусть *p, q, r* с индексами или без них суть пропозициональные переменные; ~, → суть логические связки, и (,) - вспомогательные символы. Определим понятие формулы.

1)p,q,r, ... - формулы;

2) если *A* и *B* - формулы, то *~A* и *A→ B* - формулы;

3) никакие другие конечные последовательности исходных символов, кроме тех, которые построены в силу пунктов (1)-(2), не являются формулами.

Другие логические связки вводятся как и для $\mathbf{L}_3$.

Матрица вида $\mathfrak{M}_n^L = \langle V_n, \sim, \rightarrow, \{1\} \rangle$ называется *n*-значной матрицей Лукасевича $(n \in N, n \geq 2)$, где $V_n = \{0, {}^1/_{n-1}, {}^2/_{n-1}, \ldots,$

$^{n-2}/_{n-1}$, 1}; ~ есть унарная и → бинарная операции отрицания и импликации соответственно, определенные на множестве V_n следующим образом:

$$\sim x = 1-x,$$

$$x \to y = min(1, 1-x+y).$$

Операции дизъюнкции и конъюнкции вводятся по определению:

$$x \vee y = (x \to y) \to y = max(x, y),$$

$$x \wedge y = \sim(\sim x \vee \sim y) = min(x, y).$$

Определим теперь функцию оценки (гомоморфизм) *v* формул языка $\mathcal{L}$ в матрице $\mathfrak{M}_n^L$. *v* есть функция оценки формул языка $\mathcal{L}$ в матрице $\mathfrak{M}_n^L$, если она удовлетворяет следующим условиям:

1) функция *v* определена для каждой формулы *A;*
2) если *A* есть пропозициональная переменная, то $v(A) \in V_n$;
3) если *A* и *B* есть формулы, то

$$v(\sim A) = \sim v(A),$$

$$v(A \to B) = v(A) \to v(B).$$

Обратим внимание, что здесь в левые части равенств входят пропозициональные связки, а в правые - операции из матрицы $\mathfrak{M}_n^L$. Будем говорить, что формула *A* является тавтологией в матрице $\mathfrak{M}_n^L$, если *v(A)* = 1 для любой функции оценки *v* в матрице $\mathfrak{M}_n^L$. Наконец, многозначная матричная логика Лукасевича $Ł_n$ есть множество тавтологий в $\mathfrak{M}_n^L$.

Отметим, что матрица $\mathfrak{M}_2^L$ является характеристической для классического пропозиционального исчисления C_2 (см. раздел 1.4), а матрица $\mathfrak{M}_3^L$ является характеристической для трехзначного исчисления $Ł_3$ (см. раздел 2.4).

3.3. Некоторые свойства $Ł_n$

Ниже мы увидим, что свойства конечнозначных логик Лукасевича $Ł_n$ определенным образом связаны с теорией чисел. Первый глубокий результат будет рассматриваться в разделе 5.3, а сейчас отметим некоторые важнейшие свойства $Ł_n$, в том числе представляющие интерес с точки зрения возможности данной связи.

3.3.1. Отношения между конечнозначными логиками Лукасевича

Основные отношения между конечнозначными логиками Лукасевича описываются следующим условием Линденбаума *[Lukasiewicz & Tarski* 1930, Теорема 19]:

$Ł_n \subseteq Ł_m$ *т.т.т., когда* $m-1$ *есть делитель* $n-1$.

Из этой теоремы имеем очевидное следствие для случая, когда *n-1* есть простое число и *(n-1)* > 1:

$$Ł_\infty \subset Ł_{kn} \subset \ldots \subset Ł_{2n} \subset Ł_n \subset Ł_2.$$

Однако только свойство полноты *(степень полноты)* конечнозначных логик $Ł_n$ выявит роль простых чисел на уровне определения $Ł_n$ как матричных логик, т.е. на уровне классов тавтологий.

3.3.2. Степень полноты для $Ł_n$ (появление простых чисел)

У Тарского *[Tarsia* 1930] дается определение степени (дедуктивной) полноты произвольной логики **L**. Мы будем пользоваться этим определением в следующей форме.

Определение 3.1. *Ординальной степенью полноты* множества аксиом логики **L**, символически $\vec{\gamma}$ **(L),** является наименьшее ординальное число $\alpha \neq 0$, такое, что не существует возрастающей последовательности типа α абсолютно непротиворечивых неэквивалентных систем аксиом, которые начинаются с **L**.

Линденбаум доказал, что ординальная степень полноты $Ł_3$ есть 3 (см. *[Lukasiewicz & Tarski* 1930]. Тарский затем обобщил эту теорему для всех *n,* таких, что *n-1* есть простое число. Таким образом, если *n*-1 есть простое число, то добавление любой формулы, которая не есть теорема $Ł_n$, но есть теорема $Ł_2$, к аксиомам $Ł_n$, дает $Ł_2$; и добавление любой формулы, которая не есть теорема $Ł_2$, к аксиомам $Ł_n$ дает противоречивость. Затем в 1930 г. «проблема степени полноты была решена для систем $Ł_n$ с произвольным натуральным *n;* это был совместный результат семинара Лукасевича и Тарского в Варшавском университете. Доказательство не было опубликовано и было найдено вновь А.Роузом *[Rose* 1951, 1952, 1969]:

Теорема 3.1. *Для любого* $n \geq 2$ *ординальная степень полноты* $Ł_n$ *есть* $d(n-1) + 1$, *где d(x) есть число всех делителей x, включая x и* 1.

Тарский *[Tarski* 1930] вводит также понятие кардинальной степени полноты логики; оно может быть переформулировано следующим образом.

Определение 3.2. *Кардинальной степенью полноты* логики **L,** символически $\gamma(\mathbf{L})$, является число логик, содержащих аксиомы логики **L**.

Для $Ł_3$ и для *n*-1 - простое число, ситуация аналогичная, т.е.

$$\gamma(\mathbf{L}) = \vec{\gamma}(\mathbf{L}) = 3 \text{ для } n\text{-}1 - \text{простое число.}$$

Легко видеть, что $\vec{\gamma}(\mathbf{L}) \leq \gamma(\mathbf{L})$. В общем случае теорема для $\gamma(\mathbf{L})$, т е. для произвольного *n,* была впервые опубликована Токажем *[Tokarz* 1974] и упрощена в *[Tokarz* 1977].

Пусть $C = \langle a_1, \ldots, a_n \rangle$ - произвольная последовательность натуральных чисел. Через $N_c(a_i)$, $1 \leq i \leq n$, будем обозначать число всех подпоследовательностей *D* из С, которые удовлетворяют следующим условиям:

(1) $a_i \in D$ и для любого $b \in D$, $a_t \geq b$,

(2) если $j \neq k$ и $a_j, a_k \in D$, то a_j-1 не является делителем a_k-1.

Для любого n $c(n) = \langle a_1, \ldots, a_k \rangle$ будет последовательностью, определяемой следующими условиями.

(i) $a_1 = n$,

(ii) $a_1 > \quad > a_k > 1$,

(iii) для любого i $1 \leq i \leq k$, a_i-1 есть делитель n-1.

Теорема 3.2. *Для конечного n, кардинальная степень полноты* $Ł_n$ *есть*

$$\left(\sum_{a_i \in c(n)} N_{c(n)}(a_i) \right) + 1.$$

На самом деле вычисление кардинальной степени полноты для произвольной $Ł_n$ возможно *только* с помощью специальной компьютерной программы, которая и была создана в 2000 г. М.Н.Рыбаковым (кафедра общей и прикладной алгебры и геометрии Тверского государственного университета). Программа несколько раз значительно усовершенствовалась, что позволило разработчику программы вычислить степень кардинальной полноты для первых 12000$Ł_n$. Значения для $n < 2000$ приведены в Таблице 1.

Рассмотрение значений кардинальной полноты логик Лукасевича $Ł_n$ представляет собой любопытную картину.

Некоторые числа являются весьма «популярными», а некоторых вообще нет и, видимо, не предвидится. Так, в десяти тысячах расширений $Ł_n$ из первых 100 натуральных чисел появляются следующие числа:

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 20, 21, 28, 35, 36, 45, 50, 55, 56, 66, 70, 78, 84, 91.

Как видно из таблицы 1, она содержит 35 различных чисел в первой тысяче расширений $Ł_n$, во второй тысяче расширений появляется всего 11 новых чисел; в третьей тысяче - 6; в четвертой тысяче - 5; в пятой тысяче - 5; в шестой тысяче - 4; в седьмой тысяче - 5 (именно здесь появляется число 91); в восьмой тысяче -4; в девятой тысяче - 1; в десятой тысяче - 2.

Из первых десяти тысяч логик Лукасевича $Ł_n$ рекордсменом является $Ł_{9241}$: её степень кардинальной полноты 2068224. Интересно, что появляющиеся «большие» числа, затем неоднократно повторяются. Только что указанное число является также степенью кардинальной полноты $Ł_{10921}$.

В связи со степенью кардинальной полноты $Ł_n$ см. ниже раздел 6.3.1 «Гипотеза о конечности корневых деревьев».

3.3.3. J_i-операторы

Особое место при изучении свойств конечнозначных логик Лукасевича занимают J_i-операторы (функции), введенные Дж.Россером и А.Тюркеттом *[Rosser & Turqette* 1952]:

$$j_i(x) = \begin{cases} 1, \text{если } x = 1 \\ 0, \text{если } x \neq 1. \end{cases}$$

Ими доказана следующая

Теорема. *J_i-функции определимы посредством* $\rightarrow$ *и* $\sim$ *в* $Ł_n$.

Это доказательство упрощено во многих работах.

ДОКАЗАТЕЛЬСТВО.

Пусть $H_1(x) = \sim x$ и $H_{k+1}(x) = x \rightarrow H_k(x)$. Рассмотрим I_i -функции:

$$I_i(x) = \begin{cases} 1, \text{если } x \leq 1 \\ 0, \text{если } x > 1 \end{cases}$$

I_i -функции определяются индукцией по *i*. Заметим, что $I_0(x) = H_k(x)$. Предположим, I_q определено для $q \leq i$ и пусть *r* есть наибольшее целое число, такое, что $H_r(i+1) > 0$. Определим $p = H_r(i+1)$-1. Тогда $p \leq i$ и мы можем определить:

$I_{i+1}(x) = \sim I_p(H_t(x))$.

Следовательно,

$J_0(x) = \sim I_0(x)$,

$J_i(x) = (\sim I_i(x)) \wedge (\sim I_{i-1}(x))$.

Заметим, что результат этой теоремы следует из критерия Р.Мак-Нотона (о нем см. ниже 3.3.6).

На самом деле J_i-операторы являются характеристическими функциями числа i, $i=0, {}^{1}/_{n-1}, {}^{2}/_{n-1}, \ldots, {}^{n-2}/_{n-1}, 1$, и обобщают некоторые свойства отрицания. В дальнейшем ходе исследования J_i-операторы будут не раз использоваться.

3.3.4. Ł$_n$ и *n*-значные логики Гёделя G$_n$

К. Гёдель *[Godel* 1932] показал, что никакая конечно-значная матрица не может быть характеристической для пропозиционального интуиционистского исчисления **H**. В связи с этим им была построена следующая логическая матрица

$\mathfrak{M}_n^G = < V, \neg, \vee, \wedge, \Rightarrow, \{1\} >$,

где

$$\neg x = \begin{cases} 1, если\ x = 0 \\ 0, если\ x \neq 0, \end{cases}$$

$x \vee y = max(x,y)$,

$x \wedge y = min(x,y)$

$$x \Rightarrow y = \begin{cases} 1, если\ x \leq y \\ y, если\ x > y, \end{cases}$$

$x \Leftrightarrow y = (x \Rightarrow y) \wedge (y \Rightarrow x)$.

Трехзначная логика Гёделя есть в точности трехзначная логика Гейтинга **G**$_3$. Таким образом, конечнозначная логика **G**$_n$ есть обобщение **G**$_3$.

Логика **G**$_n$ аксиоматизирована различными способами *[Thomas* 1962], *[Hosoi* 1966], *[Хомич* 1986]. Причем, в первой и последней работе аксиоматизация производится за счет добавления к интуиционистскому пропозициональному исчислению **H** каждый раз только одной импликативной аксиомы.

Нетрудно показать, что операции из **G**$_n$ выразимы посредством операций из Ł$_n$, т. е. **G**$_n$ функционально вложима в Ł$_n$. Для этого надо определить $\neg x$ и $x \Rightarrow y$. Это также следует из критерия Мак-Нотона, однако стоит показать, как это выглядит на самом деле. Заметим, что $\neg x$ есть не что иное, как оператор Россера-Тюркетта $J_0(x)$. В свою очередь Р. Чиньоли *[Cignoli* 1982] показал, что

$x \Rightarrow y = J_1(x \rightarrow y) \vee y$.

В итоге операции из **G**$_n$ определяются в Ł$_n$. Это позволяет строить аксиоматизацию Ł$_n$ на основе интуиционистской импликации, что и было впервые сделано Р.Чиньоли *[Cignoli* 1982] (см. раздел 3.5.)

3.3.5. Функтор Слупецкого для Ł$_n$

В *[Rosser & Turqette* 1952] дано также обобщение результата Е. Слупецкого относительно $Ł_3^T$ (см. главу 2):

Пусть $T_{\frac{n-2}{n-1}}(x) = \frac{n-2}{n-1}$ *для всех* $x \in V$. *Тогда система функций*

$\{x \rightarrow y, \sim x, T_{\frac{n-2}{n-1}}(x)\}$ *функционально полна.*

В свою очередь обобщением этого результата стала теорема Эванса-Шварца *[Evans & Schwartz* 1958]:

Пусть $T_i(x) = i$, *где* $0 < i < 1$. *Тогда система функций* $\{x \rightarrow y, \sim x, T_i(x)\}$ *является функционально полной т.т.т., когда* $(n-1, i) = 1$, *т.е. n-1 и i есть взаимно простые числа.*

Этот результат независимо был открыт Р.Клэем *[Clay* 1962] как следствие теоремы:

Система функций $\{x \rightarrow y, \sim x, T_i(x)\}$ *является функционально полной т.т.т., когда* $(n-1, i_1, \ldots, i_k) = 1$, *где* $0 < i_k < 1$ *и* $0 < k < n$.

Ниже, в пятой главе, будет дано строгое определение понятия *функциональной полноты.*

3.3.6. Критерий Мак-Нотона об определимости операций в Ł$_n$

В общем случае на вопрос о том, какие операции (функции) можно опеределить в Ł$_n$, дает ответ критерий определимости Р.Мак-Нотона *[McNaughton* 1951], который является следствием фундаментальной теоремы Мак-Нотона об определимости операций (функций) в континуальной логике Лукасевича Ł$_\infty$ (см. Приложение; раздел 1).

Функция $f(\frac{x_1}{n-1},\ldots,\frac{x_k}{n-1}) = \frac{x}{n-1}$ *определима в матрице для* $Ł_n$ *тогда и только тогда, когда НОД*$(x_1, \ldots, x_k, n-1)$ *есть делитель x (НОД -наибольший общий делитель).*

С помощью критерия Мак-Нотона доказывается много важных теорем относительно *n*-значных логик Лукасевича, в том числе теоремы 3.1 и 3.2. в предыдущем разделе. См. об этом *[Токаж* 1979].

3.4. Аксиоматизация $Ł_n$

В разделе 2.5 была рассмотрена аксиоматизация трехзначной логики Лукасевича $Ł_3$, предложенная М.Вайсбергом. Однако неясно, как этот способ аксиоматизации может быть распространен на конечнозначные логики $Ł_n$. Правда, ему же принадлежит аксиоматизация $Ł_n$ для случая, когда *n-1* есть *простое число*. Как отмечается в *[Lukasiewicz & Tarski* 1930], расширение этого результата на произвольное конечное *n* принадлежит Линденбауму. Позже М.Вайсбергом *[Wajsberg* 1935] был предложен общий метод аксиоматизации широкого класса конечнозначных логик, куда входят также все *n*-значные логики Лукасевича. Однако метод, предложенный Вайсбергом, весьма громоздок и практически мало пригоден.

Две неудачные попытки аксиоматизировать $Ł_n$ были предприняты Дж.Россером и А.Тюркеттом *[Rosser & Turquette* 1945, 1950].

(Из последней работы следует, что в $Ł_n$ выводим закон сокращения $(p \to (p \to q)) \to (p \to q)$, что не верно ни для какого $n \geq 3$.)

Наконец ими был разработан метод аксиоматизации, *[Rosser & Turquette* 1952], который включает в себя в качестве исходного условия общезначимость законов *транзитивности, перестановки* и *утверждения консеквента* (см. выше гл. 1). Кроме этого здесь впервые было указано на обязательное наличие в аксиоматизируемой логике J_i-операторов. Все эти условия выполняют, например, конечнозначные логики Лукасевича $Ł_n$ (Теорема 3.5.). Этот метод имеет место также для произвольного числа выделенных значений и распространяется на предикатные многозначные логики. Однако, как и предыдущий метод, он оказался весьма общим и громоздким в применении.

Только в начале 70-х годов появились сразу две аксиоматизации $Ł_n$. После того как было дано алгебраическое доказательство полноты для бесконечнозначной логики Лукасевича $Ł_\infty$ (см. Приложение), появилась возможность распространить этот метод на *n*-значный случай, что и было сделано Р.Григолия *[Григолия* 1973], *[Grigolia* 1977]. Аксиоматизация $Ł_n$, предложенная Григолия, основана на том,

что к четырем аксиомам для бесконечнозначной логики Лукасевича $Ł_\omega$ добавляются последовательно характеристические аксиомы для каждого *n*. Выглядит это следующим образом:

1. $p \to (q \to p)$
2. $(p \to q) \to ((q \to r) \to (p \to r))$
3. $((p \to q) \to q) \to ((q \to p) \to p)$
4. $(\sim p \to \sim q) \to (q \to p)$
5. $np \to (n-1)p$.

Если $n>3$, то добавляется следующая аксиома:

6. $(n-1)\,((\sim p)^j \vee (p \wedge (j-1)p))$,

где $1 < j < n-1$ и j не делит n-1; np и p^j есть сокращения для $p \vee p \vee \ldots \vee p$ (n раз) и $p \wedge p \wedge \ldots \wedge p$ (j раз) соответственно.

Правила вывода: modus ponens и подстановка.

Другой метод аксиоматизации предложен М.Токажем *[Tokarz* 1974], который для этого существенно использовал критерий Р.Мак-Нотона для выразимости операций в $Ł_\infty$. Однако оба метода (особенно последний) требуют добавления формул слишком большой длины. Поэтому интерес представляет работа Р.Тузьяка *[Tuziak* 1988], где аксиоматизация для $Ł_n$ проще, чем во всех предыдущих работах (правда, в другой сигнатуре, чем исходная у Лукасевича), и, главное, не опирается на такие сильные метатеоремы, как алгебраическое доказательство полноты для $Ł_\infty$ или критерий Мак-Нотона для $Ł_\infty$, хотя для доказательства полноты и используются средства алгебры Линденбаума.

Новая аксиоматизация выглядит следующим образом для любого

$$p \to^{k+1} q = p \to (p \to^{k} q) \text{ и } p \equiv q = (p \to q) \wedge (q \to p).$$

$n \geq 2$. Используются следующие сокращения: $p \to^{0} q = q$,

1. $(p \to q) \to ((q \to r) \to (p \to r))$.
2. $p \to (q \to p)$.
3. $((p \to q) \to q) \to ((q \to p) \to p)$.
4. $(p \to^{n} q) \to (p \to^{n-1} q)$.
5. $p \wedge q \to p$.
6. $p \wedge q \to q$.
7. $(p \to q) \to ((p \to r) \to (p \to q \wedge r))$.
8. $p \to p \vee q$.
9. $q \to p \vee q$.
10. $(p \to r) \to ((q \to r) \to (p \vee q \to r))$.
11. $(\sim p \to \sim q) \to (q \to p)$.
12. $(p \equiv (p \to^{s-2} \sim p)) \to^{n-1} p$ для любого $2 \le s \le n\text{-}1$, такого, что s не есть делитель n-1.

Правила вывода: modus ponens и подстановка.

Обратим внимание, что при $n = 2$ и $n = 3$ аксиома (12) отсутствует. При $n = 2$ мы имеем аксиоматизацию классической пропозициональной логики. Тогда аксиома (4) есть

$$(p \to (p \to q)) \to (p \to q).$$

При $n = 3$ аксиома (4) есть

$$(p \to (p \to (p \to q))) \to (p \to (p \to q)).$$

Если $n = 4$, тогда аксиома (12) приобретает вид

$$(p \equiv \sim p) \to ((p \equiv \sim p) \to ((p \equiv \sim p) \to p)).$$

При этом достаточно рассматривать только простые числа s в аксиоме (12).

Аксиоматизация предикатной логики $Ł_n$ представлена в *[Urquhart* 1986].

Имеются различные исчисления с устранением сечения (или семантические таблицы) для $Ł_n$ (см. *[Rousseau* 1967], *[Takahashi* 1967], *[Carnielli* 1987], *[Hahnle* 1993], *[Baaz, Fermuller, Zach* 1994], *[Gil, Torrens, Verdu* 1997], *[Aguzzoli, Ciabattoni, Di Nola* 1999]). Обратим внимание, что в *[Prijatelj* 1996] предложено генценовское исчисление логик $Ł_n$, в основе которого лежит ограничение структурного правила *сокращения.* Заметим, что одна из версий ограничения закона сокращения появляется уже в аксиоматизации Григолия, а у Тузьяка

аксиома (4) есть в точности ограничение закона сокращения в гильбертовской форме.

3.5. Алгебраизация $Ł_n$

Первые работы в области алгебраизации $Ł_n$ принадлежат Г Мойсилу, который задался целью построить алгебраический аппарат для n-значных логик Лукасевича, играющий ту же роль, что и булевы алгебры для классической логики. В *[Moisil* 1940] были построены алгебры для $Ł_3$ и $Ł_4$, а в *[Moisil* 1941] (см также *[Moisil* 1963]) эти алгебры были обобщены на n-значный случай. Полученные алгебры были названы *n-значными алгебрами Лукасевича;* они представляют собой алгебру де Моргана (см. выше раздел 1.5), снабженную множеством операторов δ_i^n, которые определяются на множестве V_n следующим образом:

$$\delta_i^n x = \delta_i^n(x/_{n-1}) = \begin{cases} 1, \text{если } i + j \ge n \\ 0, \text{если } i + j < n. \end{cases}$$

Заметим, что в *[Suchon* 1974] дается определение операторов σ_i^n в матрице Лукасевича $\mathfrak{M}_n^L$.

Приведем аксиоматизацию класса всех n-значных алгебр Лукасевича, принадлежащую Л.Итурриоз *[Iturrioz* 1977]. В этой работе введено понятие *симметрической алгебры Рейтинга порядка n:*

$\langle L, \vee, \wedge, \Rightarrow, \sim, \sigma_1^n, \ldots, \sigma_{n-1}^n, 0, 1\rangle$ есть *n-значная алгебра Лукасевича* $(n \ge 2)$, если $\langle L, \vee, \wedge, \Rightarrow, \sim, 0, 1\rangle$ есть симметрическая алгебра Рейтинга (см раздел 2.5) и σ_i^n, $1 \le i \le n\text{-}1$ суть унарные операторы, которые удовлетворяют следующим условиям:

(L1) $\sigma_i^n(x \vee y) = \sigma_i^n x \vee \sigma_i^n y$,

(L2) $\sigma_i^n(x \Rightarrow y) = \bigwedge_{j=i}^{n-1} (\sigma_j^n x \Rightarrow \sigma_j^n y)$,

(L3) $\sigma_i^n \sigma_j^n x = \sigma_j^n x$, $1 \le i, j \le n\text{-}1$,

(L4) $\sigma_1^n x \vee x = x$,

(L5) $\sigma_i^n \sim x = \sim \sigma_{n-i}^n x$,

(L6) $\sigma_1^n x \vee \sim \sigma_1^n x = 1$,

где $\bigwedge_{j=i}^{n-1} x_j$ стоит вместо $x_i \wedge x_{i+1} \wedge \ldots \wedge x_{n-1}$.

Эквациональный класс n-значных алгебр Лукасевича обозначим посредством $\mathcal{L}_n$. Приведем пример $\mathcal{L}_n$:

$\mathcal{L}_n = \langle V_n, \vee, \wedge, \sim, \Rightarrow, 0, 1\rangle$, где

$x \vee y = max(x, y)$,

$x \wedge y = min(x, y)$,

$\sim x = 1-x$,

$x \Rightarrow y$ есть импликация Геделя (см. выше 3.3.4), $\sigma_i^n x$ определены выше. Поскольку

$$x \Rightarrow y = y \vee \sim\sigma_{n-1}x \vee \bigvee_{i=1}^{n-1}(\sigma_{n-1}(x \wedge y) \wedge \sim\sigma_{n-i-1}\, x)$$ [8]

то характеризации n-значных алгебр Лукасевича, данные Г.Мойсилом и Л.Итурриоз, эквивалентны.

Свойства эквационального класса $\mathcal{L}_n$ исследовались в *[Cignoli* 1970], *[Balbes & Dwinger* 1974], *[Iturrioz* 1977, 1983] и других работах

Очевидно, что алгебра $\mathcal{L}_n$ есть дважды алгебра Гейтинга, поскольку

$x \Leftarrow y = \sim(\sim y \Rightarrow \sim x)$.

Еще один факт: алгебра $\mathcal{L}_n$ есть алгебра Клини *[Sicoe* 1967].

Обратим внимание, что J_i-операторы Россера-Тюркетта выразимы в n-значной алгебре Лукасевича. Установим, что $\sigma_n^n x = 1$ и $\sigma_0^n = 0$. Тогда

$J_i^n(x) = \sigma_{n-i}(x) \wedge \sim\sigma_{n-i-1}(x)$.

Более того,

$$\sigma_i^n(x) = \bigvee_{j=1}^{i} J_{n-j}^n(x),\ 1 \le i \le n\ .$$

Таким образом, в n-значной алгебре Лукасевича операторы σ_i^n можно заменить на операторы J_i^n.

Теперь сделаем важное замечание. А. Роуз обнаружил, что для случая $n \ge 5$ n-значная алгебра Лукасевича соответствует не n-значным логикам Лукасевича $\mathbf{Ł_n}$, а их фрагментам. Это значит, что посредством операций $\vee$, $\wedge$, $\sim$ и J_i (или σ_i^n) нельзя выразить импликацию Лукасевича $x \to y$ для случая $n \ge 5$. Это же самое открытие сделал А. Тюркетт *[Turqette* 1969], правда, совершенно по другому поводу, а именно при обобщении трехзначной логики Бочвара-Клини на n-значный случай.

Отсюда возникает проблема построения адекватной алгебры для

n-значной логики Лукасевича $\mathbf{Ł_n}$, тогда построенные алгебры естественно называть *алгебрами Лукасевича-Мойсила.*

Подходящие алгебры для $\mathbf{Ł_n}$ были построены Р.Григолия *[Григолия* 1973] как ограничение MV-алгебр, введенных Ч.Ч.Ченом *[Chang* 1958] и являющихся алгебраической семантикой для бесконечнозначной логики Лукасевича $\mathbf{Ł_\infty}$ (см. Приложение), на конечный случай. В этих же работах Р.Григолия доказал, что произвольная конечная $\mathcal{L}_n$-алгебра Лукасевича (не Лукасевича-Мойсила) представима в виде прямого произведения $\mathcal{L}_m$-алгебр, где $m \le n$ и m-1 делит n-1 ($\mathcal{L}_m$-алгебра есть матрица $\mathfrak{M}_m^L$).

Однако эти алгебры не основаны непосредственно на решеточной структуре.

В *[Cignoli & de Gallego* 1981] строится пятиэлементная алгебра для $\mathbf{Ł_5}$, в основе которой лежит алгебра де Моргана с дополнительными условиями для новых унарных операторов , а в *[Cignoli* 1980] введена *собственно n-значная алгебра Лукасевича.*

Пусть $S_n = \{(i, j) \in N \times N: 3 \le i \le n\text{-}2,\ 1 \le j \le n\text{-}4,\ j < i\}$, если $n > 5$, и $S_n = \varnothing$, если

$n < 5; T_n = \{(i, j) \in N \times N: 2 \le i \le n\text{-}2,\ 1 \le j \le n\text{-}3,\ j < i\}$,

если $n \ge 4$, и $T_n = \varnothing$, если $n < 4$.

Собственно n-значная алгебра Лукасевича есть система

$\langle L, \vee, \wedge, \Rightarrow, \sim, \{\sigma_i^n\}_{1 \le i \le n-1}, \{F_{ij}^n\}_{(i,j) \in S_n}, 0, 1\rangle$

такая, что

$\langle L, \vee, \wedge, \Rightarrow, \sim, \{\sigma_i^n\}_{1 \le i \le n-1}, 0, 1\rangle$

есть n-значная алгебра Лукасевича-Мойсила $\mathcal{L}_n$, и F_{ij}^n есть бинарные операции, связанные с $\mathcal{L}_n$ следующими тождествами:

$$\delta_k^n(F_{ij}^n(x, y)) = \begin{cases} 0, \text{если } k \le i - j \\ J_i^n(x) \wedge J_j^n(y), \text{если } k > i - j \end{cases} \quad 1 \le k \le n-1.$$

Эквациональный класс всех собственно n-значных алгебр Лукасевича обозначим посредством S_n. Очевидно, что для $2 \le n \le 4$, $S_n = \mathcal{L}_n$, поскольку в этом случае $S_n = \varnothing$.

Примером S_n является алгебра $\mathcal{L}_n$, если к ней добавим следующие операции:

$$F_{ij}^n(x, y) = F_{ij}^n\left(\tfrac{r}{n-1}, \tfrac{s}{n-1}\right) = \begin{cases} \frac{n-1-i+j}{n-1}, \text{если} (r, s) = (i, j) \\ 0, \text{в остальных случаях} \end{cases} \quad (i, j) \in S_n.$$

Нетрудно вычислить, что число таких операций есть

$(n(n-5)+2)/2$, поскольку такова мощность множества S_n для $n \geq 5$.
Остается показать, что n-значная импликация Лукасевича $x \to y$ выразима в сигнатуре алгебры S_n:

$$x \to y = (x \Rightarrow y) \vee \sim x \vee \bigvee_{(i,j) \in T_n} F_{ij}^n(x, y).$$

Последняя формула опять же говорит о нетривиальности импликации Лукасевича →.
То, что операции $F_{ij}^n(x, y)$ выразимы посредством исходных операций n-значной логики Лукасевича $\sim x$ и $x \to y$, следует из критерия Р.Мак-Нотона, тем не менее:
$F_{ij}^n(x, y) = (x \to y) \wedge J_i^n(x) \wedge J_j^n(y)$.
В этой же работе Р.Чиньоли дает аксиоматизацию пропозициональной логики $Ł_n$ в сигнатуре алгебры S_n, т.е. на основе интуиционистской импликации $\Rightarrow$.

4. ИНТЕРПРЕТАЦИИ $Ł_n$

С *появлением и развитем многозначных логик вопрос об их интерпретации стновился все более актуальным. Не случайно два последних обзора по многозначной логике [Rose 1981] и [Urquhart 1986] заканчиваются именно этой темой; а проблема интерпретации истинностных значений является центральной и, видимо, сложнейшей проблемой для теории многозначных логик. Суть последней была сформулирована З.Йорданом: «Без интерпретации приписывания определенного логического значения числу n "истинностных значений", любое n-значное исчисление остается абстрактной структурой» [Jordan 1945]. Может показаться удивительным, что несмотря на то исключительное развитие, которое получили многозначные логики Лукасевича, вопрос об интерпретации истинностных значений в этих логиках все еще обсуждается в современной литературе. Вот что пишет по этому поводу Данна Скотт. «Перед тем как вы примете многозначную логику как долгожданного брата, попробуйте понять, что могут означать дробные истинностные значения. И имеют ли они какой-либо смысл? Каково концептуальное подтверждение "промежуточных значений"'» [Scott 1976]. Остается также неясным, отмечает Д. Скотт, обоснование логических операций в $Ł_n$. Даже сейчас можно утверждать, что логические свойства импликации Лукасевича остаются совершенно загадочными и только выход за сферу чисто логического позволяет выявить некоторые свойства последней.*

Несколько неожиданным оказалось то, что в 70-е годы было предпринято сразу несколько попыток описания многозначных логик Лукасевича в терминах бивалентных оценок (см. ниже). Закончилось это описание формальной интерпретацией $Ł_n$ строго в терминах классических истинностных значений T («истина») и F («ложь»), из которых образуются T-F-последовательности длиной s (булевы вектора), а затем на булевой алгебре мощностью 2^s строятся классы эквивалентности, которые и выступают в качестве истинностных значений для $Ł_n$. Так мы приходим к интерпретации $Ł_n$, названной нами «фактор-семантикой». Поэтому настоящую главу можно было бы назвать так: «Конечнозначные логики Лукасевича с точки зрения классической».

4.1. Тезис Сушко

В 1975 г. Р.Сушко озадачил сторонников и адептов многозначных логик построением бивалентной семантики для трехзначной логики Лукасевича $Ł_3$ *[Suszko* 1975] и внес сумятицу в умы многозначников, продолжающуюся по сей день, выдвижением тезиса о том, что каждая пропозициональная логика является *двузначной [Suszko* 1977].
Пусть *For* обозначает множество формул пропозиционального языка $\mathcal{L}$, а {0, 1} - множество истинностных значений. Тогда LV_3 есть множество всех функций $t : For \to \{0, 1\}$ таких, что для любых $\alpha, \beta \in For$ справедливы следующие условия:

(0) $t(\alpha) = 0$ или $t(\sim\alpha) = 0$
(1) $t(\alpha \to \beta) = 1$ всегда, когда $t(\beta) = 1$
(2) если $t(\alpha) = 1$ и $t(\beta) = 0$, то $t(\alpha \to \beta) = 0$
(3) если $t(\alpha) = t(\beta)$ и $t(\sim\alpha) = t(\sim\beta)$, то $t(\alpha \to \beta) = 1$
(4) если $t(\alpha) = t(\beta) = 0$ и $t(\sim\alpha) \neq t(\sim\beta)$, то $t(\alpha \to \beta) = t(\sim\alpha)$
(5) если $t(\sim\alpha) = 0$, то $t(\sim\sim\alpha) = t(\alpha)$
(6) если $t(\alpha) = 1$ и $t(\beta) = 0$, то $t(\sim(\alpha \to \beta)) = t(\sim\beta)$
(7) если $t(\alpha) = t(\sim\alpha) = t(\beta)$ и $t(\sim\beta) = 1$, то $t(\sim(\alpha \to \beta)) = 0$.

В итоге мы получили адекватную семантику для трехзначной логики Лукасевича $Ł_3$ *[Suszko* 1975].
При таком подходе элементы трехзначной матрицы Лукасевича 1, 1/2 и 0 не рассматриваются как *логические* значения; они предстают, по Сушко, как *алгебраические* значения. Введенная нами в разделе 3.2 функция оценки *v* является как раз алгебраической оценкой: гомоморфизмами, отображающими алгебру формул

в алгебру того же типа истинностных значений (в логическую матрицу). Тогда как, по Сушко, логической оценкой являются бивалентные оценки, рассмотренные как характеристические функции множества формул.

Обычно степень сложности описания многозначной логики увеличивается вместе с числом истинностных значений. Но в некоторых случаях сложность может быть упрощена применением дополнительных связок, «идентифицирующих» исходные (матричные) значения. Так использование J-операторов Россера и Тюркетта (см. раздел 3.3.3) дает возможность получить единообразное описание логических оценок для конечнозначных логик Лукасевича $Ł_n$ *[Malinowski* 1977].

Тезис Сушко вызвал определенную критику. Например, Г.Малиновский *[Malinowski* 1994] сконструировал трехзначную *квази-матричную* логику, для которой метод Сушко не может быть применен. Обсуждению дилеммы двузначности и многозначности посвящена значительная часть работы *[Beziau* 1997].

На самом деле, как мы увидим в следующей главе, есть *принципиальная* разница между классической двузначной логикой и многозначной (в том числе трехзначной логикой Лукасевича; см. раздел 5.2.4), выражающаяся в различных мощностях классов замкнутых функций.

4.2. Метод Скотта

Д.Скотт *[Scott* 1973, 1974], заменяя истинностные значения оценками, пытается придать более очевидную характеристику конечным многозначным конструкциям. Оценки являются двухзначными (bivalent) функциями и задают распределение множества высказываний данного языка по типам, соответствующим исходным логическим значениям. Две вышеупомянутые статьи содержат только набросок общего метода и его реализацию для n-значных логик Лукасевича.

Пусть *For* - множество формул данного пропозиционального языка $\mathcal{L}$ и $V=\{v_0, v_1, \ldots, v_{n-1}\}$ $(n \geq 1)$- конечное множество оценок: элементы множества V являются (в этом случае) произвольными функциями $v_i : For \to \{t, f\}$ с t, обозначающим истину, и f- ложь.

Под *типом* высказываний языка $\mathcal{L}$ относительно V мы понимаем произвольное множество Z_β вида

$$Z_\beta = \{\alpha \in For : v_i(\alpha) = v_i(\beta), \text{ для произвольного } i \in \{0, 1, \ldots, n-1\}\}.$$

Можно легко увидеть, что, используя *n*-элементное множество оценок, можно ввести максимально 2^n типов: так, например, двухэлементное множество оценок $\{w_0, w_1\}$ (см. таблицу) определяет четыре типа: Z_1, Z_2, Z_3, Z_4. Накладывая ограничения на оценки, мы сокращаем число типов. Только что рассмотренное множество оценок будет определять три (самое большее) типа: Z_1, Z_2, Z_4, когда мы потребуем, чтобы $w_0(\alpha) \leq w_1(\alpha)$ для любого $\alpha \in For$, два типа: Z_2, Z_3, если $w_0(\alpha) \neq w_1(\alpha)$ для каждого $\alpha \in For$, и Z_1, Z_4, при условии, что $w_0 = w_1$.

	w_0	w_1
Z_1	f	F
Z_2	f	T
Z_3	t	F
Z_4	t	T

Такие типы являются аналогами логических значений. В *[Scott* 1973] о них говорится как об «индексах». Рассмотренный выше пример показывает, что данная значность (valency) меньше 2^n может быть получена несколькими способами. Какие из этих редукций должны быть приняты в расчет, зависит от свойств пропозициональных связок, которые, со своей стороны, являются операциями над типами, т.е. отображениями последовательностей типов в множество типов. Правильный выбор ограничивающих условий ведет к относительно простому описанию рассматриваемых связок.

Используя вышеприведенный метод, Скотт получает описание импликативной системы *n*-значной логики Лукасевича с помощью (*n*-1)-элементного множества оценок

$$VL_n^* = \{v_0, v_1, \ldots, v_{n-2}\}$$

такого, что для любых $i, j \in \{0, 1, \ldots, n-2\}$ и $\alpha \in For^*$ (For^* используется для обозначения множества формул языка $\mathcal{L}^*$, включающего связки отрицания ~ и импликации $\to$*):*

$$(mon) \qquad v_j(\alpha) = t \text{ всегда, когда } v_i(\alpha) = t \text{ и } i \leq j$$

и, более того, $v_0(\alpha_1) \neq f$ и $v_{n-2}(\alpha_2) \neq t$ для некоторых $\alpha_1, \alpha_2 \in For^*$. Нижеприведенная таблица показывает, что множество VL_n^* определяет *n* типов $Z_1, Z_2, \ldots, Z_{n-1}$ высказываний:

	ν_0	ν_1	ν_2	…	ν_{n-3}	ν_{n-2}
Z_0	t	t	t	…	t	t
Z_1	f	t	t	…	t	t
Z_2	f	f	t	…	t	t
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Z_{n-2}	f	f	f	…	f	t
Z_{n-1}	f	f	f	…	f	f

Функция $f(Z_i) = n-i-1/n-1$ является одно-однозначным обратно направленным отображением множества типов в универсум матрицы Лукасевича $\mathfrak{M}_n^L$: Z_0соответствует в матрице 1 и Z_{n-1} соответствует 0 (см. раздел 3.2). Связки отрицания и импликации определяются следующим образом: $Z_I \to Z_i = Z_{max(0,i-1)}$, $\sim Z_I = Z_{n-i-1}$. Соответственно, для любого $i,j,k \in \{0, 1, \ldots, n-2\}$,

$(\sim)$ $\nu_k(\sim\alpha) = t$, если и только если $\nu_{n-k-2}(\alpha) = f$

$(\to)$ $\nu_k(\alpha \to \beta) = t$, если и только если $\nu_j(\beta) = t$ всегда, когда $i+k \le j$ и $\nu_i(\alpha) = t$.

Простое вычисление показывает, что множество всех формул языка $\mathcal{L}^*$, истинных при произвольной оценке $\nu_i \in VL_n^*$, есть в точности содержание матрицы $\mathfrak{M}_n^*$, т.е. (~, →)-редукт матрицы Лукасевича $\mathfrak{M}$:

$$E(\mathfrak{M}_n^*) = \{\alpha \in For^* : \nu_i(\alpha) = t, \text{ для } i \in \{0, 1, \ldots, n-2\}\}.$$

Одновременно, однако, отношение следования $\models_n^* \subseteq 2^{For^*} \times For^*$:

$X \models_n^* \alpha$, *если и только если* $\nu_i(\alpha) = t$ *всегда, когда* $\nu_i(X) \subseteq \{t\}$

для произвольного $\nu_i \in VL_n^*$ не совпадает с $\models_n$ (следование Лукасевича, редуцированное для языка $\mathcal{L}^*$). Для подтверждения этого достаточно проверить, что $\{\alpha\to\beta, \alpha\} \not\models_n^* \beta$ всегда, когда $\{\alpha\to\beta, \alpha\} \models_n \beta$.

$\models_n^*$ называется *условным суждением.* Из этого должно быть понятно, можно ли расширить это (отношение) на весь язык $\mathcal{L}$ и как это можно сделать.

Д. Скотт предлагает, чтобы равенства формы «$\nu_i(\alpha) = t$», для $i \in \{0, \ldots, n-2\}$, читались как «(утверждение) α истинно в степени *i*». Следовательно, он предполагает, что числа в ряду $0 \le i \le n-2$ символизируют *степени заблуждения в отклонении от истины*

(degrees of error in deviation from the truth). Степень 0 - самая сильная и соответствует «совершенной» истине или отсутствию заблуждения: все тавтологии логики Лукасевича являются схемами утверждений, имеющих в качестве своей степени заблуждения 0. Кроме того, импликация Лукасевича может быть удобно истолкована в этих терминах: предположив $i+j \le n-2$, мы получаем, что

$\nu_i(\alpha \to \beta) = t$ и $\nu_j(\alpha) = t$ дает $\nu_{i+j}(\beta) = t$.

Так, используя высказывания $\alpha \to \beta$, можно выразить величину *различия (shift)* между степенями заблуждения посылки и заключения, которая является мерой заблуждения всей импликации. Для подтверждения этого Д. Скотт приводит пример из евклидовой геометрии.

К вопросу об интерпретации импликации Лукасевича → Д. Скотт возвращается в работе *[Scott* 1976] в разделе «Логика заблуждений» (A logic of errors). Здесь Скотт делает замечание о том, что многоместное (multiple) отношение следования

$A_0, A_1 \cdots A_{n-1} \vdash B_0, B_1 \cdots B_{m-1}$

имеет простую интерпретацию в терминах степени заблуждения *i*: *всегда, когда* $i \ge A_t$ *для всех* $t < n$, *тогда* $i \ge B_u$ *для некоторых* $u < m$.

Остается добавить, что логика заблуждений Скотта была подвергнута критике Дж.Смайли *[Smiley* 1976]. Например, указывается, что в подобных терминах нельзя проинтерпретировать операцию отрицания ~.

4.3. Интерпретация Уркварта

А.Уркварт *[Urquhart* 1973] предлагает семантику крипковского типа

$\mathcal{K}_n = \langle S_n, \le, \vdash \rangle$

для некоторых конечнозначных логик, в том числе и для логик Лукасевича $Ł_n$. Он рассматривает отношение $\vdash$ между натуральными числами из множества $S_n = \{0, 1, \ldots, n-2\}$ и формулами: $\vdash \subseteq S_n \times For$. Имеет место следующая

Лемма. *Если* $x \vdash \alpha$ *и* $x \le y \in S_n$, *то* $y \vdash \alpha$.

Роль оценок в $\mathcal{K}_n$ выполняется отображением $F: Var \to 2^{S_n}$, таким, что отношение $\vdash_F$, $x \vdash_F p$, т.т.т., когда $x \in F(p)$, удовлетворяет *лемме.* Отношение $\vdash_F$ расширяется на множество всех формул, соответствующих условиям, зависимым от связок. Тогда формула α является *x-истинной в* $\mathcal{K}_n$, $x \vdash \alpha$, если $x \vdash_F \alpha$ для произвольного F такого, как рассмотрено выше. Формула α является $\mathcal{K}_n$ *истинной,* т.т.т.,

когда она истинна в точке 0, т.е. выполнено, что $0 \vdash \alpha$. $\mathcal{K}_n$ является *семантикой* системы, определенной данной матрицей $\mathfrak{M}_n$, когда множество всех $\mathcal{K}_n$-истинных формул равно содержанию $\mathfrak{M}_n$, т.е. когда $E(\mathfrak{M}_n)=\{\alpha \in For : 0 \vdash \alpha\}$.

Для n-значных логик Лукасевича $\vdash$ должно удовлетворять следующим условиям:

$x \vdash \alpha \to \beta$, т.т.т., когда $y \vdash \alpha$ дает $x+y \vdash \beta$ всегда, когда $x+y \in S_n$

$x \vdash \sim\alpha$ т.т.т., когда $(n-2)-x \nvdash \alpha$

$x \vdash \alpha \vee \beta$ т.т.т., когда $x \vdash \alpha$ или $x \vdash \beta$

$x \vdash \alpha \wedge \beta$ т.т.т., когда $x \vdash \alpha$ и $x \vdash \beta$

$x \vdash \alpha \equiv \beta$ т.т.т., когда $x \vdash \alpha \to \beta$ и $x \vdash \beta \to \alpha$.

Заметим, что «перевод» оценок Скотта из VL* для случая $\vdash$ можно осуществить в соответствии с эквивалентностью:

$i \vdash \alpha$ т.т.т., когда $v_i(\alpha) = t$.

Вместо приведения доказательства эквивалентности между этой модельной теорией и матрицами Лукасевича посмотрим, каким образом Уркварт пытается установить связь между формальной семантикой и интуитивными соображениями.

Элементы S_n рассматриваются как моменты времени, где n -последний элемент в S_n - зафиксирован в качестве некоторой будущей даты, а 0 интерпретирован как настоящий момент. Таким образом, «$x \vdash_F \alpha$» читается как «α доказуемо в момент x». Высказывание может быть доказуемым или не доказуемым в данный момент. Например, высказывание о будущем событии может быть доказуемым или не доказуемым сейчас. Однако, если α доказуемо сейчас, то оно доказуемо и во все последующие моменты. Это означает, что мы думаем о высказываниях не как о неопределенных по времени (temporally indefinite) (например, «Сейчас Линкольн является президентом»), а как об определенных по времени (temporally definite) (например, «Линкольн является президентом в 1971 году н.э.»). До сих пор наше неформальное объяснение, считает Уркварт, находится в соответствии с философской мотивировкой, данной в *[Lukasiewicz* 1930].

При описанной выше интерпретации импликация Лукасевича $\alpha \to \beta$ доказуема в x, если и только если всегда, когда α доказуема в момент y, β доказуема в момент $x+y$ (т.е. в момент на x моментов отстоящий в будущее от y). Формула ~α доказуема в момент x, если и только если α не доказуема в момент, который на x моментов предшествует

последнему моменту в нашем временном ряду. Таким образом, обе связки Лукасевича - «импликация» и «отрицание» - проявляют значительные отличия от обычных операторов импликации и отрицания.

Уркварт говорит, что такой способ понимания выявляет источники трудностей в достижении полностью интуитивной интерпретации многозначных логик Лукасевича, и он утверждает, что «естественные» связки импликации и отрицания скорее должны удовлетворять следующим стандартным условиям.

$x \vdash \alpha \to \beta$ т.т.т., когда для некоторого $y \in S_n$ ($y \vdash \beta$ всегда, когда $x \le y$ и $y \vdash \alpha$), $x \vdash \sim\alpha$ т.т.т., когда $y \vdash \alpha$ не верно для любого $y \in S_n$.

Обратим внимание на рецензию Д.Райна *[Rine* 1974], в которой содержательная интерпретация для $Ł_n$ была подвергнута критике. Райн отмечает, что смысл леммы не всегда согласуется с синтаксисом естественного языка. Рассмотрим следующее утверждение α: «Джон играет в теннис» («John plays tennis»); и пусть {0, ..., n} обозначает временное пространство с того времени, когда Джон впервые играет в теннис (0), и до того времени, когда он последний раз играет в теннис (n). Тогда, продолжает Райн, не ясно, почему не могут существовать x, y, где x < y, такие, что α имеет место во всех $\{0, \ldots, n\text{-}x\}$ и $\{n\text{-}y, \ldots, n\}$, но не между n-x и n-y.

В итоге мы приходим к тому, что любая *содержательная* интерпретация истинностных значений в $Ł_n$ сталкивается с серьезными трудностями. И еще большие трудности возникают, когда это содержание мы пытаемся перенести на интерпретацию логических связок $Ł_n$. Все дело в том, и на это указывает А.Уркварт *{Urquhart* 1986], что логика неопределенностей, логика вероятностей и логика заблуждений не являются истинностно-функциональными логиками, и поэтому подобная интерпретация $Ł_n$ не является адекватной. Напомним, что уже А.Прайор *[Prior* 1957], интерпретируя $Ł_3$ как логику случайности (т.е. третье истинностное значение интерпретируется как случайность), приходит к выводу, что при подобной интерпретации конъюнкция в $Ł_3$ не может быть истинностно-функциональной (см. выше раздел 2.7.). Итак, основная трудность содержательной интерпретации многозначных логик состоит в том, что вкладывая содержание (смысл) в определенное множество истинностных значений, мы затем пытаемся совместить этот смысл с истинностно-функциональным свойством многозначных логик. Это свойство заключается в том (как и для классической

логики), что приписывание истинностного значения сложному высказыванию есть функция от значений элементарных высказываний, входящих в него. Что же касается непосредственно самой $Ł_n$, то оказалось, что она имеет сугубо *теоретико-числовую природу и связана со свойствами простых чисел* (см. следующие главы).
Тем не менее есть выход из создавшегося положения, состоящий в том, чтобы истинностные значения многозначных логик, которыми обычно являются различные множества чисел: дробные числа, натуральные, целые, и т. д., - проинтерпретировать в терминах исходного классического множества истинностных значений T (истина) и F (ложь). По существу впервые такая интерпретация была предложена Постом *[Post* 1921] для его логики $\mathbf{P}_n$, где *n*-значные высказывания интерпретируются в терминах классических высказываний. А уже непосредственно в терминах T-F-последовательностей интерпретация для многозначной логики Лукасевича $Ł_n$ была предложена М.Бёрдом.

4.4. Фактор-семантика

Вначале введем следующие понятия. Пусть $\mathbf{B} = \{\mathbf{T}, \mathbf{F}\}$, т. е. B есть множество классических истинностных значений. Посредством B^s обозначим прямое произведение s раз одинаковых множеств, равных B:

$B^S = B \times B \times \ldots \times B$ (s сомножителей).

Тогда при $s \geq 2$ B^S есть множество всех T-F-последовательностей (булевых векторов) длины s, которое записывается так:

$B^S = \{\langle a_1, \ldots, a_s\rangle \mid a_i \in B\}$, $1 \leq i \leq s$.

Поскольку B есть двухэлементное множество, то число элементов множества B^s равно 2^s. Элементы множества B обозначим посредством α, β, γ с индексами или без них. Алгебра

$\mathcal{A}^B = \langle B^S, \neg^+, \supset^+, \vee^+, \wedge^+\rangle$

есть булева алгебра, где операции $\neg^+, \supset^+, \vee^+, \wedge^+$ определяются на множестве B^s посредством булевых (т. е. классических) операций $\neg, \supset, \vee, \wedge$ следующим образом: для любых T-F-последовтельностей $\alpha = \langle a_1, \ldots, a_s\rangle$ и $\beta = \langle b_1, \ldots, b_s\rangle$

$\neg^+\alpha = \langle \neg a_1, \ldots, \neg a_s\rangle$,

$\alpha \supset^+ \beta = \langle a_1 \supset b_1\rangle, \ldots, \langle a_s \supset b_s\rangle$,

$\alpha \vee^+ \beta = \langle a_1 \vee b_1\rangle, \ldots, \langle a_s \vee b_s\rangle$,

$\alpha \wedge^+ \beta = \langle a_1 \wedge b_1\rangle, \ldots, \langle a_s \wedge b_s\rangle$.

Поскольку компоненты a_i и b_i последовательностей α и β принимают классические истинностные значения T и F (или 1 и 0), то указанные операции над компонентами - это просто логические операции над двоичными переменными. Тогда сами операции $\neg^+, \supset^+, \vee^+, \wedge^+$ естественно называть *покомпонентными* (булевыми) операциями.
Рассмотрим интерпретацию $Ł_n$, предложенную М.Бёрдом. Им вводится одноместный оператор $\mathbf{d}(\alpha)$, который преобразует T-F-последовательности из B^s таким образом, что все вхождения T стоят в начале последовательности:

$\mathbf{d}(\alpha) = \langle \mathbf{T}, \mathbf{T}, \ldots, \mathbf{T}, \mathbf{F}, \mathbf{F}, \ldots, \mathbf{F}\rangle$.

Множество всех таких T-F-последовательностей обозначим посредством B_T^S. Элементы из B_T^S будем обозначать посредством $\alpha^T, \beta^T, \ldots$.
Рассмотрим логическую матрицу

$\mathfrak{M}_{s+1}^L = \langle B_T^S, \mathbf{d}, \neg^{\mathbf{d}}, \rightarrow^{\mathbf{d}}, \{\mathbf{T}^S\}\rangle$,

где операции $\mathbf{d}$, $\neg^{\mathbf{d}}$ и $\rightarrow^{\mathbf{d}}$ определяются следующим образом:

1. $\mathbf{d}(\alpha) = \alpha^T$.

2. $\neg^{\mathbf{d}}(\alpha^T) = \mathbf{d}(\neg^+(\alpha^T))$.

3. $\alpha^T \rightarrow^{\mathbf{d}} \beta^T = \mathbf{d}(\alpha^T \supset^+ \beta^T)$.

Имеет место следующая

Теорема 1. *Матрицы* $\mathfrak{M}_n^L = \langle V_n, \sim, \rightarrow, \{1\}\rangle$ *и* $\mathfrak{M}_{s+1}^L = \langle B_T^S, \mathbf{d}, \neg^{\mathbf{d}}, \rightarrow^{\mathbf{d}}, \{\mathbf{T}^S\}\rangle$ *изоморфны, где* $\mathfrak{M}_n^L$ *есть матрица для n-значной логики Лукасевича.*

Таким образом, имеется интерпретация истинностных значений, будь то натуральные числа или дробные, в терминах классических истинностных значений T и F. Например, истинностное значение 0 интерпретируется T-F-последовательностью, в которой все вхождения есть F; 1/3 - последовательностью <T, F, F>, т. е. числитель указывает

на число вхождений T, а знаменатель есть длина последовательности, обозначенная числом *s (- n-1).*

Обратим внимание, что результат Теоремы 1 имеет также место, если множество истинностных значений B_T^S заменим на множество B_F^S, т. е. оператор d перерабатывает каждую T-F-последовательность в такую, что все вхождения F стоят в начале. Тогда истинностное значение 1/3 интерпретируется T-F-последовательностью <F, F, T>. Поэтому имеет смысл обобщить подобную интерпретацию так, чтобы она строилась независимо от выбора множества T-F-последовательностей в качестве истинностных значений. Такую интерпретацию мы назвали *фактор-семантикой [Karpenko* 1983], и строится она следующим образом

С булевой алгеброй

$$\mathcal{A}^B = < B^S, \neg^+, \supset^+, \vee^+, \wedge^+ >$$

ассоциируем логическую матрицу

$$\mathfrak{M}_s^c = < B^S, \neg^+, \supset^+, \vee^+, \wedge^+, \{T^S\} >.$$

Последняя есть не что иное, как прямое произведение классической двузначной матрицы

$$\mathfrak{M}_2^c = < \{T, F\}, \neg, \supset, \vee, \wedge, \{T\} >$$

s раз на саму себя.

Для любого $\alpha \in B^S$ обозначим через $\eta(\alpha)$ число компонент элемента α, которые равны T. Тогда $\alpha \cong \beta$, если $\eta(\alpha) = \eta(\beta)$ и $B^S/\cong$ есть фактор-множество множества B^s по отношению эквивалентности $\cong$. Очевидно, что мощность множества $B^S/\cong$ равна s+1. Если $\alpha \in B^S$, тогда $|\alpha|$ будет обозначать класс эквивалентности, определенный по α. Фактор-множество $B^S/\cong$ снабдим операциями $\neg^L$ и $\rightarrow^L$ следующим образом: для $|\alpha|, |\beta| \in B^S/\cong$ пусть $\neg^L|\alpha| = |\neg^+\alpha|$ и $|\alpha| \rightarrow^L |\beta| = |\alpha' \supset^+ \beta'|$, где $\alpha' \in |\alpha|, \beta' \in |\beta|$ и $\alpha' R^L \beta'$, причем отношение R^L определяется так: $<a_1, .. , a_s> R^L <b_1, ..., b_s>$ т.т.т., когда

1) $\forall i \le s\ (a_i = T \Rightarrow b_i = T)$, если $\eta(\alpha) \le \eta(\beta)$,

2) $\forall i \le s\ (b_i = T \Rightarrow a_i = T)$, если $\eta(\alpha) > \eta(\beta)$.

Заметим, что отношение R^L является отношением толерантности, т.е. оно рефлексивно и симметрично, но в общем случае не транзитивно. В этом обнаруживается еще один неожиданный аспект импликации Лукасевича.

Таким образом, после операции «факторизации» и определения логических операций на полученных классах эквивалентности матрица

$$\mathfrak{M}_s^c = < B^S, \neg^+, \supset^+, \vee^+, \wedge^+, \{T^S\} >$$

преобразуется в матрицу

$$\mathfrak{R}_{s+1}^L = < B^S/\cong, \neg^L, \rightarrow^L, \{|T^S|\} >$$

(операции дизъюнкции и конъюнкции как выразимые через исходные здесь опустим).

Теорема 2. *Матрицы* $\mathfrak{M}_n^L$ *и* $\mathfrak{R}_{s+1}^L$ *изоморфны.*

ДОКАЗАТЕЛЬСТВО. Требуемый изоморфизм достигается посредством отображения φ такого, что для $|\alpha| \in B^S/\cong$

$$\varphi(|\alpha|) = \frac{\eta(\alpha)}{s}.$$

Очевидно, что φ есть взаимнооднозначное соответствие. Покажем, что изоморфизм имеет место, т.е.

$$(*)\quad \varphi(\neg^L|\alpha|) = \sim\varphi(|\alpha|),$$

$$(**)\ \varphi(|\alpha| \rightarrow^L |\beta|) = \varphi(|\alpha|) \rightarrow \varphi(|\beta|).$$

Следующая последовательность равенств является доказательством (*).

$$\varphi(\neg^L|\alpha|) = \varphi(|\neg^+\alpha|) = \frac{s-\eta(\alpha)}{s} = 1 - \frac{\eta(\alpha)}{s} = 1 - \varphi(|\alpha|) = \sim \varphi(|\alpha|).$$

Для доказательства (**) возьмем $\alpha' \in |\alpha|$ и $\beta' \in |\beta|$ такие, что $\alpha' R \beta'$.

(1) $\eta(\alpha) \le \eta(\beta)$. Тогда очвидно, что правая часть (**) равна 1. Далее, $|\alpha| \rightarrow^L |\beta| = |\alpha' \supset^+ \beta'| = |T^S|$. Следовательно, левая часть (**) равна $\varphi(|T^s|) = 1$, что и требовалось доказать

(2) $\eta(\alpha) > \eta(\beta)$. Тогда правая часть (**) в силу определения φ и $\rightarrow$ равна $-\frac{\eta(\alpha)}{s} + \frac{\eta(\beta)}{s}$. Но согласно определению $\rightarrow^L$ и $\supset^+$ число вхождений T в $\alpha' \supset^+ \beta'$ равно $\eta(\beta) + (s - \eta(\alpha))$. Следовательно, левая часть (**) также равна $-\frac{\eta(\alpha)}{s} + \frac{\eta(\beta)}{s}$. Теорема доказана

Таким образом, логическая матрица

$$\mathfrak{R}_{s+1}^L = < B^S/\cong, \neg^L, \rightarrow^L, \{|T^S|\} >$$

является характеристической для *n*-значного исчисления логики Лукасевича $\mathbf{L_n}$ (см. аксиоматизацию в разделе 3.4).

Главный смысл фактор-семантики заключается в том, что теперь в качестве истинностных значений выступают определенные подмножества s-членных T-F-последовательностей из множества B^s. Например, истинностное значение 1/3 интерпретируется множеством $\{<T, F, F>, <F, T, F>, <F, F, T>\}$.

В общем случае мощность множества $|\alpha| \in B^s/\cong$ вычисляется по формуле для биноминальных коэффициентов

$$C_m^k = \frac{m!}{k!(m-k)!}.$$

В нашем случае $k = \eta(\alpha)$ и $m = s$. Тогда, например, мощность множества $|\alpha|$, состоящего из T-F-последовательностей длиной $s = 5$, в каждую из которых число вхождений T есть $\eta(\alpha) = 3$, равно 10.

5. ЛОГИКА КАК ФУНКЦИОНАЛЬНАЯ СИСТЕМА

Существуют два основных способа изучения логических систем: внешний и внутренний. К первому относится представление логики в виде класса аксиоматизируемых тавтологий или в виде класса аксиоматизируемых тождеств (многообразий). Последнее относится к алгебраической семантике логических исчислений. Однако, как и все другие семантики, рассмотренные в предыдущей главе, этот способ не позволяет заглянуть в истинную сущность логики. Это можно сделать только представив логику в виде функциональной системы, что мы и назвали внутренним способом. Есть исходное множество логических функций и определенная на этом множестве одна-единственная операция — суперпозиция. Так мы приходим к пониманию логики как алгебры функций. Именно на этом пути в 1970 г. В.К.Финн обнаружил, что конечнозначные логики Лукасевича $Ł_{n+1}$ по своим внутренним свойствам являются функционально предполными т.т.т., когда n есть простое число. Развитию этого результата посвящаются дальнейшие главы книги.

5.1. Логики Поста

Многозначная логика строилась Постом *[Post* 1921] как обобщение двузначной классической логики, развитой А.Уайтхедом и Б.Расселом в «Principia Mathematica», где исходными логическими связками служат отрицание и дизъюнкция. В логике Поста, которую будем обозначать $\mathbf{P}_n$, исходными связками также являются отрицание $\neg$ и дизъюнкция $\vee$, и каждая пропозициональная переменная может принимать одно из n различных значений истинности: $t_1, t_2, \ldots, t_n$, где n - натуральное число; t_1 интерпретируется как «истина», t_n - как «ложь».

Однако мы предпочтем стандартное определение n-значной матрицы Поста $\mathfrak{M}_n^P$, которая задается следующим образом:

$\mathfrak{M}_n^P = \langle V_n, \neg, \vee, \{n\text{-}1\} \rangle$, где

V_n есть множество $\{0, 1, 2, \ldots, n\text{-}1\}$, $\{n\text{-}1\}$ - множество выделенных значений. Матричные операции определяются так:

x	$\neg$x
0	1
1	2
.	.
.	.
.	.
n-2	n-1
n-1	0

Отрицание Поста $\neg x$ называется *циклическим* отрицанием:

$$\neg x = x + 1 (mod\ n),$$

$$x \vee y = max(x, y).$$

Рассмотрим конкретный пример n-значной логики Поста, а именно $\mathbf{P}_3$, где истинностные значения для удобства обозначим, как в трехзначной логике Лукасевича $Ł_3$; и пусть выделенным значением будет 1

Тогда $x \vee y$ имеет истинностную таблицу точно такую же, как $x \vee y$ в $Ł_3$, но существенное отличие заключается в операции отрицания:

x	$\neg$x
1	$^1/_2$
$^1/_2$	0
0	1

x	$\neg\neg$x
1	0
$^1/_2$	1
0	$^1/_2$

Заметим, что в отличие от $Ł_3$ и G_3 в $\mathbf{P}_3$ (как и во всех $\mathbf{P}_n$) есть операция (в данном случае циклическое отрицание $\neg$), которая на множестве классических истинностных значений {0, 1} принимает отличное от них значение, например, $\neg 1 = {}^1/_2$. Поэтому можно указать формулу, которая является тавтологией в $\mathbf{P}_3$, но не является тавтологией классической логики $\mathbf{C}_2$

$$\neg\neg\neg(x \vee (\neg x \vee \neg\neg x)).$$

Отметим также, что не всякая тавтология $\mathbf{C}_2$ с исходными связками отрицания и дизъюнкции является тавтологией $\mathbf{P}_3$, например в $\mathbf{P}_3$, как и

в $Ł_3$, не имеет места закон исключенного третьего $x \vee \neg x$, но зато имеет место закон исключенного четвертого:

$x \vee \neg x \vee \neg\neg x$.

И вообще, в P_n имеет место закон исключенного (n+1)-го.

Самое главное свойство многозначной логики Поста P_n заключается в том, что она функционально полна, т. е. любая операция многозначной логики в ней определима. С функционально полными логиками мы уже встречались: это n-значные логики Лукасевича $Ł_n$ с различными обобщениями функтора Слупецкого T(x) (см. раздел 3 3 5). Как частный случай, имеем: $Ł_3^T = P_3$.

5.1.1. Функциональная полнота P_n

Сначала покажем, как в матрице $\mathfrak{M}_n^P$ определяются наиболее важные операции, а именно константы из V_n, унарные операции $J_i(x)$ и $\sim x$, и бинарная операция $x \wedge y$.

Заметим, что последовательное применение операции $\neg$, дает

$x + 1 = \neg x$,

$x + 2 = \neg^2 x$,

$x + 3 = \neg^3 x$,

. . .

$x + (n-1) = (x + (n-2)) + 1 = \neg^{n-1} x$,

и в итоге

$\neg^0 = x = \neg^n x$.

Следовательно,

$n-1 = max(x, x+1, ..., x+n-1)$

и

$k = \neg^{k+1}(n-1)$ для $k = 0, 1, 2, ..., n-1$.

Далее, пусть

$$J_i(x) = \neg\Big(\bigwedge_{t \neq n-1-i} \{x + t\}\Big).$$

Если x = i, тогда $J_i(x) = n-1$

Будем придерживаться последней работы.

Если $x \neq i$, тогда

$$J_i(x) = \neg \bigwedge_{t \neq n-1-i} \{x + t\} + 1 = (x + (n-1) - x) + 1 = 0.$$

Следовательно,

$$j_i(x) = \begin{cases} n-1, & \text{если } x = i \\ 0, & \text{если } x \neq i. \end{cases}$$

Наконец, определим функции $j_{s,i}$:

$j_{s,i}(x) = \neg s + (J_i(x) \vee n\text{-}1\text{-}s)$.

Предположим, $f: V_n \to V_n$ есть любая функция. Тогда

$$f(x) = \bigvee_{i=0}^{n-1} J_{f(i),i}(x).$$

В частности,

$$\sim x = \bigvee_{i=0}^{n-1} f_{n-1-i,i}(x).$$

Следовательно,

$x \wedge y = \sim(\sim x \vee \sim y)$.

Теперь покажем, что множество связок n-значной логики Поста P_n является *функционально полным*.

Известно, что в n-значной логике есть аналог совершенной дизъюнктивной нормальной формы (с д н. ф):

$$f(x_1, ..., x_m) = \bigvee_{(\sigma_1, ..., \sigma_m)} J_{\sigma_1}(x_1) \wedge ... \wedge J_{\sigma_m}(x_m) \wedge f(\sigma_1, ..., \sigma_m),$$

где дизъюнкция берется по всем возможным наборам $\sigma_1, ..., \sigma_m$ значений переменных $x_1, ..., x_m$. Это значит, что формулу n-значной логики над множеством элементарных операций $\{0, 1, ..., n-1, J_0(x), ..., J_n(x), x \vee y, x \wedge y\}$ можно преобразовать в с.д.н.ф.

Доказательство полностью аналогично доказательству для случая $n = 2$. Выше мы показали, что указанное множество операций выразимо посредством $\neg x$ и $x \vee y$. Отсюда следует функциональная полнота логики Поста P_n.

5.2. Оператор замыкания, полнота и нредполнота классов функций

Многие специалисты, связанные с вычислительной техникой, инженеры, прикладные математики и физики представляют модели многозначной логики в виде *функциональной системы [Кудрявцев* 1981], обозначаемой *(P_n,* C), где P_n есть множество всех функций n-значной логики Поста (или множество всех функций счетнозначной логики P_ω) с заданной на нем операцией суперпозиции **C**. Функция, полученная из функций $f_1, ..., f_k$ подстановкой их друг в друга и/или

переименованием аргументов, называется *суперпозицией* $f_1, \ldots, f_k$ *[Яблонский* 1959].

Пусть $F \subseteq P_n$. Множество всех формул над F (определение формулы над F задается индуктивно) обозначим через $<F>$. Каждой формуле над F сопоставляется некоторая функция из P_n, которую по определению реализует эта формула. Множество всех формул $<F>$ приводит к множеству $[F]$ функций, реализуемых формулами из $<F>$, и называется суперпозициями над F. Множество $[F]$ называется *замыканием* класса функций F, если оно содержит все суперпозиции функций над классом F и не содержит никаких других функций. Таким образом, функциональная замкнутость множества F означает, что всякая функция, представимая через функции из F, также принадлежит F.

Оператор замыкания $[\ ]$ удовлетворяет следующим условиям:

(i) $F \subseteq [F]$,

(ii) $[[F]] = [F]$,

(iii) $F_1 \subseteq F_2$ **влечет** $[F_1] \subseteq [F_2]$,

(iv) **Множество функций замкнуто, если** $F = [F]$.

Примеры

1. Класс $F = P_n$, очевидно, является замкнутым классом.
2. Класс функций от одной переменной, очевидно, является замкнутым классом.
3. Класс функций от двух переменных не есть замкнутый класс, поскольку суперпозицией дизъюнкции $x \vee y$ является функция от трех переменных $x \vee y \vee z$.

Можно сказать, что замкнутость класса функций F обозначает собою сохранение при суперпозиции «наследственных» свойств этих функций. Пример (3) как раз показывает, что свойство быть функцией от двух переменных при суперпозиции не сохраняется.

Теперь дадим другое, качественно отличающееся от предыдущего понимания, определение многозначной логики: *n-значной логикой, порожденной множеством F, называется тройка* $(F, <F>, [F])$. Однако в ряде задач в этой тройке изучаются только соответствия между F и $[F]$, а множество $<F>$ выступает лишь как средство, определяющее оператор $[\ ]$. Тем самым фактически переходят к изучению оператора замыкания, а сама функциональная система $(P_n, \mathbf{C})$, частным случаем которой является классическая логика $(\mathbf{C}_2, \mathbf{C})$, зачастую отождествляется с многозначной логикой, т. е.

$(P_n, \mathbf{C})$ выступает в качестве модели многозначной логики. Эта модель, в отличие от рассмотренных выше алгебр истинностных значений, является *алгеброй функций.* Именно такой подход к изучению логик Лукасевича $\mathbf{L_n}$ станет для нас главенствующим.

Труднейшей проблемой при изучении функциональных систем является проблема конструирования новых функций из данного множества функций. Проблема эта возникает и в пропозициональном исчислении, представленном формульной моделью, и в синтезе автоматов, и в универсальной алгебре; но именно в логике, представленной в виде функциональной системы, этой проблеме уделяется специальное внимание. Н.Р.Емельянов *[Емельянов* 1985] показал, что в n-значной логике для любого фиксированного $n>2$ задача о выразимости функции посредством операции суперпозиции через функции определенной системы является NP трудной задачей, т е. для её решения не существует полиномиальных алгоритмов. В дальнейшем мы напрямую столкнемся с этой проблемой, а именно с необходимостью выражения нужной функции. В результате будут появляться формулы (суперпозиции), содержащие тысячи и даже миллиарды вхождений исходных связок.

Важнейшим свойством функциональной системы является свойство функциональной полноты, необходимое, например, в случае реализации любой переключательной схемы. В первую очередь этим объясняется наличие огромной литературы по алгебрам Поста.

Система функций $F = \{f_1, \ldots, f_k, \ldots\}$ из P_n называется функционально полной, если любая функция из P_n представима посредством суперпозиций функций из системы F. В связи с этим указанную выше проблему можно сформулировать так: является ли некоторое множество F функционально полным?

Примеры:

1. Система $F = P_n$ полна. Очевидно, что множество всех функций из P_n представляет полную систему.
2. Система Россера и Тюркетта $F = \{0, 1, \ldots, n\text{-}1, J_0(x), J_{n-1}(x), x \vee y, x \wedge y\}$ является полной в P_n (см. раздел 4).
3. Система Поста $F = \{\neg x, x \vee y\}$ полна в P_n. Для доказательства нужно посредством суперпозиций функций $\neg x$ и $x \vee y$ выразить все функции системы Россера и Тюркетта (см. раздел 4).

Обычно доказательство полноты конкретных систем в P_n производится с помощью метода сведения к заведомо полным системам.

Существует, кроме того, ряд *признаков полноты,* в которых рассматриваются множества функций, содержащих некоторые совокупности функций от одной переменной и еще только одну

функцию, существенно зависящую не менее чем от двух переменных. Функция $f \in P_n$ называется *существенной,* если она зависит не менее чем от двух переменных и принимает все *n* значений из множества V_n.
В терминах замыкания можно дать другое определение полноты, эквивалентное исходному. Система *G* функций из замкнутого класса *F* полна в *F* (порождает *F),* если $[G] = F$. Система функций *F* полна в P_n, если $[F] = P_n$.
Понятие замкнутого класса может быть применено к решению вопросов об обосновании неполноты некоторых систем. Например, рассмотрим систему $F = \{\sim x, x \vee y\}$. Из определения операций $\sim x$ и $x \vee y$ следует, что обе функции принадлежат к классу функций, сохраняющих множество истинностных значений $\{0, n\text{-}1\}$, т.е. для любого набора σ, состоящего из 0 и *n-1,* значение функции $f(\sigma)$ является 0 или *n*-1. Очевидно, что $F = [\sim x, x \vee y]$ является примером еще одного замкнутого класса. В силу сохранения множества $\{0, n\text{-}1\}$ замкнутый класс $[\sim x, x \vee y]$ не содержит, например, константу 1. Значит, при $n \geq 3$ *F* не будет полной системой. На этом примере видно, что хотя система $\{\sim x, x \vee y\}$ и является обобщением системы $\{\neg x, x \vee y\}$ булевых функций, она не является полной. Заметим также, что система функций $\{\sim x, x \to y\}$ тоже принадлежит к классу функций, сохраняющих $\{0, n\text{-}1\}$. В силу этого *n*-значная логика Лукасевича $Ł_n$ не является функционально полной при $n \geq 3$.
В общем случае распознавание того, является ли произвольная система функций *F* полной или нет, является сожнейшей технической проблемой для *n*-значных логик. Выделяются два подхода к решению задачи о полноте. В первом случае ставится вопрос о существовании алгоритма, устанавливающего полноту или неполноту системы функций; во втором - рассматривается совокупность всех предполных классов функций в P_n. Система *F* функций называется *предполной* в P_n, если *F* представляет не полную систему, но добавление к *F* любой функции *f* такой, что $f \in P_n$ и $f \notin F$, преобразует *F* в полную систему. Или, в терминах замыкания: F предполна в P_n, если $[F] \neq P_n$ и $[F \cup \{f\}] = P_n$, где $f \in P_n$ и $f \notin F$. Важная роль предполных классов функций видна из следующей теоремы, которая формулирует *критерий функциональной полноты:* система функций *F* *n*-значной логики полна тогда и только тогда, когда она не содержится целиком ни в одном предполном классе.
Известно, что в булевой алгебре функций, т.е. в C_2, существует только 5 предполных классов. Проблема полноты для P_3 была решена С. В.

Яблонским *[Яблонский* 1954], где полностью описываются все 18 предполных в P_3 классов. В *[Яблонский* 1958] получено также описание нескольких серий предполных классов в случае $n \geq 4$. Этот список расширялся рядом авторов, пока И.Розенбергом *[Rosenberg* 1965] было анонсировано, а в *[Rosenberg* 1970] дано описание всех предполных классов в *n*-значной логике. Как установлено в *[Захарова, Кудрявцев, Яблонский* 1969], количество предполных в P_n классов с ростом *n* растет сверхэкспоненциально. Хотя число предполных классов $\pi(n)$ конечно для любого *n,* однако очень быстрый их рост указывает на малую практическую эффективность предполных классов для решения проблемы полноты, что видно из следующей таблицы *[Rosenberg* 1973]:

n	2	3	4	5	6	7	8
$\pi(n)$	5	18	82	643	15 182	7 848 984	$>5\,10^{11}$

5.2.1. Максимальная *n*-значная непостовская логика

Особый интерес представляет следующий класс функций. Пусть T_n обозначает множество всех функций из P_n, которые сохраняют 0 и *n*-1, т. е. $f(x_1, ..., x_m) \in T_n$ т.т.т., когда $f(x_1, ..., x_m) \in \{0, n\text{-}1\}$, где $x_i \in \{0, n\text{-}1\}$, $1 \leq i \leq m$. Из теоремы Яблонского о функционально предполных классах функций в *n*-значной логике *[Яблонский* 1958] следует, что данный класс функций T_n является предполным.
Примером предполной в P_3 логики является трехзначная логика Лукасевича $Ł_n$. Последнее следует из работы В. К. Финна *[Финн* 1969] о критерии функциональной полноты $Ł_3$, где показано, что $Ł_3 = T_3$.
Рассмотрим матричное определение логики T_n, соответствующей множеству функций T_n, которую В.К.Финн *[Finn* 1975] называет «максимальной *n*-значной непостовской логикой»:

$$\mathfrak{M}_n^T = \langle V_n, \sim x, x \wedge y, J_0(x), ..., J_{n-1}(x), N_1(x), ..., N_{n-2}(x), \{n\text{-}1\} \rangle,$$

где
$\sim x$, $x \wedge y$ и $J_i(x)$ - функции, определенные выше,

$$N_i(x) = \begin{cases} i, \text{если } x \in \{1, ..., n-2\} \\ \sim x, \text{если } x \in \{0, n-1\} \end{cases} \quad (1 \leq i \leq n-2).$$

(Обратим внимание, что полужирным шрифтом у нас обозначаются логические ситемы, а курсивом - множество всех суперпозиций,

полученных из исходных функций этой системы. Например, множество всех суперпозиций, полученных из исходных функций ~ и → логики Лукасевича $Ł_n$, будем обозначать посредством $Ł_n$.)

Сигнатуру матрицы $\mathfrak{M}_n^T$ можно значительно упростить. Пусть

$\mathfrak{M}_n^{T^*} = < V_n, \sim x, x \to^{T^*} y, \{n\text{-}1\}$, где

1) если $n = 3$, то $x \to^{T^*} y = x \to y$;

2) если $n > 3$, то

$$x \to^{T^*} y = \begin{cases} n-2, \text{ если } x = y \text{ и } x, y \in \{1,\ldots,n-2\} \\ x \to y, \text{ в остальных случаях.} \end{cases}$$

Множество всех функций матрицы $\mathfrak{M}_n^{T^*}$ обозначим посредством T_n^*.

Теорема. $T_n^* = T_n$ *для любого* $n \geq 3$.

ДОКАЗАТЕЛЬСТВО.

I. $T_n \subseteq T_n^*$

Сначала определим в T_n^* импликацию Лукасевича $x \to y$:

$$x \to y = \sim((y \to^{T^*} x) \to^{T^*} \sim(y \to^{T^*} x)) \to^{T^*} (x \to^{T^*} y).$$

Легко показать, что

$$x \to y = \sim((y \to x) \to \sim(y \to x)) \to (x \to y).$$

Поскольку $x \to^{T^*} y$ отличается от $x \to y$ только для случая, когда $x = y$ и $x, y \in \{1, \ldots, n\text{-}2\}$, то остается проверить этот случай. Тогда имеем

$$x \to y = \sim((n\text{-}2) \to^{T^*} ((n\text{-}1)\text{-}(n\text{-}2))) \to^{T^*} (n\text{-}2) =$$

$$= \sim((n\text{-}2) \to^{T^*} 1) \to^{T^*} (n\text{-}2) = ((n\text{-}1)\text{-}2) \to^{T^*} (n\text{-}2) = n\text{-}1.$$

Следовательно, $Ł_n \subseteq T_n^*$. Поскольку $x \wedge y \in Ł_n$, то $x \wedge y \in T_n^*$.

Как уже отмечалось, Б. Россер и А. Тюркетт *[Rosser & Turquette* 1952] показали, что для любого $n \geq 3$ и любого $i \in V_n$, $J_i(x) \in Ł_n$.

Отсюда, $J_i(x) \in T_n^*$. Остается показать, что $N_i(x) \in T_n^*$.

1) $n = 3$. Тогда

$$N_1(x) = \sim x;$$

2) $n \geq 3$. Тогда

$$N_1(x) = (x \to^{T^*} x) \to^{T^*} J_0(x),$$

$$N_2(x) = (x \to^{T^*} x) \to^{T^*} N_1(x),$$

. . . . . .

$$N_{n-2}(x) = (x \to^{T^*} x) \to^{T^*} N_{n-3}(x),$$

Такимобразом, $T_n \subseteq T_n^*$.

II. $T_n^* \subseteq T_n$.

Выше мы показали, что T_n^* включает в себя T_n. Но T_n является функционально предполньш в P_n множеством функций для любого $n \geq 3$. Поскольку T_n^* не является функционально полным множеством функций (функции $\sim x$ и $x \to^{T^*} y$ сохраняют множество значений $\{0, n\text{-}1\}$), то $T_n^* \subseteq T_n$.

Таким образом, $T_n^* = T_n$.

5.2.2. Базисы. Штрих Шеффера для P_n

К проблеме полноты примыкает задача о базисах, состоящая в указании всех полных в замкнутом классе $F \subseteq P_n$ подмножеств, никакое собственное подмножество которых уже не полно в F, т.е. *базисом* является минимальная полная независимая система функций, удаление из которой любой функции делает систему неполной. Примером базиса в P_2 служит система функций $\{x \wedge y, 0, 1, x \oplus y \oplus z\}$ и $\{x \vee y, \neg x\}$, но не $\{x \vee y, x \supset y, \neg x\}$, поскольку $x \supset y = \neg x \vee y$. Представляют интерес базисы, состоящие из одной функции, которые называются *штрихом Шеффера*. Пусть $F \subseteq P_n$. Тогда функция f из P_n есть штрих Шеффера (или единственный генератор) для F, если любая функция из F выразима посредством конечного числа суперпозиций функции f или, по-другому, если $[f] = F$. В P_n базисом, состоящим из одной-единственной функции, является функция Вебба $W_n(x, y)$ [Webb 1935],

которая определяется посредством исходных операций *n*-значной логики Поста следующим образом:

$$W_n(x, y) = \neg(x \vee y).$$

Или, по-другому:

$$W_n(x, y) = max(x, y) + 1 (mod\ n).$$

(В *[Webb* 1936] дается простое доказательство функциональной полноты для $W_n(x, y)$. посредством определения исходных операций *n*-значной логики Поста.)

Если в P_2 имеются только две полные системы, состоящие из одной функции: штрих Шеффера $x|y$ и стрелка Пирса $x \uparrow y$, с которой при $n = 2$ функция Вебба совпадает (см. выше раздел 1.3.1), то в P_3 таких функций имеется 3774 и 90 из них коммутативны *[Rousseau* 1967].

И. Розенберг *[Rosenberg* 1978] приводит библиографию по функциям Шеффера включительно по 1978 г. Теория двухместных штрихов Шеффера для функционально полных *n*-значных логик и эффективные правила их построения даны в*[Pmkava* 1981].

Дадим пример характеристического свойства функции Шеффера для P_n (см. *[Яблонский* 1986]): *функция* $f(x_1, \ldots, x_m)$ *из* P_n, *где* $n \geq 3$, *является функцией Шеффера т.т.т., когда* $f(x_1, \ldots, x_m)$ *порождает все функции одной переменной, принимающие не более n-1 значений.*

Отсюда следует еще один критерий функциональной полноты для P_n.

5.2.3. Штрих Шеффера для Ł$_n$

В каждом случае, однако, возникает проблема построения функции Шеффера для функционально неполных логик. Интересен следующий результат. Дж. Мак-Кинси *[McKinsey* 1936] сконтруировал штрих Шеффера для *n*-значной логики Лукасевича $Ł_n$:

$$Exy = CxC\{CNy\}yNCyN\{Cy\}Ny,$$

где C и N - импликация и отрицание в нотации Лукасевича, а скобки указывают на *n-2* вхождение заключенного в них выражения. Для единообразия обозначим функцию *E*xy как $x \to^E y$.

Используя $J_i(x)$-функции, которые не были известны Дж.Мак-Кинси, можно значительно упростить определение $x \to^E y$. Заметим, что $J_{n-1}(y) = N\{Cy\}Ny$ и $J_0(y) = N\{CNy\}y$. Тогда

$$x \to^E y = x \to ((y \to J_{n-1}(y)) \to J_0(y)).$$

Применив контрпозицию к консеквенту, получим нужное определение.

$$x \to^E y = x \to ((y \to J_{n-1}(y)) \to J_0(y)).$$

Для сравнения с импликацией Лукасевича $x \to y$ определим $x \to^E y$ следующим образом:

$$x \to^E y = \begin{cases} x \to^E 0 = n-1 \text{ и } x \to^E n-1 = (n-1) - x \text{ для всех } x \\ x \to y, \text{ в остальных случаях}. \end{cases}$$

Теперь нужно посредством $x \to^E y$ определить функции ~x и $x \to y$. Мак-Кинси это делает следующим образом:

$$(a)\ n\text{-}1 = (x \to^E x) \to^E ((x \to^E x) \to^E (x \to^E x)),$$

$$(b)\ \sim x = x \to^E n\text{-}1,$$

$$(c)\ x \to y = x \to^E (n\text{-}1 \to^E y).$$

Множество всех суперпозиций функции $x \to^E y$ обозначим посредством E_n. Таким образом, $E_n = Ł_n$ для любого $n \geq 2$.

Мак-Кинси показывает, что функция Вебба не может быть определена в терминах ~x и $x \to y$, за исключением случая, когда *n*+1=2. Таким образом, здесь впервые дано доказательство того, что множество функций $Ł_n$ не является функционально полным ни для какого $n \geq 2$.

На это же обращает внимание в рецензии на эту статью Х.Карри *[Curry* 1937]. Более того, Карри указывает на важное различие между результатами Вебба и Мак-Кинси, которое заключается в том, что функция Шеффера у Мак-Кинси сама определяется через исходные, а у Вебба нет. На это различие также обращает внимание в рецензии на эту же статью У.Куайн *[Quine* 1937], назвав функцию Вебба «чуждой» (foreign) к $Ł_n$, хотя посредством функции Вебба исходные функции $Ł_n$ определяются.

Учитывая замечание Куайна, Х.Хендри и Дж.Массей *[Hendry & Massey* 1969] предлагают назвать функцию *f* «настоящей» (indigenous) функцией Шеффера для $F \subseteq P_n$, если *f* есть штрих Шеффера для *F* и в добавление к этому *f* определима посредством *F*. Нас будет интересовать именно такой штрих Шеффера.

Исследования штриха Шеффера для $Ł_n$ работой Мак-Кинси не закончились. А.Роуз *[Rose* 1952] обратил внимание, что функция $x \to^E y$ не является коммутативной, и предложил новое определение штриха Шеффера для $Ł_n$, которое значительно проще:

$$x \to^D y = x \to^D \sim y.$$

Однако новый штрих Шеффера не имеет места для случая, когда $n = 3i$, поскольку *i* тогда является неподвижной точкой.

Интересно, что точно такой же штрих Шеффера для $Ł_n$ был построен в *[Hendry & Massey* 1969]. Наконец, в другой работе *[Rose* 1968] (через шестнадцать лет) Роуз доопределяет функцию $x \to^{D} y$ таким образом, что она является штрихом Шеффера для любого множества $Ł_n$.Однако в разделе 6.3 мы будем ориентироваться на способ определения штриха Шеффера для $Ł_n$, предложенного Мак-Кинси.

Отметим также, что штрих Шеффера для T_n^* построен в *[Карпенко* 1989].

Известно, что число базисов в P_2 равно 42, и мощность базиса не превышает 4 (как раз пример такого базиса мы приводили). В связи с этим интересные результаты содержит работа М.Миякавы *Miyakawa* 1981]. Он установил, что число базисов в P_3 равно 6 763 769, причем мощность базиса (максимальный ранг) не превышает 6. При этом указано число базисов для каждого из рангов от 1 до 6 (расчеты велись с помощью ЭВМ). Таким образом, выяснена точная структура P_3, в связи с чем возникает сложнейшая проблема перечисления базисов для произвольного $n>2$.

5.2.4. Континуальность $Ł_3$

Глобальной задачей для многозначной логики как функциональной системы остается описание решетки замкнутых классов данной модели многозначной логики. Для двузначной логики эта задача полностью решена Э. Постом в начале 20-х годов. В *[Post* 1921] установлено, что мощность множества замкнутых классов в P_2 счетна, а в *[Post* 1941] дается полное описание решетки замкнутых классов, каждый класс строится эффективно и показано, что каждый замкнутый класс имеет конечный базис. Эти классы названы *классами Поста.* Из этих результатов следуют решения задач о выразимости, полноте, базисах и др. В этой же работе поставлен вопрос об описании всех замкнутых классов в P_n.

Однако с многозначной логикой дело обстоит совсем по-другому. Оказалось, что имеются существенные различия между классической двузначной логикой и многозначной, говорящие *о принципиальной несводимости второй к первой.* Ю.И.Янов показал, что в отличие от P_3 для всякого $n \geq 3$ существует в P_n замкнутый класс, не имеющий базиса *[Янов & Мучник* 1959]; в свою очередь А. А.Мучник показал, что для всякого $n \geq 3$ существует в P_n замкнутый класс со счетным базисом *[Янов & Мучник* 1959]. Непосредственно к этому примыкает следующий примечательный результат: *для всякого* $n \geq 3$ P_n

содержит континуум различных замкнутых классов, т. е. уже P_3 содержит континуум различных замкнутых классов. Вообще-то говоря, точная природа такого различия между двузначной и трехзначной логиками неясна. В силу указанной трудности (континуальности множества замкнутых классов) исследуется «локальная» информация о структуре окрестности некоторого произвольного замкнутого класса. В связи с тем, что уже P_3 содержит континуум различных замкнутых классов, возникает вопрос о мощности замкнутых классов других трехзначных логик. Здесь особый интерес представляет трехзначная логика Рейтинга $\mathbf{G}_3$ в силу её «родства» с интуиционистской логикой **H**. Исследованию функциональных свойств **G**3 (первая матрица Яськовского) посвящена монография М.Ф.Раца *[Раца* 1990].

Обратим внимание на следующий важный результат *[Раца* 1982]: *множество функций* $\mathbf{G}_3$ *включает континуум замкнутых классов со счетными базисами, а также континуум замкнутых классов, вообще не имеющих базиса.*

Поскольку $\mathbf{G}_3$ функционально вложима в $Ł_3$ (см. раздел 2.9), то таковыми же континуальными свойствами обладает и трехзначная логика Лукасевича $Ł_3$. Более того, Рац показал, что G_3 является предполным в T_3, т.е. G_3 предполно в $Ł_3$.

5.3. Функциональные свойства $Ł_n$ (Теорема В.К.Финна)

Здесь нам будет удобней работать со следующей формулировкой конечнозначных логик Лукасевича, которая эквивалентна исходной.

Под n+1-значной матрицей Лукасевича $\mathfrak{M}_{n+1}^{L}$ будем понимать матрицу следующего вида:

$\mathfrak{M}_{n+1}^{L} = \langle V_{n+1}, \sim, \to, \{n\}\rangle \quad (n \geq 2, n \in \mathbf{N})$, где

$V_{n+1} = \{0, 1, 2, \ldots, n\}$,

$\{n\}$ – множество выделенных значений.

Функции $\sim x$ и $x \to y$ определяются на множестве V_{n+1} следующим образом:

$\sim x = n\text{-}x$,

$x \to y = min(n, n\text{-}x+y)$.

Функции $x \vee y$ и $x \wedge y$ определяются через исходные:

$$x \vee y = (x \to y) \to y = max\ (x, y),$$

$$x \wedge y = \sim(\sim x \wedge \sim y) = min\ (x, y).$$

Множество всех суперпозиций функций $\sim x$ и $x \to y$ обозначим посредством $Ł_{n+1}$.

Ранее отмечалось, что множество функций $Ł_3$ является функционально предполным в P_3, т. е. добавление к $Ł_3$ какой-либо функции из P_3, не содержащейся в $Ł_3$, превращает $Ł_3$ в P_3. $Ł_3$ оказалось одним из предполных классов Яблонского, а именно тем, который сохраняет 0 и n-1 и обозначается посредством T_3. Таким образом, $Ł_3 = T_3$.

Возникает следующий нетривиальный вопрос: каковы функциональные свойства множества функций $Ł_{n+1}$ для любого n?

Ответ дан В.К.Финном в тезисах докладов *[Финн* 1970], а затем опубликовано подробное доказательство в совместной статье с Д.А.Бочваром *[Бочвар & Финн* 1972].

Пусть $I_{\xi\eta}(x)$ есть функции, определяемые следующим образом:

$$I_{\xi\eta}(x) = \begin{cases} \eta, если\ x = \xi \\ 0, если\ x \neq \xi \end{cases} \quad (0 < \xi, \eta < n).$$

Истинностными таблицами, отвечающими указанным функциям, будут таблицы вида

x	0	1	...	i	...	n-1	n
$I_{\xi\eta}(x)$	0	0	...	j	...	0	0,

где ξ=i, η=j, 1≤i, j≤n-1.

Обозначим посредством I_{n+1} множество всех $I_{\xi\eta}(x)$-функций, определимых в T_{n+1}.

Лемма 1. *Множество функций $Ł_{n+1}$ является функционально предполным в P_{n+1} т.т.т., когда все функции $I_{ij}(x)$, $1 \le i, j \le n-1$, принадлежат $Ł_{n+1}$. Причем, если $Ł_{n+1}$—предполное в P_{n+1} множество функций, то $Ł_{n+1} = T_{n+1}$ [Бочвар & Финн* 1972].

Заметим, что лемма 1 является обобщением результата В.К.Финна *[Финн* 1969] о функциональных свойствах $Ł_3$. Доказательство же самой леммы 1 есть по существу доказательство следующего утверждения: любая функция $f \in T_{n+1}$, которая не равна константе 0, определима посредством суперпозиции $x \vee y$, $x \wedge y$, I-функций и J-функций (эта суперпозиция есть аналог полной дизъюнктивной нормальной формы для двузначной логики).

Таким образом, ответственным за предполноту $Ł_{n+1}$ в P_{n+1} является наличие в $Ł_{n+1}$ множества функций I_{n+1}. Возникает вопрос: для каких n имеет место $I_{n+1} \subset Ł_{n+1}$, т. е. для каких n $Ł_{n+1} = T_{n+1}$?

Лемма 2. *$I_{n+1} \subset Ł_{n+1}$ т.т.т., когда n есть простое число [Бочвар & Финн* 1972].

Доказательство леммы 2 хотя и весьма громоздко, но является конструктивным, так как указан алгоритм построения суперпозиций исходных базисных функций $\sim x$, $x \to y$, равных соответствующим $I_{\xi\eta}(x)$-функциям. Отсюда, в частности, следует, что при $n = p$, где p - простое число, можно указать эффективный способ построения формулы, отвечающей функции $f \in Ł_{n+1}$, использующий I-функции и нормальные формы (I-J-с.д.н.ф.), рассмотренные при доказательстве леммы 1.

Из леммы 1 и леммы 2 получаем теорему 1, которая дает критерий функциональной предполноты множества функций $Ł_{n+1}$:

Теорема 1. *$Ł_{n+1} = T_{n+1}$ т.т.т., когда для любого $n \ge 2$ n есть простое число.*

Таким образом, множество функций в логике Лукасевича $Ł_{12}$ образует предполное множество, а множество функций в логике $Ł_{13}$ не является таковым.

В итоге мы имеем новое определение понятия простого числа: произвольное натуральное число $n \ge 2$ является простым т.т.т., когда множество всех функций $Ł_{n+1}$, соответствующее n+1-значным матричным логикам Лукасевича, есть функционально предполное множество в P_{n+1}, а именно $Ł_{n+1} = T_{n+1}$. Отсюда следует, что существует бесконечная *последовательность* p_s+1-значных логик Лукасевича *(p_s - s-е* в порядке возрастания простое число в натуральном ряду чисел), которым соответствует последовательность предполных множеств функций, такая, что $Ł_{p_s+1} = T_{p_s+1}$ для всех s=1,2,....

Заметим, что в *[Финн* 1976] устанавливается связь между понятиями степени (дедуктивной) полноты и функциональной предполноты:

Если $Ł_{n+1} = T_{n+1}$, где $n \ge 2$, то $\vec{\gamma}(Ł_{n+1}) = 3$.

5.3.1. Еще одно доказательство (А.Уркварт)

Ещё В.К.Финн обратил внимание на то, что теорема 1 может быть основана на критерии Мак-Нотона об определимости функций в $Ł_{n+1}$ (см. раздел 3.6), поскольку из теоремы Мак-Нотона следует

утверждение леммы 2. Действительно, $I_{\xi\eta}(x) \in Ł_{n+1}$ т.т.т., когда наибольший общий делитель чисел *n* и ξ делит нацело η.
На то, что в основу доказательства данной теоремы может быть положен критерий Мак-Нотона, обратил также внимание А.Уркварт *[Urquhart* 1986], который независимо от В.К.Финна дает доказательство теоремы 1. Это доказательство является самодостаточным, поэтому приведем его полностью.
Обозначим посредством НОД(x,y) наибольший общий делитель x и y; для конечного множества $X=\{x_1,\ldots,x_m\}$ обозначим посредством НОД(X) наибольший общий делитель $x_1,\ldots,x_m$.
Ключевой леммой для характеризации $Ł_{n+1}$-определимых функций является:
Лемма 3. *Пусть F - множество функций на* $\{0,\ldots,n\}$, *содержащее все* $Ł_{n+1}$-*определимые функции. Если* $X \subseteq \{0,\ldots,n\}$ *является F-замкнутым, то* $НОД(x, y) \in X$ *для всех* $x, y \in X$.
ДОКАЗАТЕЛЬСТВО.
Для всех $x,y \in X$, если $x>y$, то $x-y \in X$. Повторяя вычитание, получаем, что остаток от деления x на y также содержится в X. Алгоритм Эвклида для нахождения $НОД(x,y)$ работает путем повторного вычисления остатков. Таким образом получаем $НОД(x, y) \in X$.
Теорема 2 (Критерий Мак-Нотона). *m-местная функция f, определенная на* $M=\{0,\ldots,n\}$, *является* $Ł_{n+1}$-определимой *m.m.m., когда для всякой m-ки* $a=\langle a_1,\ldots,a_m\rangle$ $f(a)$ *делится на* $НОД(\{a_1,\ldots,a_m,n\})$.
ДОКАЗАТЕЛЬСТВО.
Необходимое условие следует из того факта, что если *x* и *y* делятся на К, то $\sim x$ и $x \to y$ также делится на К.
Для доказательства достаточности мы должны охарактеризовать лишь $Ł_{n+1}$-замкнутые множества. Пусть X_k будет множеством чисел из М, кратных *k.* Покажем, что $Ł_{n+1}$-замкнутые множества - это в точности множества X_k для *k*, являющегося делителем *n.* И наоборот, если X является $Ł_{n+1}$-замкнутым множеством, пусть $k = НОД(X)$. По лемме 3 $k \in X$. Так как *n* принадлежит каждому $Ł_{n+1}$-замкнутому множеству, *k* является делителем *n, n = qk* для некоторого *q.* Пусть теперь *y=pk* будет числом из М, кратным *k.* Тогда $y = n-(q-p)k$ и, следовательно, $y \in X$, показывая тем самым, что $X = X_k$.
Утверждаемая характеризация $Ł_{n+1}$-определимых функций теперь легко следует из описания $Ł_{n+1}$-замкнутых множеств.
Теорема 2 в действительности дает нам больше информации, чем мы утверждали, а именно характеризацию всех функций, определимых с

помощью функций, включающих множество Лукасевича. Пусть X - подмножество делителей *n.* Будем говорить, что X является НОК-замкнутым, если (1) $1 \in X$ и (2) если $x, y \in X$, то $НОК(x,y)$, наименьшее общее кратное *x* и *y,* принадлежит X. Далее, для $Y \subseteq \{0,\ldots,n\}$, $Y \neq \varnothing$ пусть F(Y) будет множеством функций на $\{0,\ldots,n\}$, удовлетворяющих условию: для всякого $k \in Y$, $a_1,\ldots,a_m \in X$ влечет $f(a) \in X_k$. Легко видеть, что F(Y) замкнуто относительно композиции функций.
Следствие 1. *Множества n+1-значных функций, которые замкнуты относительно композиции и включают* $Ł_{n+1}$-*определимые функции, являются в точности множествами F(Y), где Y есть НОК-замкнутое подмножество* $\{0,\ldots,n\}$.
ДОКАЗАТЕЛЬСТВО.
По теореме 2 замыканиями таких множеств функций должны быть множества вида X_k для некоторого *k,* являющегося делителем *n.* Множество $\{k|X_k$ − F-замкнутое множество$\}$ является НОК-замкнутым. Остается лишь проверить, что каждое НОК-замкнутое подмножество множества М является F-замкнутым множеством некоторого множества функций F. Соответственно пусть Y будет НОК-замкнутым подмножеством множества М. Мы хотим показать, что множества X_k для каждого $k \in Y$ являются в точности *F(Y)*-замкнутыми множествами. Пусть X будет *F(Y)*-замкнутым подмножеством множества М. По теореме 2, $X = X_k$ для *k,* являющегося некоторым делителем *n.* Пусть $p = НОК(\{q|X \subseteq X_q, q \in Y\})$. Пусть $X \subseteq X_p$; чтобы показать $X = X_p$, достаточно показать, что $p \in X$. Определим функцию *f* посредством $f(k) = p$, $f(a) = 0$ для $a \neq k$. Теперь если $k \in X_q$ для $q \in Y$, то $X = X_k \subseteq X_q$ и, следовательно, $p \in X_q$, так как X_p является наименьшим множеством X_r, $r \in Y$, содержащим X. Отсюда следует, что $f \in F(Y)$. Таким образом, $p \in X$. Доказательство завершено.
Следствие 2. *Множество функций* $Ł_{n+1}$ *образует предполное множество m.m.m., когда n - простое число.*
ДОКАЗАТЕЛЬСТВО.
Единственными НОК-замкнутыми подмножествами делителей для простого *n* являются {1, *n}* и {1}. Если *n* не является простым числом, то могут быть и другие.
Замечание. Между доказательствами В.К.Финна и А.Уркварта было опубликовано еще одно доказательство о функциональных свойствах $Ł_{n+1}$, отличное как от первого, так и от второго. Это доказательство

принадлежит Г.Хендри *[Hendry* 1983], и здесь функциональная предполнота называется *минимальной неполнотой.*

6. Структурализация простых чисел

Представлен алгоритм, перерабатывающий любую логику Лукасевича $Ł_{n+1}$ *в функционально предполную. Таким образом, натуральный ряд превращается в своеобразную последовательность простых чисел. Это, в свою очередь, приводит к разбиению натурального ряда чисел на классы эквивалентности, такие, что в каждом классе находится одно и только одно простое число. Отсюда приходим к структурализации простых чисел в виде корневых деревьев, примеры которых приведены в тексте. В основе построения деревьев лежит обратная функция Эйлера* $\varphi^{-1}(m)$. *Для вычисления значений этой функции В.И.Шалаком разработана компьютерная программа, которая в совокупности с другими строит сами деревья. Поскольку каждое корневое дерево представимо в виде р-абелевой группы, то оказывается, что эта структура является фундаментальной для теории чисел.*

6.1. Разбиение множества логик Лукасевича $Ł_{n+1}$ на классы эквивалентности относительно свойства предполноты

Вернемся к результату теоремы В.К.Финна, в которой дается критерий функциональной предполноты для множества функций $Ł_{n+1}$, соответствующего конечнозначным логикам Лукасевича $Ł_{n+1}$. Этот критерий состоит в том, что $Ł_{n+1}$ предполно т.т.т., когда *n* есть простое число. С другой стороны, была построена матричная логика T_{n+1}, множество функций T_{n+1} которой предполно для любого $n \geq 2$. Очевидно, что $Ł_{n+1} = T_{n+1}$ только для случая, когда *n* есть простое число. Возникает следующий вопрос: нельзя ли построить такую последовательность *n*+1-значных логик, чтобы они были предполны для любого $n \geq 2$, но при этом сохраняли свойства логик Лукасевича? Казалось бы, это противоречит результату теоремы 1 о критерии функциональной предполноты для множества функций $Ł_{n+1}$, но тем не менее такую последовательность логик можно построить и вот в каком смысле.

Из всех трех доказательств теоремы 1 в гл. 5 (В. К. Финна, Г. Хендри и А. Уркварта) только в первом случае *явно* указано, что несет ответственность за непредполноту в $Ł_{n+1}$ в случае, когда $n \neq p$, где *p* - простое число, а именно в этом случае в $Ł_{n+1}$ определимы не все $I_{\xi\eta}(x)$-функции:

$$I_{\xi\eta}(x) = \begin{cases} \eta, если\ x = \xi \\ 0, если\ x \neq \xi \end{cases} \quad (0 < \xi, \eta < n).$$

Например, пусть *n* + 1 = 10, и для удобства обозначим множество истинностных значений V_{10}, как у Лукасевича, т.е.

$$V_{10} = \{0, {}^1/_9, {}^2/_9, {}^3/_9, {}^4/_9, {}^5/_9, {}^6/_9, {}^7/_9, {}^8/_9, 1\}.$$

Из критерия Мак-Нотона следует, что в $Ł_{10}$ нельзя, например, определить функцию $I_{\frac{3}{9}\frac{1}{9}}(x)$, когда $x = {}^3/_9$, поскольку НОД(3, 9) =3, а НОД(3, 7) = 1. Уже в силу этого $Ł_{10}$ непредполна. Очевидно, если числитель и знаменатель для всех *1/n* из *V* являются взаимно-простыми числами, т.е. НОД(/, *n)* = 1, то определимость *I*-функций в $Ł_{n+1}$ сохраняется. Отсюда следует, что ответственными за непредполноту $Ł_{10}$ являются значения *3/9* и *6/9,* числитель и знаменатель которых не являются взаимно-простыми числами. Если из F_{10} мы вычеркнем эти значения, то останется восемь значений. Пусть *n* = 8, тогда

$$V_8 = \{0, {}^1/_7, {}^2/_7, {}^3/_7, {}^4/_7, {}^5/_7, {}^6/_7, 1\}.$$

Таким образом, от $Ł_{10}$ мы перешли к $Ł_8$, а в силу теоремы 1 $Ł_8$ является предполной логикой. В общем случае, исходя из определения I-функций, рассматриваются только промежуточные истинностные значения и вычеркиваются все значения, в которых НОД(*i*/,*n*)≠1. Итак, для того чтобы перестроить некоторое произвольное множество истинностных значений *V,* надо определить число чисел *i* из ряда 1, 2, ..., *n-1,* взаимно-простых с *n*, и прибавить два крайних истинностных значения (0 и 1). Функция, которая определяется для всех целых положительных *n* и представляет собой число чисел, не превосходящих *n* и взаимно-простых с *n,* называется *функцией Эйлера* (Л.Эйлер: 1707-1783), которая, согласно К.Гауссу (1777-1855), обозначается знаком φ(*n*). Так, в начале 80-х годов XX века была переоткрыта функция Эйлера.

(Функция φ(*n*) была введена Л.Эйлером (1707-1783) в одной из работ, опубликованной в 1763 г. Иногда функция Эйлера называется *totient function.* В «Англо-русском словаре математических терминов» (отв. ред. П. С Александров / М: Мир, 1994) термин «totient» переводится как *(φ-функция Эйлера, тотиент-функция.)*

Посредством функции φ(*n*) из множества истинностных значений V_{n+1} получаем множество $V_{\varphi(n)+2}$. Тогда построение предполной логики $Ł_{p+1}$ из произвольной $Ł_{n+1}$ сводится к переработке произвольного числа *n* в *p,* где *p* - простое число. Таким образом, каждый раз *n* из V_{n+1} перерабатывается в $\varphi(n)+1$. Итак, мы установили, что в основе операции «вычеркивания» промежуточных значений, в которых $(i,\ n)\ \neq 1,$ лежит теоретико-числовая функция Эйлера φ(*n*).

Примеры:

$\varphi(1) = 1,$	$\varphi(5) = 4,$	$\varphi(9) = 6,$
$\varphi(2) = 1,$	$\varphi(6) = 2,$	$\varphi(10) = 4,$
$\varphi(3) = 2,$	$\varphi(7) = 6,$	$\varphi(11) = 10,$
$\varphi(4) = 2,$	$\varphi(8) = 4,$	$\varphi(12) = 4.$

Функция Эйлера φ(*n*) имеет следующие хорошо известные свойства, которые нам понадобятся в дальнейшем *[Виноградов* 1981].

(*i*) $\varphi(p) = p-1.$

(*ii*) φ(*n*) является мультипликативной, т.е. если $(n_1,\ n_2) = 1,$ то $\varphi(n_1 \cdot n_2) = \varphi(n_1) \cdot \varphi(n_2).$

(*iii*) Для любого простого числа *p,* $\varphi(p^\beta) = p^{\beta-1}(p-1).$ Из (1)и (2) следует, что

$$\varphi(n) = n\left(1-\frac{1}{p_1}\right)\cdot\left(1-\frac{1}{p_2}\right)\ldots\left(1-\frac{1}{p_r}\right) = n\prod_{p|n}\left(1-\frac{1}{p}\right),$$

где знак $p|n$ означает, что множители произведения берутся при всех возможных простых делителях числа *n.*

(iv) Значение φ(*n*) является четным числом для любого $n \geq 3$ [2]

(Имеется интернетовский сайт, посвященный свойствам функции Эйлера φ(*n*) , с обширной библиографией (см. *[Totient Function* 2000]]). См. также *[Maier & Pomerance* 1988] и *[Spyroponlus* 1989]. С алгебраической точки зрения целые числа, не превышающие данного числа и взаимно простые с ним, образуют группу, эта группа является циклической т.т.т., когда *n* = 2, 4, или *n* представимо в форме p^κ или $2p^\kappa$ *[Абрамович & Стиган* (ред.) 1979].)

Пусть $\varphi^*(n) = \varphi(n)+1.$ Тогда, если *n* =*p,* то $\varphi^*(n) = (p-1)+1 = p.$ Отсюда, если $Ł_{n+1}$ предполна, т.е. если и = *p,* то функция $\varphi^*(n)$ не изменяет предполноту $Ł_{p+1}.$

Заметим, что для приведенных выше примеров значений функции φ(*n*) значение $\varphi^*(n) = p,$ но это имеет место не для всех *n.*

Например, φ*(16) = 9, а $Ł_{9+1}$ не является предполной логикой, поскольку *9≠p.* Однако, как мы уже знаем, φ*(9) = 7 и, следовательно, $Ł_{7+1}$ предполна. Тогда посредством $\varphi_k^*(n)$ обозначим *k*-е применение функции φ*(*n*). В частности, если *n* = *p,* то для любого *k* $\varphi_k^*(n) = p.$ Очевидно, для любого *n,* число *k* конечно, т.е. *для всякого n существует k такое, что* $\varphi_k^*(n)$ *есть простое число*[3].

(Для любого $n \leq 100000$ значение $\varphi_k^*(n)$ можно вычислить, используя таблицу значений φ(*n*) в *[Lai & Gillard* 1968]. Предыдущие две хорошо известные таблицы значений φ(*n*) принадлежат, соответственно, Дж.Сильвестру (для $n \leq 1000),$ изданной в 1883 г., и Дж. Глэйшеру *[Glaisher* 1940] (для $n \leq 10000$). Заметим, что при издании тех или иных таблиц значений функций обычно преследуются опеределенные цели. Так, оба предыдущих автора заинтересованы не столько в φ(*n*) как таковой, но скорее в $\Sigma\varphi(n)$ и в обратной функции от φ(*n*). В настоящем случае интерес, кажется, лежит в нахождении решений уравнений вида $\varphi(n) = \varphi(n+1)$ и других подобных уравнений. В свою очередь, издание таблицы значений φ(*n*) в *[Абрамович & Стиган* 1979, таблица 24.6] (для $n \leq 1000$) предлагает также таблицу первообразных корней. Имеется $\varphi(\varphi(n))$ первообразных корней числа *n.* Нас же, как будет видно из следующего раздела, будет интересовать именно функция обратная к φ(*n*) . Таблица значений этой фунции для $m \leq 5000$ (см. Таблицу 2) позволяет построить графы в виде корневых деревьев для первых 52 простых чисел.)

Таким образом, мы имеем метод, а на самом деле *алгоритм,* по которому любое натуральное число *n* перерабатывается в простое *p,* а следовательно, логику $Ł_{n+1}$ в логику $Ł_{p+1}$:

0. Пусть $n = n_1$ и $n_1 \neq p_t.$

1. $\varphi_1^*(n_1) = p_t,$ *или* $\varphi_1^*(n_1) = n_2,$ где $n_2 < n_1.$

2. $\varphi_2^*(n_2) = p_t,$ или $\varphi_2^*(n_2) = n_3,$ где $n_3 < n_2.$

.

.

.

k. $\varphi_k^*(n_k) = p_t.$

Итак, любая логика $Ł_{n+1}$ перерабатывается в предполную логику $Ł_{p+1}$ посредством указанной схемы алгоритма. Например, логика $Ł_{138+1}$

перерабатывается в логику $Ł_{13+1}$, при этом $k = 4$. В итоге, функция $\varphi_k^*(n)$ порождает на натуральном ряду N бесконечную последовательность простых чисел *[Карпенко* 1983]:

2, 2, 3, 3, 5, 3, 7, 5, 7, 5, 11, 5, 13, 7, 7, 7, 17,
7, 19, 7, 13, 11, 23, 7, 13, 13, 19, 13, 29, 7, 31,
…

В этом и заключается ответ на поставленный в начале этого раздела вопрос, о построении такой последовательности n+1-значных логик, чтобы они были предполны для любого $n \geq 2$, но при этом сохраняли свойства логик Лукасевича, т.е. построен алгоритм, при котором любая логика Лукасевича $Ł_{n+1}$ перерабатывается в предполную логику вида $Ł_{\varphi_k^*(n)+1}$.

А теперь отметим, что в августе 2000 г. Карпенко была обнаружена точно такая же последовательность чисел в интернетовском сайте Н.Слоана [Sloane 1999]. (Этот сайт есть расширенный вариант «Энциклопедии числовых последовательностей» *[Sloane & Plouffe* 1995].) Но как раз главный интерес представляет дальнейшее развитие этого результата.

Из существования указанного алгоритма следует, что функция $\varphi_k^*(n_k)$ индуцирует разбиение множества логик $Ł_{n+1}$ на классы эквивалентности по отношению $\cong$, т.е. $Ł_{n_1+1} \cong Ł_{n_2+1}$ т.т.т., когда $\exists k \exists l$ такие, что $\varphi_k^*(n_1) = \varphi_l^*(n_2)$. Отсюда, в каждом классе эквивалентности содержится одна и только одна предполная логика $Ł_{p_s+1}$, где p_s - s-е по порядку простое число. Сами классы будем обозначать посредством $\mathcal{X}_{p_s+1}$, например, $\mathcal{X}_{p_3+1} = \{6, 9, 11, 13\}$, где $p_3 = 5$.

6.2. Построение классов $\mathcal{X}_{p_s+1}$ (обратная функция Эйлера)

В связи с данным разбиением логик Лукасевича $Ł_{n+1}$ встает задача построения по произвольной предполной логике $Ł_{p+1}$ соответствующего ей класса $\mathcal{X}_{p+1}$. Для этого нужно определить функцию, обратную функции $\varphi_k^*(n)$. Это можно сделать, зная множество значений обратной функции Эйлера $\varphi^{-1}(m)$, которая

определяется отношением $\varphi^{-1}(m) = \{n: \varphi(n) = m\}$. Например, если $\varphi(n) = 4$, то это уравнение имеет ровно четыре решения, а именно $\varphi^{-1}(4)=\{5,8, 10, 12\}$.

Наверное, впервые на проблему решения подобных уравнений обратил внимание Э.Люка (E.Lucas: 1842-1891). По крайней мере, в книге И.В.Арнольда *[Арнольд* 1939] читаем: «следуя Люка сгруппированы числа n с одним и тем же значением функции $\varphi(n)$ в пределах для $\varphi(n)$ от 1 до 100, то есть дана таблица функции обратной по отношению $\varphi(n)$» (Таблица 4). В следующем году появилась таблица значений функции $\varphi^{-1}(m)$ для $m \leq 2500$ *[Glaisher* 1940].

Интересно, что в монографии *[Bolker* 1970] поставлена проблема найти все решения уравнения $\varphi(n)= 24$. Решение этой проблемы приводится в книге *[Burton* 1976]:

$$\varphi^{-1}(24) = \{35, 39, 45, 52, 56, 70, 72, 84, 90\}$$

(В книге *[Серпинский* 1968] предлагается следующая задача под № 245: «Найти все натуральные числа $n \leq 30$, для которых $\varphi(n) = d(n)$, где $\varphi(n)$- функция Эйлера, а $d(n)$ - число натуральных делителей числа n». Рассмотрим только случай $n = 30$. Делителями числа 30 являются числа 1, 2, 3, 5, 6, 10, 15 и 30, т.е. d(30) = 8. Значит, надо решить уравнение $\varphi(n)= 8$, где $n \leq 30$. Или, по-другому, найти значения для обратной функции Эйлера $\varphi^{-1}(8)$, т.е. определить множество $\{n: \varphi(n) = 8\}$ для $n \leq 30$. Это множество состоит из чисел (15, 16, 20, 24, 30}. И более того (см. ниже), ни для каких других $n > 30$ $\varphi(n) \neq 8$.)

Заметим, что ни в одной из этих книг по теории чисел обозначение $\varphi^{-1}(m)$ не встречается. Только в 1981 г. появилась статья Х.Гупты, специально посвященная свойствам обратной функции Эйлера $\varphi^{-1}(m)$ *[Gupta* 1981].

(Отметим, что в этой работе дается очень простая, не встречающаяся ранее в литературе, формула сведения (reduction formula) для $\varphi(n)$. Имеем $\varphi(1) = 1$; для $n \geq 2$ мы можем написать $n = pu$, где p есть простой делитель n, и u есть некоторое целое число ≥ 1. Тогда легко видеть, что в зависимости от того, делит p или не делит u, $\varphi(n) = p\varphi(u)$ или $(p-1)\varphi(u)$.)

Множество значений $\varphi^{-1}(m)$ пусто для всех нечетных и многих четных значений $m > 1$. Например, из таблицы значений $\varphi^{-1}(m)$ для $n \leq 100$, приведенной в *[Арнольд* 1939], следует, что числа 14, 26, 34, 38, 50, 62, 68, 74, 76, 86, 90, 94 и 98 не являются значениями $\varphi(n)$. Более того, из результата *[Rechman* 1977] следует, что количество четных чисел, не являющихся значениями $\varphi(n)$, *бесконечно.* Всё это имеет

существенное значение для построения классов $\mathcal{X}_{p,+1}$. Заметим также, что для любого (p-1) больше 2, $\varphi^{-1}(p-1)$ содержит, по крайней мере, элемент $2p$, поскольку в силу мультипликативности φ(n) имеем $\varphi(2p) = \varphi(2)\cdot\varphi(p) = 1\cdot(p-1) = p-1$. Отсюда, в каждом классе $\mathcal{X}_{p+1}$, кроме исходной логики $\mathcal{L}_{p+1}$, имеется обязательно логика $\mathcal{L}_{2p+1}$.

В *[Gupta* 1981] определяются нижняя и верхняя границы для любого непустого множества $\varphi^{-1}(m)$. Пусть n - любой элемент $\varphi^{-1}(m)$. Тогда, очевидно, $m \le n$, что и является нижней границей.

Используя формулу

$$n\prod_{p|n}\left(1-\frac{1}{p}\right)$$

для вычисления φ(n), получим верхнюю границу, а именно никакое n, для которого φ(n)= m, не превосходит число $U(m)$, где

$$U(m) = m\prod_{(p-1)|m} p/(p-1).$$

Дается также способ улучшения верхней оценки.

Таким образом, непустое множество $\varphi^{-1}(m)$ всегда конечно.

Вообще-то говоря, это следует уже из того, что число делителей m конечно.

(Отметим также результат из статьи *[Pomerance* 1980], где установлено, что множество значений $\varphi^{-1}(m) > m^c$ для бесконечного множества чисел $m \in \mathbf{N}$, если $c \cong 0.55092$.)

Для нас важно, однако, что в *[Gupta* 1981] предлагается способ построения множества $\varphi^{-1}(m)$ по любому m, являющемуся значением функции φ(n). Это делается следующим образом.

Пусть n есть элемент $\varphi^{-1}(m)$ для данного m. Допустим, что p есть наименьший делитель n. Пусть $n = p^{\beta}\cdot u$, где $(u, p) = 1$. Отсюда следует, что u не имеет простого делителя $\le p$. Очевидно, что

$$m = \varphi(n) = \varphi(p^{\beta})\cdot\varphi(u).$$

Для того чтобы это имело место, необходимо, чтобы наше p являлось таким, что $(p-1)|m$ и u принадлежало к такому подмножеству множества $\varphi^{-1}(m)/\varphi(p^{\beta})$, которое состоит из тех его элементов, которые не имеют простых делителей $\le p$. Такое подмножество обозначим посредством $\varphi_p^{-1}(m/\varphi(p^{\beta}))$. Тогда понятно, что каждый элемент из $p^{\beta}\varphi_p^{-1}(m/\varphi(p^{\beta}))$ является решением уравнения $\varphi(x) = m$. При этом p пробегает по всем тем простым числам, которые удовлетворяют условию $(p-1)|m$, а β — по всем тем значениям, для которых $\varphi(p^{\beta})$ делит m. Значения β , для

которых $m/\varphi(p^{\beta})$ является нечетным числом > 1, выбрасываются.

ПРИМЕР. Пусть m = 36. Чтобы получить простые числа p, для которых $(p-1)|m$, выпишем все делители m; затем, добавив 1 к каждому из них, оставим только простые числа. Итак, $36 = 2^2 \cdot 3^2$, поэтому делителями 36 являются числа

1, 2, 4; 3, 6, 12; 9, 18, 36.

Добавив 1, получаем

2, 3, 5; 4, 7, 13; 10, 19, 37.

Простые числа из этого списка расположим в обратном порядке:

37, 19, 13, 7, 5, 3, 2.

Будем считать, что мы уже располагаем значениями функции $\varphi^{-1}(x)$ для всех x < 36. Нам понадобятся следующие множества:

x	$\varphi^{-1}(x)$
1	{1, 2}
2	{4, 6}
6	{7, 9, 14, 18}
18	{19, 27, 38, 54}

Тогда имеем следующую таблицу вычислений:

p	β	$m/\varphi(p^{\beta})$	$p^{\beta}\varphi_p^{-1}(m/\varphi(p^{\beta}))$
37	1	1	37 {1} = {37}
19	1	2	19. ∅
13	1	3	–
7	1	6	7. ∅
5	1	9	–
3	1	18	3 {19} = {57}
3	2	6	9 {7} = {63}

Для следующего шага нам потребуются все нечетные элементы из $\varphi^{-1}(36)$, которые мы уже вычислили. Этим объясняется, почему мы расположили простые числа в обратном порядке:

p	β	$m/\varphi(p^\beta)$	$p^\beta \varphi_p^{-1}(m/\varphi(p^\beta))$
2	1	36	2 {37, 57, 63} = = {74, 114, 126}
2	2	18	4 {19, 27} = {76, 108}
2	3	9	–

В итоге, $\varphi^{-1}(36) = \{37, 57, 63, 74, 76, 108, 114, 126\}$.

Стоит сказать, что рассмотренный метод построения множества $\varphi^{-1}(m)$ является довольно-таки громоздким, тем более, что для построения $\varphi^{-1}(m)$ нужно знать $\varphi^{-1}(x)$ для всех $x < m$, а также, в общем случае, разложение элементов из $\varphi^{-1}(x)$ на простые множители. Но зато этот метод является *эффективным.* В свою очередь, мы можем предложить эвристическое применение этого метода.

Пусть для сравнения опять $m = 36$. Выпишем все делители числа 36: 1, 2, 3, 4, 6, 9, 18, 36. Теперь рассмотрим различные представления числа 36 произведениями делителей, *причем такими, которые являются значениями функции* $\varphi(n)$. Будем идти слева направо, например, $2 \cdot 18 = 36$. Поскольку $\varphi(3) = 2$, $\varphi(19) = 18$, а $(3, 19) = 1$, то $\varphi(3) \cdot (19) = \varphi(3 \cdot 19) = \varphi(57)$. Отсюда $n = 57$. Так как n нечетно, то значениями $\varphi^{-1}(36)$ будет также $2 \cdot n = 114$. В результе имеем:

$m = 36 = \varphi(37)$, $n = 37$;

$m = 36 = 1 \cdot 36 = \varphi(2) \cdot \varphi(37) = \varphi(2 \cdot 37) = \varphi(74)$, $n = 74$;

$m = 36 = 2 \cdot 18 = \varphi(3) \cdot \varphi(19) = \varphi(3 \cdot 19) = \varphi(57)$, $n = 57$;

$m = 36 = 1 \cdot 2 \cdot 18 = \varphi(2) \cdot \varphi(3) \cdot \varphi(19) = \varphi(2 \cdot 3 \cdot 19) = \varphi(114)$, $n = 114$;

$m = 36 = 2 \cdot 1 \cdot 18 = \varphi(2^2) \cdot \varphi(19) = \varphi(2^2 \cdot 19) = \varphi(76)$, $n = 76$;

$m = 36 = 2 \cdot 1 \cdot 3^2 \cdot 2 = \varphi(2^2) \cdot \varphi(3^3) = \varphi(2^2 \cdot 3^3) = \varphi(108)$, $n = 108$;

$m = 36 = 3 \cdot 2 \cdot 6 = \varphi(3^2) \cdot \varphi(7) = \varphi(3^2 \cdot 7) = \varphi(63)$, $n = 63$;

$m = 36 = 1 \cdot 3 \cdot 2 \cdot 6 = \varphi(2) \cdot \varphi(3^2) \cdot \varphi(7) = \varphi(2 \cdot 3^2 \cdot 7) = \varphi(126)$, $n = 126$;

Наличие эффективного метода построения множества значений $\varphi^{-1}(m)$ позволяет в принципе построить алгоритм, который по любой предполной логике $Ł_{p+1}$ строит её класс эквивалентности $\mathcal{X}_{p+1}$. Идея алгоритма состоит в следующем.

Рассмотрим функцию, обратную $\varphi^*(n)$, т.е. функцию $\varphi^{-1^*}(m)$. Пусть $m = p$, *где* p - простое число.

0. Из p вычитаем 1, т.е. имеем $p-1$.
1. Находим множество значений $\varphi^{-1}(p-1)$, т е. находим множество $\{n: \varphi(n) = p-1\}$. Это множество может состоять из двух классов: $\{v_o\}_1$ и $\{v_e\}_1$, где $\{v_o\}_1$ - класс нечетных значений, не содержащих данное p, а $\{v_e\}_1$-класс четных значений. Класс $\{v_e\}_1$ каждый раз из дальнейших рассуждений отбрасывается, поскольку, $v_e - 1$, как нечетное число, не может быть значением функции Эйлера $\varphi(n)$. Если класс $\{v_o\}_1$ пуст, как, например, в случае $\varphi^{-1^*}(3)$ или $\varphi^{-1^*}(5)$, то класс эквивалентности $\mathcal{X}_{p+1}$ построен. Если же $\{v_o\}_1$ непусто, то находим множество значений $\varphi^{-1}(v_o - 1)$ для каждого v_0 из класса $\{v_o\}_1$. Имеем два подслучая.
2. (a) $\{v_o\}_2 = \varnothing$ или (b) $\{v_o\}_2 \neq \varnothing$. В первом случае процес построения $\mathcal{X}_{p+1}$ закончен. Если же $\{v_o\}_2 \neq \varnothing$, то все повторяется. Имеем два подслучая.
3. (a) $\{v_o\}_3 = \varnothing$ или (b) $\{v_o\}_3 \neq \varnothing$.

.
.
.

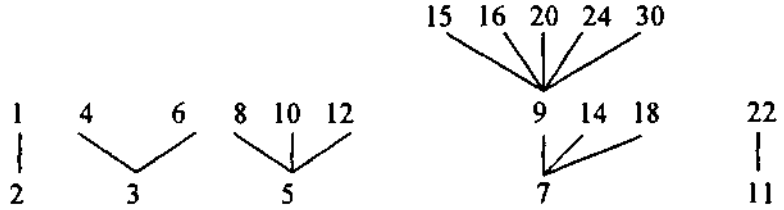
?

Для примера возьмем предполную логику $\mathcal{X}_{37+1}$. Как ранее было установлено, $\varphi^{-1}(36) = \{37, 57, 63, 74, 76, 108, 114, 126\}$, т е. $\varphi^{-1^*}(37) = \{57, 63\} \cup \{74, 76, 108, 114, 126\}$, где $\{v_o\}_1 = \{57, 63\}$ и $\{v_e\}_1 = \{74, 76, 108, 114, 126\}$. Рассмотрим $\varphi^{-1^*}(57)$ и $\varphi^{-1^*}(63)$.

Поскольку множество значений $\varphi^{-1^*}(63)$ пусто, так как число 62 не является значением функции $\varphi(n)$, то остается $\varphi^{-1^*}(57)$ Находим,что $\varphi^{-1^*}(57) = \{87\} \cup \{116, 174\}$, где $\{v_o\}_2 = \{87\}$ и $\{v_e\}_2 = \{116, 174\}$. Поскольку $\varphi^{-1^*}(87) = \varnothing$, то на этом построение класса $\mathcal{X}_{37+1}$ закончено. Добавив к каждому элементу построенного множества по 1, получим класс эквивалентности логик Лукасевича для предполной логики $\mathcal{X}_{37+1} = \{57, 63, 74, 76, 108, 114, 126, 87, 116, 174\}$.

6.3. Графы для простых чисел

Перейдем от классов эквивалентности логик Лукасевича X_{p+1} к классам эквивалентности X_p, что приводит к разбиению натурального ряда чисел на классы эквивалентности, такие, что в каждом классе содержится одно и только одно простое число. Указанный метод, просредством которого производится данное разбиение, дает способ представления каждого простого числа в виде связного, не содержащего циклов, графа с одной выделенной вершиной, т.е. в виде *корневого дерева,* которое обозначим посредством T_p. Корнем дерева T_p является само *p,* а множеством вершин - множество элементов X_p. В итоге от логик Лукасевича мы переходим непосредственно к самим простым числам. Приведем графы для первых 5 простых чисел



Дальше будем представлять деревья, как их принято сохранять в текстовых файлах. Для сравнения, приведенные выше пять деревьев выглядят следующим образом:


```
2
        1
3
        4
        6
5
        8
        10
        12
7
        9
                15
                16
                20
                24
                30
        14
        18
11
        22
```

В *[Карпенко* 1983] приведены корневые деревья для первых 13 простых чисел (см. также *[Карпенко* 1989]). Здесь доведем этот список до первых 25 простых чисел (это как раз простые числа, входящие в первую сотню натурального ряда чисел):

23
46
29
58
31
62
37
57
87
116
174
63
74
76
108
114
126
41
55
81
123
164
165
249
332
498
176
200
220
246
264
300
330
162
75
82
88
100
110
132
150

43
49
65
85
129
255
256
272
320
340
384
408
480
510
147
172
196
258
294
128
136
160
170
192
204
240
104
105
159
212
318
112
130
140
144
156
168
180
210
86
98
47
94

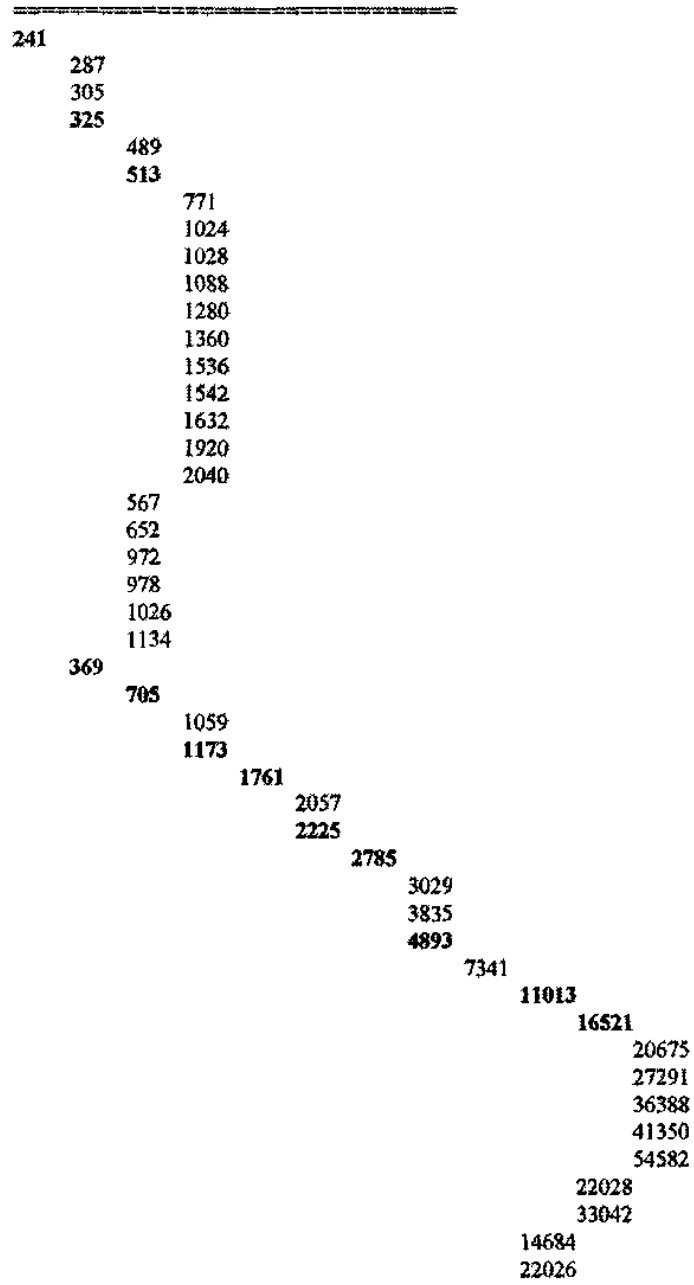
53
106
59
118
61
77
93
141
213
321
425
535
856
1070
1284
561
725
843
957
1437
2157
2876
4314
1916
2874
1065
1124
1136
1276
1420
1450
1686
1704
1914
2130
615
656
704
748

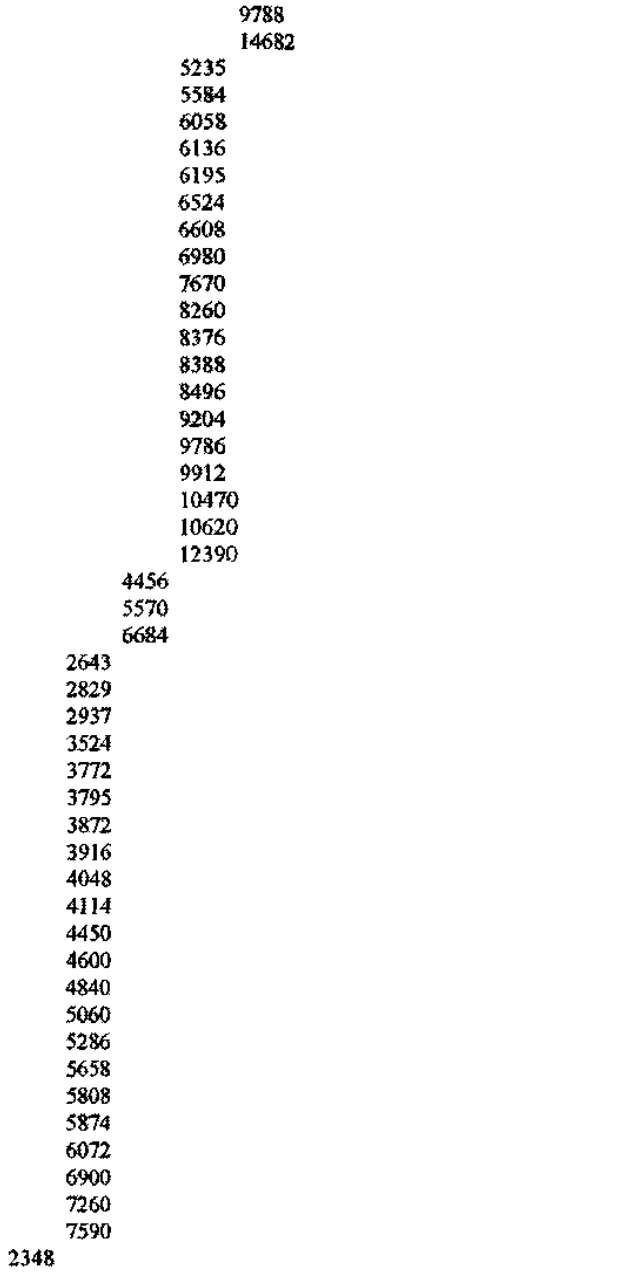
800
820
850
880
984
1056
1122
1200
1230
1320
428
642
284
426
188
282
99
122
124
154
186
198
67
134
71
142
73
91
95
111
121
143
155
175
183
225
339
435
452
464
580

678
696
870
231
244
248
286
308
310
350
366
372
396
450
462
242
117
177
267
345
519
692
1038
356
368
460
534
552
690
236
354
135
146
148
152
182
190
216
222
228
234
252
270

79
158
83
166
89
115
178
184
230
276
97
119
153
194
195
208
224
238
260
280
288
306
312
336
360
390
420

В *[Карпенко* 1983] при использовании таблицы значений $\varphi^{-1}(m)$ для $m \le 2500$ (см. *[Glaisher* 1940]) были построены деревья для первых 42 простых чисел, а затем для первых 50 простых чисел *[Карпенко* 1989]. Далее появляется простое число 241 (№ 53), для которого корневое дерево построено с помощью компьютерной программы.


241
287
305
325
489
513
771
1024
1028
1088
1280
1360
1536
1542
1632
1920
2040
567
652
972
978
1026
1134
369
705
1059
1173
1761
2057
2225
2785
3029
3835
4893
7341
11013
16521
20675
27291
36388
41350
54582
22028
33042
14684
22026


9788
14682
5235
5584
6058
6136
6195
6524
6608
6980
7670
8260
8376
8388
8496
9204
9786
9912
10470
10620
12390
4456
5570
6684
2643
2829
2937
3524
3772
3795
3872
3916
4048
4114
4450
4600
4840
5060
5286
5658
5808
5874
6072
6900
7260
7590
2348

3522
1335
1412
1424
1472
1564
1780
1840
2118
2136
2208
2346
2670
2760
752
940
1128
1410
385
485
579
595
663
765
1149
1532
2298
772
776
832
884
896
952
970
1040
1120
1152
1158
1164
1190
1224
1248
1326
1344
1428
1440
1530

1560
1680
429
465
699
885
1329
2505
2672
3340
4008
5010
1772
2658
932
944
1180
1398
1416
1770
482
488
495
496
525
789
1052
1578
572
574
610
616
620
650
700
732
738
744
770
792
858
900
924
930
990
1050

Компьютерная программа для построения корневых деревьев, представляющих простые числа, была создана в начале 1995 г. В.И.Шалаком. Программа состоит из трех подпрограмм:
(1) ERATOS - порождение простых чисел методом решета Эратосфена;
(2) INVEULER - вычисление значений обратной функции Эйлера $\varphi^{-1}(m)$;
(3) PTREES - построение корневых деревьев.
Наиболее трудоемкой операцией является (2), поскольку для вычисления $\varphi^{-1}(m)$ в общем случае нужно иметь для $\varphi^{-1}(x)$ всех $x < m$, а также знать разложение натуральных чисел на простые множители.
(В июле 2000 г. в Интернете с помощью поисковой системы AltaVista был обнаружен сайт с компьютерной программой для вычисления значений обратной функции Эйлера $\varphi^{-1}(m)$ и обсуждением того, как это делать *[Rytin* 1999]. Эта программа полезна тем, что вычисляет искомые значения сразу для произвольного m. С ее помощью программа для построения корневых деревьев может быть значительно упрощена.)
С помощью программы В.И.Шалака были вычислены значения $\varphi^{-1}(m)$ для $m < 200000$, что позволило построить корневые деревья для первых 729 простых чисел. Построение корневого дерева для простого числа 5521, чей порядковый номер 730, потребовалось уже вычисления $\varphi^{-1}(m)$ для $m \leq 285062$. В конце 1996 г. это число было преодолено, в результате чего были построены деревья для первых 2370 простых чисел. В последнем случае потребовалось вычисление обратной функции Эйлера $\varphi^{-1}(m)$ для $m \leq 798279$. Чтобы преодолеть простое число 21089 (№ 2371), необходимо было просчитать $\varphi^{-1}(m)$ для $m \leq 2215802$(см. *[Карпенко* 1997]).
В августе 2000 г. В.И.Шалаком были расширены возможности компьютерной программы, а также введена оценка мощности (число вершин) деревьев. Программа «высеяла» 1207706 простых чисел, что позволило просчитать значения обратной функции Эйлера для $m \leq 3317744$. Построение корневых деревьев остановилось на простом числе 30689 (№ 3310), поскольку при вычислении $\varphi^{-1}(3228368)$ было получено следующее множество значений {5104689, 6053205, 6456752, 8070940, 10209378, 12106410}. Однако с помощью программы М.Рытина вычисления дали следующий результат $\varphi^{-1}(5104688) = \{\varnothing\}$ и $\varphi^{-1}(6053204) = \{\varnothing\}$. Таким образом, построение графа для числа 30689 оказалось завершенным.

Неравномерность мощностей деревьев, представляющих простые числа, поразительна. Для основной части простых чисел корневое дерево состоит из двух элементов *{p, 2p}*. Но встречаются такие «монстры», как число 3313 (№ 466) - 2125 элементов. Пока абсолютный рекорд принадлежит графу для числа 21089 (№2371) - 5557 элементов, число 30689 (№ 3310) - 2255 элементов. В последующем такой феномен только усиливается.
Статистическое распределение корневых деревьев, представляющих простые числа, и их свойства - вопрос для специального исследования. Отдельного рассмотрения заслуживает вопрос о связи между данным простым числом p (корнем дерева) и тем наибольшим простым числом, которое, требуется при нахождении нужных значений обратной функции Эйлера $\varphi^{-1}(m)$. Речь идет о некоторой функции $P(p)$, порождающей это большое простое число. Причем это последнее число стремительно растет. Оказывается, в каждом простом числе содержится «информация» о своем, назовем так, *максимальном напарнике.*

6.3.1. Гипотеза о конечности корневых деревьев

Подчеркнем, что все построенные деревья для простых чисел являются *конечными.* Если бы четных чисел, не являющихся значениями $\varphi(n)$, было бы конечное множество, то начиная с некоторого простого числа p, все классы эквивалентности $\mathcal{X}_p$ были бы бесконечной мощности. Но, как уже отмечалось, из результата *[Rehman* 1977] следует, что существует бесконечное множество четных чисел, не являющихся значениями $\varphi(n)$. Таким образом, необходимое условие для конечности каждого класса эквивалентности $\mathcal{X}_p$ найдено. Наша гипотеза состоит в следующем (и это вынесено в заголовок статьи *[Karpenko* 1986]).

ГИПОТЕЗА 1. $\forall p(|\{n \ \exists k(\varphi_k^*(n) = p)\}| < \aleph_0$,

т. е. для каждого простого числа p его класс эквивалентности $\mathcal{X}_p$ конечен. Соответственно, каждое корневое дерево $\mathcal{T}_p$ тоже конечно.
Обратим внимание, что, по-видимому, существует некоторая связь между кардинальной степенью полноты логик Лукасевича $Ł_n$ (см. раздел 3.3.2 и Приложение 1) и корневыми деревьями для простых чисел, т.е. между функциями $\gamma(n)$ и $\varphi^{-1^*}(p)$. Простое

наблюдение показывает, что именно для *больших* корневых деревьев функция *γ(p)* также дает большие значения. Поскольку степень кардинальной полноты конечно-значных логик $\mathbf{Ł}_n$ всегда конечна, то нахождение такой связи между указанными функциями дало бы утвердительное решение данной гипотезы.
Итак, каждое простое число *p* представимо в виде корневого дерева с выделенной вершиной *p*. В итоге происходит *структурализация* простых чисел, и здесь мы находимся только в начале этого процесса.

6.4. *p*-абелевы группы

Известно, что корневые деревья широко используются в комбинаторике, вычислительной технике, химии и физике и в особенности при решении различных перечислительных задач. Поэтому соответствие между корневыми деревьями и простыми числами может оказаться весьма полезным.
В связи с этим обратим внимание на работу А.Хэлса *[Hales* 1971], где по каждому корневому дереву строится *p*-абелева группа, т е коммутативная группа, порядки всех элементов которых являются степенями фиксированного простого числа.
Пусть *p* - произвольное простое число. Вершины дерева, кроме корневой, выступают в качестве образующих элементов x_1 ..., x_n, и для каждого направленного ребра $\mathbf{i} \rightarrow \mathbf{j}$ (сверху вниз) принимается соотношение $p\mathbf{x_i} = \mathbf{x_j}$ (или, если j = t, где t - корень дерева, $px_i = 0$ *p*-абелева группа G_p имеет p^n элементов. Поскольку *p* в G_p является произвольным простым числом, то данное представление имеет сугубо теоретический интерес. Но так как в нашем случае каждое простое число *p* представимо в виде только «своего» корневого дерева $\mathcal{T}_p$, то теперь для каждого такого дерева строится конкретная *p7*-абелева группа. Например, пусть *p* = 3. Тогда по корневому дереву $\mathcal{T}_3$ (см. выше), имеющему кроме корня две вершины $x_1 = 4$ и $x_2 = 6$, следующим образом задается *p*-абелева группа G_3:

$$\mathbf{x_1} \oplus \mathbf{x_1} \oplus \mathbf{x_1} = 0 \quad \text{и} \quad \mathbf{x_2} \oplus \mathbf{x_2} \oplus \mathbf{x_2} = 0,$$

где $\oplus$ есть групповая операция, 0 есть единичный элемент группы и G_3 имеет 9 ($=3^2$) элементов. Таким образом, каждое простое число обладает определенной G_p-структурой.
Как следует из *[Hales* 1971], *p*-абелевы группы представлены классом эквивалентных корневых деревьев. Перечисление этих классов поставлено Хэлсом в качестве сложнейшей проблемы. Эта проблема

решена в *[Schulz* 1982], где алгоритмически строятся (вычисляются на компьютере) все представления данной группы. В результате имеем

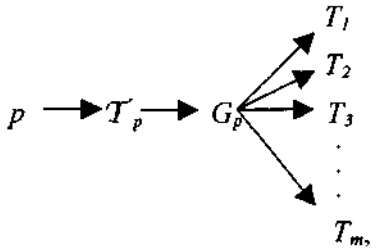


Рис. 1
где *p* - простое число, $\mathcal{T}_p$ - корневое дерево, представляющее это число, G_p - *p*-абелева группа, представляющая данное $\mathcal{T}_p$, $\{T_1, T_2, T_3, \ldots, T_m\}$ - класс корневых деревьев, представляющих данное G_p и изоморфных $\mathcal{T}_p$. Отсюда следует, что существуют простые числа, которые представимы некоторым *классом* изоморфных деревьев, хотя для многих корневых деревьев класс эквивалентности состоит из самого этого дерева, например, $\{\mathcal{T}_2\}, \{\mathcal{T}_3\}, \{\mathcal{T}_5\}$ и т.д.
Обратим внимание на весьма важный факт: не каждому корневому дереву из класса всех корневых деревьев можно поставить в соответствие простое число в предложенном алгоритме (см. раздел 6.3). И эта невозможность принципиальна. Например, нельзя закодировать простым числом следующее дерево.

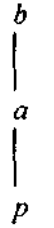


Рис.2
Поскольку вершина *a* должна обозначаться нечетным числом v_0 (в силу того, что имеется вершина *b),* то должна существовать вершина с числом *2a* (в силу мультипликативности функции $\varphi(n)$ $\varphi(2 \cdot v_0) = \varphi(2) \cdot \varphi(v_0) = 1 \cdot \varphi(v_0) = \varphi(v_0)$), т.е. дерево приобретает следующий вид:

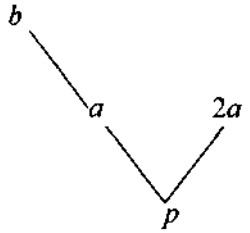


Рис.3

Но и такое дерево не кодируется у нас простым числом, поскольку для существования только одного такого ребра от *a* к *b* должно выполняться условие $N_\varphi(m) = 1$, где $N_\varphi(m)$ есть число чисел *n,* для которых функция $\varphi(n) = m$, т.е. найдется только одно такое *n.*

(Число $N_\varphi(m)$ получило название «многочисленности» (multiplicity) *[Guy* 1994]. Крупнейший польский математик В. Серпинский высказал гипотезу, что все целые числа выступают в качестве $N_\varphi(m)$. П. Эрдёш доказал *[Erdos* 1958], что, появившись однажды, число $N_\varphi(m)$ появляется бесконечно много раз. Гипотеза Серпинского была доказана К. Фордом *[Ford* 1998, 1998].)

Здесь мы вышли на известную *гипотезу Кармикайла* (R.Carmichael) о функции Эйлера φ(*n*), которая утверждает, что для любого *m* уравнение $\varphi(n) = m$ либо неразрешимо, либо имеет *по крайней мере* два решения. Например, для *m*= 14 не имеет решений, а для *m* = 54 - только два решения 81 и 162. Как показано в *[Ribenboim* 1996], эта гипотеза эквивалентна утверждению, что существуют $m \neq n$ такие, что $\varphi(n) = \varphi(m)$.

Гипотеза была доказана Кармикайлем в статье *[Carmichael* 1907]. Этот результат появился даже в качестве упражнения в книге *[Carmichael* 1914]. Однако им же самим была обнаружена ошибка в доказательстве *[Carmichael* 1922] и с этого времени гипотеза остается открытой. Был предпринят целый ряд попыток найти контрпример или хотя бы определить нижнюю границу для контрпримера (см. *[Klee* 1947, 1969], *[Masai & Valette* 1982], *[Hagis* 1986]), пока в статье *[Schlafly & Wagon* 1994] было показано, что любой контрпример гипотезе Кармикайла должен иметь больше чем 10 000 000 цифр. Это значит, возвращаясь к рис 3, что в этом огромном интервале, если *a* является значением функции φ(x), то всегда найдется как минимум два решения уравнения $\varphi(x) = a$, а именно b_1 и b_2. Тогда рис. 3 примет следующий вид:

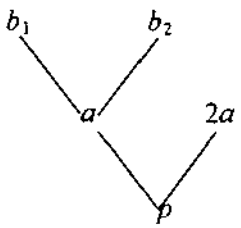


Рис.4

Правда, для первой тысячи простых чисел подобного дерева не нашлось.

Одним словом, соответствие между корневыми деревьями и простыми числами не является взаимно-однозначным. (В *[Gobel* 1980] устанавливается взаимно-однозначное соответствие между множеством всех натуральных чисел и множеством корневых деревьев.). А это приводит к тому, что могут существовать такие классы эквивалентности $\{T_1, T_2, T_3, \ldots, T_m\}$, в каждом из которых нет ни одного дерева T_p. Как раз пример класса изоморфных деревьев, приведенных в *[Schulz* 1982], является таковым. Здесь возникает истинная

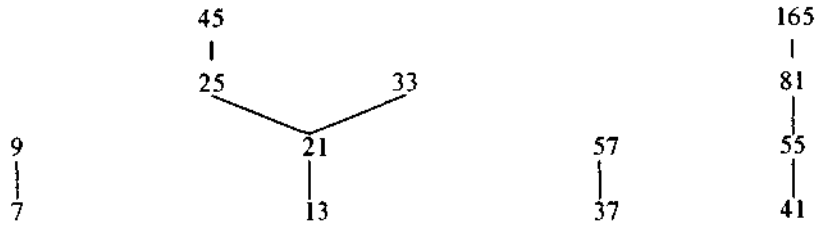
ПРОБЛЕМА. *Определить подкласс p-абелевых групп, которые характеризуются только* T_p*-деревьями.*

Тогда на этом пути можно было бы утвердительно решить гипотезу Кармикайля, если окажется, что этот подкласс *p*-абелевых групп обладает свойством, которое не допускает представления в виде деревьев, подобных приведенным на рис 2 и 3. Однако деревьям, приведенным на этих рисунках, соответствуют простые числа, если сами деревья представить в «более компактном виде». Тем более, что из доказательства гипотезы Серпинского следует, что мощность *большинства* деревьев необъятна

6.5. Сокращенные корневые деревья

Обратим внимание, что полужирным шрифтом в дереве для простого числа 241 выделены вершины, для которых $\varphi^{-1^*}(v_0) \neq \varnothing$, т.е. выделены только вершины дерева, которые кодируются нечетными числами v_0, такими, что v_0-1 есть значение функции Эйлера. Деревья с таким множеством вершин назовем *сокращенными корневыми деревьями* (их еще называют деревьями без висячих вершин, кроме корневой). Таким образом, число вершин сокращенного корневого

дерева (С.К.Д.) есть число применений функции $\varphi^{-1}(v_0)$. Приведем С.К.Д. для некоторых простых чисел. Для *большинства* простых чисел С.К.Д. будут обозначаться точками и кодироваться самими простыми числами, например, в первой сотне натуральных чисел таких деревьев будет 18 2, 3, 5, 11, 17, 19, 23, 29, 31, 47, 53, 59, 67, 71, 79, 83, 89, 97. Приведем другие С.К.Д. для нескольких первых простых чисел:



Далее, начиная с простого числа 101 (№ 26) и до числа 541 (№ 100), будем представлять деревья (в данном случае С.К.Д.) опять же, как их принято сохранять в текстовых файлах:

(Для простого числа 181 в *[Карпенко* 1983] и, соответственно, в *[Карпенко* 1989] С.К.Д. построено неправильно.)


```
101
103
107
109
    133
        161
            205
        201
113
    145
        185
        273
            289
                585
127
131
137
139
149
151
157
    169
        261
            393
163
167
173
179
```

181

209

265

217

333

501

625

689

865

1377

1665

785

985

1113

1185

297

191

193

221

253

301

453

381

441

357

537

197

199

211

223

227

229

233

295

343

361

693

1041

1965

239

241

325

513

369

705

1173

1761

2225

2785

4893

7341

11013

16521

385

765

465

885

1329

525

251

257

263

269

271

277

329

417

281

283

293

307

311

313

477

717

507

529

801

1025

1285

1353

897

1005

317

331

337

347

349

353

445

359

367

373

379

383

389

397

469

553

621

933

1401

401

505

637

409

419

421

633

431

433

481

861

1293

1941

2913

545

685

657

665

777

439

443

449
493
581
649
981
1473
1197
565
849
457
461
463
467
479
487
491
499
503
509
521
523
541
589
813
841
1225
2163
2209
1617
837

Теперь нетрудно показать, что деревья на рис. 2 и 3, если их рассматривать как С.К.Д., соответственно кодируются простыми числами, например, 401 и 1381:

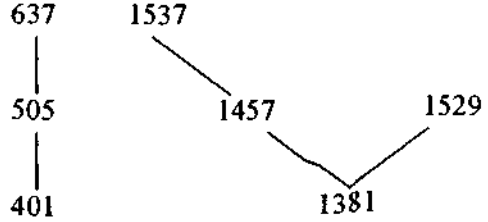


Рис.4

Представление простых чисел в виде С.К.Д. ставит новые проблемы. Пусть T_r - число корневых деревьев, имеющих r вершин. Вопрос о числе изоморфных корневых деревьев порядка r детально рассмотрен в книге *[Харари & Палмер* 1977]. Там же в качестве примера приведены корневые деревья не выше четвертого порядка, т.е. для $r \le 4$:

(Сайт в Интернете (см. *[Ruskey* 1996-2000]) начинается сразу с графов для корневых деревьев ранга $r \le 5$, число которых 9.)

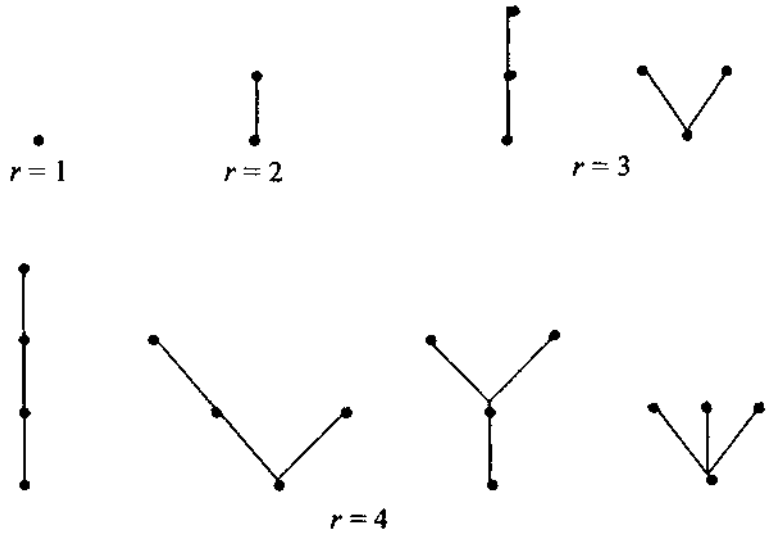


Рис.5

Можно показать, что для каждого из деревьев на рис. 5, взятого как С.К.Д., существует простое число. Для всех них (кроме второго дерева в $r = 4$) примеры можно найти в разделе 6.4. Оставшееся же дерево, как уже показано, кодируется простым числом 1381, которое является наибольшим для деревьев ранга 4. Все С.К.Д. ранга 5 также кодируются простыми числами, наибольшее из которых уже 26177.

В связи с предложенными представлениями простых чисел возникает ряд вопросов:

1. Характеризуется ли *каждое* натуральное число $n \geq 2$ $\mathcal{T}_p$ -деревом, число вершин которого соответствует этому числу? Например, число 2 характеризуется $\mathcal{T}_2$ деревом (и многими другими, а именно всеми $\mathcal{T}_p$ деревьями вида *{p, 2p}),* 3 - $\mathcal{T}_3$-деревом; 4 -$\mathcal{T}_5$ деревом; 5 - $\mathcal{T}_{457}$-деревом; 6 - $\mathcal{T}_{17}$-деревом; для числа 7 соответсвующего дерева для первой сотни простых чисел нет.
2. Этот же вопрос относится и к С.К.Д. для $n \geq 1$.Например, число 1 характеризуется $\mathcal{T}_2$-деревом; 2 - $\mathcal{T}_7$ деревом; 3 - $\mathcal{T}_{277}$-деревом; 4 - $\mathcal{T}_{41}$-деревом; 5 -$\mathcal{T}_{13}$ деревом; 6 - $\mathcal{T}_{43}$-деревом; 7 - $\mathcal{T}_{73}$-деревом, и т.д. В свою очередь, для всякого ли изоморфного С.К.Д. поряда *r* существует простое число? Для $r \leq 4$, как было показано, это имеет место.
3. Конечна или бесконечна мощность простых чисел представимых: a) одним и тем же $\mathcal{T}_p$; b) одним и тем же С.К.Д. Например, в первой сотне простых чисел содержится 60 $\mathcal{T}_p$-деревьев вида *[p, 2p]* и 77 С.К.Д. ранга 1; 4 С.К.Д. ранга 2, и т.д.

Таким образом, от разбиения натурального ряда чисел на классы эквивалентности, в каждом из которых содержится одно и только одно простое число, мы перешли к различным разбиениям на классы эквивалентности самого ряда простых чисел.

Одним из таких отношений эквивалентности может служить число *k* применений обратной функции Эйлера $\varphi^{-1^*}(v_0)$. При k =1, как уже отмечалось, в класс эквивалентности из первой сотни простых чисел попадают 77 чисел (включая число 2), из второй сотни - 76 чисел. При $k = 2$ имеем, соответственно, 2 и 6.

Другим отношением эквивалентности на множестве простых чисел может служить количество вершин *n* $\mathcal{T}_p$-дерева, начиная с $n = 2$, 3, 4, 5, ...; или количество вершин *n* С.К.Д., начиная с $n = 1, 2, 3,4,5,...$

В Таблицах чисел (таблица 3) приведены оценки мощности обычных корневых деревьев $\mathcal{T}_p$ и сокращенных корневых деревьев С.К.Д. для $p \leq 1000$.

7. Матричная логика для простых чисел

Оказывается, можно построить такую матричную логику $\mathbf{K}_{n+1}$, *которая имеет класс тавтологий т.т.т., когда n есть простое число. Более того, удается показать, что по своим функциональным свойствам* $\mathbf{K}_{n+1}$ *совпадает с логикой Лукасевича* $\mathbf{Ł}_{n+1}$ *для случая, когда n есть простое число. Отсюда приходим к идее построения штриха Шеффера для простых чисел. В результате получаем формулу, в которую штрих Шеффера входит 648 042 744 959 раз. Логика* $\mathbf{K}_{n+1}$ *определеяет класс тавтологий, алгебро-логических полиномов, каждый из которых задает алгоритм вычисления простых чисел. Эти вычисления весьма громоздки, но если рассмотреть комбинацию двух матричных логик* $\mathbf{K}_{n+1}$ *и* $\mathbf{K'}_{n+1}$, *то можно определить закон порождения классов простых чисел. Доказывается, что в эти классы попадают все простые числа. Таким образом, множество простых чисел разбивается на определенные классы эквивалентности, задаваемых некоторыми свойствами импликации Лукасевича.*

7.1. Характеризация простых чисел посредством матричной логики K_{n+1}

Возвращаясь к теореме В.К.Финна о критерии функциональной предполноты класса функций $\mathbf{Ł}_{n+1}$ (см. выше раздел 5.3), напомним, что главный результат состоит в том, что простые числа характеризуются посредством предполных классов функций, соответствующих конечнозначным логикам Лукасевича $\mathbf{Ł}_{n+1}$. Это наводит на мысль о построении такой многозначной матричной логики (обозначим ее посредством $\mathbf{K}_{n+1}$), которая имеет класс тавтологий т.т.т., когда *n* есть простое число. Тогда простые числа определялись бы классами тавтологий логики $\mathbf{K}_{n+1}$ *[Карпенко* 1982].

Определим матрицу $\mathfrak{M}^K_{n+1}$ следующим образом:

$$\mathfrak{M}^K_{n+1} = < V_{n+1}, \sim, \rightarrow^K, \{n\} > \quad (n \geq 3,\ n \in N),$$

где

$$\sim x = n - x,$$

$$x \to^K y = \begin{cases} y, \text{ если } 0 < x < y < n \text{ и } (x,y) \neq 1 \ (i) \\ y, \text{ если } 0 < x = y < n \quad (ii) \\ x \to y, \text{ в остальных случаях} \quad (iii), \end{cases}$$

где $(x, y) \neq 1$ обозначает, что x и y не являются взаимнопростыми числами, т. е. x и y имеют общий делитель, отличный от 1, а $x \to y$ есть импликация Лукасевича, которую для сравнения с импликацией $x \to^K y$ определим следующим образом.

$$x \to y = \begin{cases} n, \text{ если } x \leq y \\ n - x + y, \text{ если } x > y. \end{cases}$$

Таким образом, функция $x \to^K y$ существенно отличается от $x \to y$, когда $0 < x \leq y < n$.

Множество всех суперпозиций функций ~x и $x \to^K y$ обозначим посредством K_{n+1}.

Следующие два свойства исходных матричных функций (как и в L_{n+1}) особенно важны.

$\sim\sim x = x$ (закон снятия двойного отрицания),

$x \to y = \sim y \to \sim x$ (контрпозиция)

В дальнейшем нам понадобятся следующие два *свойства отношения делимости* (сокращенно: с.о.д.) (см. *[Бухштаб* I960]):

I (с.о.д). Если числа x и y делятся на z, то их сумма $x + y$ делится на z.

II (с.о.д). Если числа x и y делятся на z, причем $x \geq y$, то и их разность x-y делится на z.

Лемма 1. *Пусть n есть простое число. Тогда, если* $x < \sim x$, *то* $x \to^K \sim x = n$.

ДОКАЗАТЕЛЬСТВО.

Сначала покажем, что $(x, n\text{-}x) = 1$, т.е. x и n-x есть взаимно-простые числа. Допустим обратное, т. е. $(x, n\text{-}x) \neq 1$. Тогда $d|x$ и $d|n\text{-}x$, где d есть делитель x и n-x, отличный от 1. Тогда из (с.о.д) следует, что $d|(x+n-x)$, т. е $d|n$. Но это противоречит условию, что n есть простое число. Таким образом, $(x, n\text{-}x) = 1$. Отсюда в силу пункта (iii) определения $x \to^K y$, $x \to^K \sim x = n$. Лемма 1 доказана.

Теперь мы можем дать определение простого числа в терминах класса общезначимых формул.

Теорема 1. *Для любого* $n \geq 3$ n *есть простое число т.т.т., когда* $n \in K_{n+1}$.

ДОКАЗАТЕЛЬСТВО.

I. *Достаточность.* Если n есть простое число, то $n \in K_{n+1}$. Пусть n есть простое число. Тогда существует такая формула U:

$$\sim((x \to^K y) \to^K \sim(x \to^K y)) \to^K (\sim(x \to^K y) \to^K (x \to^K y)),$$

что U = n. Рассмотрим подформулы $U_1 = (x \to^K y) \to^K \sim(x \to^K y)$ и $U_2 = \sim(x \to^K y) \to^K (x \to^K y)$. Очевидно, что для тех случаев, когда $x \to^K y = 0$ или $x \to^K y = n$, $U = n$. В силу леммы 1, если $x \to^K y < n/2$, то $U_1 = n$, и тогда $\sim U_1 = 0$. Отсюда в силу пункта (*iii*) определения $x \to^K y$, $\sim U_1 \to^K U_2 = n$, т. е. U = n. Если же $x \to^K y > n/2$, то $U_2 = n$. Отсюда, $\sim U_1 \to^K U_2 = n$, т. е. U = n.

II. *Необходимость.* Если $n \in K_{n+1}$, то n есть простое число. Докажем контрпозицию этого утверждения. Пусть n не есть простое число. Тогда n имеет делители (по крайней мере один), отличные от 1 и n. Пусть d есть один из таких фиксированных делителей. Посредством D обозначим множество элементов вида $m \cdot d$, где $m \in \{1, 2, \ldots, (n/d)\text{-}1\}$. Таким образом, D есть класс положительных чисел, сравнимых по модулю d, такой, что $m \cdot d \equiv 0 (\mathrm{mod}\ d)$. Покажем, что множество D замкнуто относительно $\sim x$ и $x \to^K y$.

Пусть $x \in D$, т е. $x = m \cdot d$. Тогда $\sim x = n\text{-}(m \cdot d)$. Из II(с.о.д) следует, что $d|n\text{-}(m \cdot d)$. Отсюда $\sim x \in D$.

Пусть $x, y \in D$ и $x = m_i \cdot d$, $y = m_j \cdot d$. Тогда $x \to^K y = m_i \cdot d \to^K m_j \cdot d$. Имеем два случая.

1) $m_i \cdot d \leq m_j \cdot d$. Из определения $x \to^K y$ следует, что $m_i \cdot d \to^K m_j \cdot d = m_j \cdot d$. Отсюда, $x \to^K y \in D$.

2) $m_i \cdot d > m_i \cdot d$. Из определения $x \to^K y$ следует, что $m_i \cdot d \to^K m_j \cdot d = n - m_i \cdot d + m_j \cdot d$. Из I(с.о.д) и II(с.о. д.) следует, что $d|n\text{-}m_i \cdot d + m_j \cdot d$. Отсюда $x \to^K y \in D$.

Следовательно, если n не есть простое число, то не существует суперпозиции $f(x)$ функций $\sim x$ и $x \to^K y$ такой, что f(x) = n.

Таким образом, теорема 2 дает новое определение простого числа. Введя обычным образом пропозициональный язык и функцикцию оценки v на нем (см. раздел 3.2), получаем, что матричная логика $\mathbf{K_{n+1}}$ имеет класс тавтологий т.т.т., когда n есть простое число, т.е. каждое простое число определяется соответствующим классом тавтологий. В связи с этим возникает нетривиальный вопрос о функциональных свойствах $\mathbf{K_{n+1}}$.

7.1.1. Функциональные свойства логики K_{n+1}

Теорема *2. Для любого* $n \geq 3$ *такого, что n есть простое число,* $K_{n+1} = Ł_{n+1}$ *[Карпенко* 1989], *[Karpenko* 1989].

ДОКАЗАТЕЛЬСТВО.

I. $K_{n+1} \subseteq Ł_{n+1}$

Из определения $x \to^K y$ следует, что множество функций K_{n+1} не является функционально полным ни для какого $n \geq 2$. По крайней мере функции ~x и $x \to^K y$ сохраняют множество значений {0,*n*}, как и множество функций $Ł_{n+1}$. Поскольку множество $Ł_{n+1}$ функционально предполно для случая, когда *n* есть простое число *[Бочвар & Финн* 1972] (см. выше раздел 5.3), то для этого случая $K_{n+1} \subseteq Ł_{n+1}$.

II. $Ł_{n+1} \subseteq K_{n+1}$

Надо показать, что функция $x \to y$ определима посредством суперпозиции ~x и $x \to^K y$. Это можно сделать с помощью следующих определений:

$$(A)\ x \to^1 y = \sim((y \to^K x) \to^K \sim(y \to^K x)) \to^K (x \to^K y)$$

$$(B)\ x \vee^1 y = (x \to^1 y) \to^1 y$$

$$(C)\ x \to^2 y = ((x \to^K y) \to^K (\sim y \to^K \sim x)) \vee^1 ((\sim y \to^K \sim x) \to^K ((x \to^K y))$$

$$(D)\ x \vee^K y = (x \to^K y) \to^K y$$

$$(E)\ x \vee^2 y = (x \vee^K y) \vee^1 (x \vee^K y) = x \vee y = max(x,y)$$

$$(F)\ x \to^3 y = (x \to^K y) \vee^2 (\sim y \to^K \sim x)$$

$$(G)\ x \vee^3 y = (x \to^3 y) \to^3 y$$

$$(H)\ x \to^4 y = ((x \vee^3 y) \to^2 (x \vee^2 y)) \to^1 (x \to^3 y)$$

$$(I)\ x \to^5 y = (x \to^4 y) \vee^1 (\sim y \to^4 \sim x) = x \to y = min(n, n - x + y)$$

Рассмотрим каждое из этих определений, выделяя в них только те свойства, которые используются для определения $x \to y$.

$(A)\ x \to^1 y = \sim((y \to^K x) \to^K \sim(y \to^K x)) \to^K (x \to^K y)$

1. Пусть $x < y$ и $y = n$. Тогда $x \to^K y = n$ *(iii)*. Отсюда $x \to^1 y = n = x \to y$.

2. Пусть x = y. Имеем два случая:

2.1. x < *n.l.* Имеем два подслучая:

2.1.1. x = 0. Тогда в силу определения $x \to^K y$ *(iii)*, $x \to^K y = n$. Отсюда, $x \to^1 y = x \to y$.

2.1.2. $x \neq 0$. Тогда $x \to^1 y = \sim(x \to^K \sim x) \to^K x$. В силу леммы 1, $x \to^K \sim x = n$. Отсюда $\sim(x \to^K \sim x) = 0$ и, следовательно, $x \to^1 y = 0 \to^K x = n = x \to y$.

2.2. x> *n/2*

2.2.1. x = *n*. Тогда $x \to^K y = n$. Отсюда $x \to^1 y = n = x \to y$

2.2.2. $x \neq n$. Тогда $x \to^1 y = \sim(x \to^K \sim x) \to^K x = (n - (n - x + n - x)) \to^K x = (2x - n) \to^K x$. Покажем, что $((2x - n), x) = 1$. Допустим обратное, т. е. $d \mid (2x - n)$ и $d \mid x$, где $d \neq 1$. Заметим, что $(2x - n) < x$ для любого x > *n/2*. Тогда из II(с.о.д.) следует, что $d \mid (x - (2x - n))$, т. е. $d \mid (n - x)$. Поскольку $d \mid x$, то из I(с.о.д.) следует, что $d \mid (n - x + x)$, т. е. $d \mid n$, а это противоречит условию теоремы о том, что *n* есть простое число. Следовательно, допущение неверно и в силу определения $x \to^K y$ *(iii)*, $\sim(x \to^K \sim x) \to^K x = n$. Отсюда $x \to^1 y = x \to y$.

3. x > y и $x = n$ Тогда $y \to^K x = n$ и $x \to^K y = y$. Отсюда $x \to^1 y = \sim(n \to^K 0) \to^K y = n \to^K y = y = x \to^K y$.

Обратим внимание на то, что $x \to^1 y$ в отличие от $x \to^K y$ всегда принимает значение *n,* когда x = y, т.е. как и импликация Лукасевича x → y. Теперь заметим, что из свойств $x \to^1 y$ следует, что при *n* =3 и *n* = 5 для любых $0 \leq x, y \leq 5$, $x \to^1 y = x \to y$. Но уже при *n* = 7, если x = 4 и y = 2, то $x \to^1 y = 7$, тогда как x→y = 5 Таким образом, если x > y, то в общем случае $x \to^1 y \neq x \to y$.

$(B)\ x \vee^1 y = (x \to^1 y) \to^1 y$

Поскольку $x \vee^1 y$ определяется анологичным образом, как и дизъюнкция Лукасевича $x \vee y = (x \to y) \to y$, то для случаев, когда $x \to^1 y = (x \to y)$, $x \vee^1 y = max(x, y)$.

$(C)\ x \to^2 y = ((x \to^K y) \to^K (\sim y \to^K \sim x)) \vee^1 ((\sim y \to^K \sim x) \to^K ((x \to^K y))$.

Рассмотрим подформулы $C_1 = (x \to^K y) \to^K (\sim y \to^K \sim x)$ и $C_2 = ((\sim y \to^K \sim x) \to^K ((x \to^K y))$:

1. x < y и y = *n*. Тогда $x \to^K y = n$. Отсюда $C_2 = n$ и, значит, $C_1 \vee^1 C_2 = n$. Следовательно, $x \to^2 y = x \to y$.

2. x = y.

2.1. x<*n*/2.

2.1.1. x = 0. Тогда $x \to^K y = n$ и, следовательно, $x \to^2 y = x \to y$ (см. случай *C.1)*.

2.1.2 $x \neq 0$. Тогда $C_1 = x \to^K \sim x$. По лемме 1, $x \to^K \sim x = n$ Отсюда $C_1 \vee^1 C_2 = n$ и, следовательно, $x \to^2 y = x \to y$.

2.2. x> *n*/2.

2.2.1. x = *n*. Тогда $\sim y \to^K \sim x = n$. Отсюда $C_1 = n$ и, значит, $C_1 \vee^1 C_2 = n$. Следовательно, $x \to^2 y = x \to y$.

2.2.2. $x \neq n$. Тогда $C_2 = \sim x \to^K x = n$ и, следовательно, $x \to^2 y = x \to y$ (см С.2.1.2).

3. x > y и $x, y \in \{1, 2, \ldots, n\text{-}1\}$. Из определения $x \to^K y$ следует, что тогда $x \to^K y = x \to y$. Поскольку $x \to y = \sim y \to \sim x$, то $x \to^K y = \sim y \to^K \sim x$. Тогда в силу определения $x \to^K y$ *(n)*, $C_1 = (x \to^K y) \to^K (\sim y \to^K \sim x) = x \to y$ и $C_2 = (\sim y \to^K \sim x) \to^K (x \to^K y) = x \to y$. Следовательно, $x \to^2 y = C_1 \vee^1 C_2 = x \to y$.

Таким образом, если x > y, то в отличие от $x \to^1 y$, $x \to^2 y = x \to y$ для всех $x, y \in \{1, 2, \ldots, n\text{-}1\}$

(D) $x \vee^K y = (x \to^K y) \to^K y$:

1. x<y.

1.1. x = 0 или/и y = *n*. Тогда $x \to^K y = n$ и, следовательно, $x \vee^K y = n \to^K y = y = max(x, y)$.

1.2. (x, y) = 1. Тогда $x \to^K y = n$ и, следовательно, $x \vee^K y = n \to^K y = y = max(x, y)$

1.3. $(x, y) \neq 1$. Тогда $x \to^K y = y$ и, следовательно, $x \vee^K y = y \to^K y = y = max(x, y)$.

2. x = y.

2.1. $x, y \in \{0, n\}$. Тогда $x \to^K y = n$ и, следовательно, $x \vee^K y = n \to^K y = y = max(x, y)$.

2.2. $x \in \{1, 2, \ldots, n\text{-}1\}$. Тогда $x \to^K y = y$ и, следовательно, $x \vee^K y = y \to^K y = y = max(x, y)$.

3. x > y. Имеем два под случая.

3.1. $x \neq n$. Тогда в силу определения $x \to^K y$, $x \vee^K y = (n - x + y) \to^K y = n - (n - x + y) + y = x = max(x, y)$.

3.2. x = *n*. Тогда $x \vee^K y = (n - n + y) \to^K y = y \to^K y$. Имеем два подслучая.

3.2.1. y = 0. Тогда $x \vee^K y = y \to^K y = n = max(x, y)$.

3.2.2. $y \neq 0$. Тогда $x \vee^K y = y \to^K y = y \neq max(x, y)$.

Таким образом, дизъюнкция $x \vee^K y$ в отличие от $x \vee y$ не является коммутативной, а именно в последнем подслучае. Отсюда $x \vee^K y \neq max(x, y)$. Заметим, что $x \vee^1 y = max(x, y)$ для этого подслучая и это свойство $x \vee^1 y$ существенно использовалось при определении $x \to^2 y$.

(E) $x \vee^2 y = (x \vee^K y) \vee^1 (y \vee^K x) = x \vee y = max(x, y)$.

Достаточно проверить случай (D.3.2) т.е. пусть $x = n$ и $y \neq 0$. Тогда $y \vee^K x = n$ и, следовательно, $x \vee^2 y = x \vee y = max(x, y)$.

(F) $x \to^3 y = (x \to^K y) \vee^2 (\sim y \to^K \sim x)$.

1. x<y.

1.1. x = 0 и/или y = *n*. Тогда $x \to^K y = n$ и $y \to^K \sim x = n$ Отсюда $x \to^3 y = n = x \to y$.

1.2. (x, y) = 1 и/или $(n - y, n - x) = 1$. Тогда $x \to^K y = n$ и/или $\sim y \to^K \sim x = n$. Отсюда $x \to^3 y = n = x \to y$.

1.3. $(x, y) \neq 1$ и/или $(n - y, n - x) \neq 1$ (например, если *n* = 11, x = 2 и y = 8, то ~y = 3 и ~x = 9). Тогда $x \to^K y = y$ и $\sim y \to^K \sim x = \sim x$ Отсюда $x \to^3 y = y \vee^2 \sim x$. Имеем два подслучая:

1.3.1. $(x + y) < n$. Тогда $y < (n - x)$. В противном случае $(x + y) > n$, что противоречит допущению. Поэтому $x \to^3 y = \sim x$

1.3.2. $(x + y) > n$. Тогда $y > (n - x)$ и, следовательно, $x \to^3 y = y$.

2. x = y.

2.1. x<*n*/2.

2.1.1. x = 0. Тогда $x \to^K y = n$ и, следовательно, $x \to^3 y = n \vee^2 (\sim y \to^K \sim x) = n = x \to y$.

2.1.2. $x \neq 0$. Тогда $x \to^K y = x$ и $\sim y \to^K \sim x = \sim x$, где $x < \sim x$ Отсюда $x \to^3 y = x \vee^2 \sim x = \sim x$.

2.2. x> *n*/2.

2.2.1. x = *n*. Тогда $\sim y \to^K \sim x = n$ и, следовательно, $x \to^3 y = (x \to^K y) \vee^2 n = x \to y$.

2.2.2. $x \neq n$. Тогда $x \to^K y = x$ и $\sim y \to^K \sim x = \sim x$, где $x > \sim x$ Отсюда $x \to^3 y = x \vee^2 \sim x = x$.

3. x > y. Поскольку $x \to^K y = x \to y$ для этого случая, то $x \to^3 y = x \to y$.

(G) $x \vee^3 y = (x \to^3 y) \to^3 y$.

1. x < y.

1.2. (x, y) = 1 и/или $(n - y, n - x) = 1$. Тогда в силу (F.1.1) $x \to^3 y = n$, и, следовательно, $x \vee^3 y = n \to^3 y = y$ *(F.3)*.

1.3. $(x, y) \neq 1$ и/или $(n - y, n - x) \neq 1$.
1.3.1. $(x + y) < n$. Тогда $x \to^3 y = \sim x$ *(F.*1.3.1) и, следовательно, $x \vee^3 y = \sim x \to^3 y$. В силу (F. 1.3.1), $y < \sim x$. Отсюда $x \vee^3 y = \sim x \to^3 y = n - (n - x) + y = x + y$ *(F.*3).
1.3.2. $(x + y) > n$. Тогда $x \to^3 y = y$ *(F.*1.3.2) и, следовательно, $x \vee^3 y = y \to^3 y$. Из условий (G. 1) и (G. 1.3.2) следует, что $y > n/2$, а из (G. 1.3) следует, что $y \neq n$. Отсюда $x \vee^3 y = y \to^3 y = y$ *(F.* 2.2.2).
2. x = y.
2.1. $x < n/2$.
2.1.1. x = 0. Тогда $x \to^3 y = n$ *(F.*2.1.1) и, следовательно, $x \vee^3 y = n \to^3 y = y$ *(F.*3).
2.1.2 $x \neq 0$. Тогда $x \to^3 y = \sim x$ *(F.*2.1.2) и, следовательно, $x \vee^3 y = \sim x \to^3 y$. Поскольку $x = y$ и $\sim x > x$, то $x \vee^3 y = \sim x \to^3 x = n - (n - x) + x = x + x = 2x$ *(F.*3).
2.2. $x > n/2$.
2.2.1. $x = n$. Тогда $x \to^3 y = x$ *(F.*2.2.2) и, следовательно, $x \vee^3 y = n \to^3 y = y$ *(F.*3)
2.2.2. $x \neq n$. Тогда $x \to^3 y = x$ *(F.*2.2.2) и, следовательно, $x \vee^3 y = x \to^3 y = x$ *(F.*2.2.2).
3. $x > y$ и $x \neq n$. Тогда $x \to^3 y = n - x + y$ (F.3). Поскольку $(n - x + y) > y$, то $x \vee^3 y = (n - x + y) \to^3 y = n - (n - x + y) + y = x$ (F.3).
(H) $x \to^4 y = ((x \vee^3 y) \to^2 (x \vee^2 y)) \to^1 (x \to^3 y)$.
1. x<y.
1.1. x = 0 и/или y = *n*. Тогда $x \to^3 y = n$ *(F.*1.1). Отсюда в силу свойств $x \to^1 y$, $x \to^4 y = n = x \to y$.
1.2. (x, y) = 1 и/или $(n - y, n - x) = 1$. Тогда $x \to^3 y = n$ *(F.* 1.2). Отсюда в силу свойств $x \to^1 y$, $x \to^4 y = n = x \to y$.
1.3. $(x, y) \neq 1$ и/или $(n - y, n - x) \neq 1$.
1.3.1. $(x + y) < n$. Тогда $x \vee^3 y = x + y$ *(G.*1.3.1), $x \vee^2 y = y$ *(E)* и $x \to^3 y = \sim x$ *(F.*1.3.1). Отсюда
$x \to^4 y = ((x + y) \to^2 y) \to^1 \sim x = (n - x - y + y) \to^1 \sim x = \sim x \to^1 \sim x = n = x \to y$ *(C.*3) и (A 2.2.2).
1.3.2. $(x + y) > n$. Тогда $x \vee^3 y = y$ *(G* 1 3 2), $x \vee^2 y = y$ *(E)* и $x \to^3 y = y$ *(F.*1.3.2). Отсюда
$x \to^4 y = (y \to^2 y) \to^1 y = n \to^1 y = y$ (C.2.2.2) и *(A3).*
Следовательно $x \to^4 y \neq x \to y$.

2. x = y.
2.1. $x<n/2$.
2.1.1. x = 0. Тогда $x \to^3 y = n$ *(F.*2.1.1). Отсюда $x \to^4 y = n = x \to y$ *(H* 1 1)
2.1.2. $x \neq 0$. Тогда $x \vee^3 y = 2x$ *(G.*2.1.2), $x \vee^2 y = x$ *(E)* и $x \to^3 y = \sim x$ *(F.*2.1.2). Отсюда
$x \to^4 y = (2x \to^2 x) \to^1 \sim x = (n - 2x + x) \to^1 \sim x = \sim x \to^1 \sim x = n = x \to y$ *(C.*3) и *(A.*2.2.2).
2.2. $x > n/2$.
2.2.1. x = n. Тогда $x \to^3 y = n$ *(F.*2.2.1). Отсюда $x \to^4 y = n = x \to y$ *(H.*1.1).
2.2.2. $x \neq n$. Тогда $x \vee^3 y = x$ *(G.*2.2.2), $x \vee^2 y = x$ *(E)* и $x \to^3 y = x$ *(F.*2.2.2). Отсюда
$x \to^4 y = (x \to^2 x) \to^1 x = n \to^1 x = x$ *(C.*2.2.2) и (A.3).
Следовательно, $x \to^4 y \neq x \to y$.
3. x>y.
3.1. $x \neq n$. Тогда $x \vee^3 y = x$ *(G.*3), $x \vee^2 y = x$ *(E)* и $x \to^3 y = x \to y$ *(F.*3). Отсюда
$x \to^4 y = (x \to^2 x) \to^1 (x \to y) = n \to^1 (x \to y) = x \to y$ *(C.*2) и *(A.*3).
3.2. $x = n$. Тогда $x \vee^2 y = n$ *(E)* и $x \to^3 y = y$ *(F.*3). Отсюда $(x \vee^3 y) \to^2 x \vee^2 y = n$ (в силу свойств $x \to^2 y$). Следовательно $x \to^4 y = n \to^1 y = y = x \to y$ (A.3).
(I) $x \to^5 y = (x \to^4 y) \vee^1 (\sim y \to^4 \sim x) = x \to y = min(n, n\text{-}x+y)$.
Рассмотрим случаи, когда $x \to^4 y = x \to y$. Поскольку $x \to y = \sim y \to \sim x$, то $x \to^4 y = \sim y \to^4 \sim x$. Отсюда, в силу свойств $x \vee^1 y$, $x \to^5 y = x \to^4 y = x \to y$.
Рассмотрим два случая из (H), в которых $x \to^4 y \neq x \to y$
1.3.2. $x < y$, $(x, y) \neq 1$ и/или $(n - y, n - x) \neq 1$, $(x + y) > n$. Тогда $x \to^4 y = y$ *(H.*1.3.2) и $\sim y \to^4 \sim x = n$ *(H.*1.3.1). Отсюда $x \to^5 y = y \vee^1 n = n = x \to y$ *(B).*
2.2.2. $x = y$, $x > n/2$ и $x \neq n$. Тогда $x \to^4 y = x$ *(H.*2.2.2) и $\sim y \to^4 \sim x = n$ *(H.*2.1.2). Отсюда $x \to^5 y = x \vee^1 n = n = x \to y$ *(B).*
Таким образом, для любых x и y, $x \to^5 y = x \to y$ и, следовательно, $L_{n+1} \subseteq K_{n+1}$. В итоге Теорема 1 доказана.
(Эта теорема имеет место также и для случая, когда *n* = 2, но тогда нужно ввести некоторые ограничения на пункты (*i*) и (ii) в

определении функции $x \to^{K} y$, что несколько осложнит доказательство, или просто положить, что для этого случая $x \to^{K} y = x \to y$.

Из этой теоремы следует (как и в случае для $Ł_{n+1}$), что существует бесконечная последовательность p_s+1-значных логик Лукасевича (p_s - к в порядке возрастания простое число в натуральном ряду чисел), которым соответствует последовательность предполных множеств функций, такая, что $Ł_{p_s+1} = T_{p_s+1}$ Для всех s= 1, 2, ... Но в отличие от $Ł_{n+1}$ в K_{n+1} только таким множествам функций соответствует матричное построение логики в определенном выше смысле.

Подчеркнем, что доказательство функциональной эквивалентности множеств функций $Ł_{n+1}$ и K_{n+1} ведется только для случая, когда n есть простое число, т.е. для последовательности простых чисел, а не для всего натурального ряда чисел. Отсюда и сложность аналитического выражения *(A) - (I),* доказывающего эту эквивалентность, которое содержит 21 345 281 вхождение импликации $x \to^{K} y$.

Теперь мы можем дать еще одно определение простого числа, но уже в терминах равенства двух классов функций:

Теорема 3. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $K_{n+1} = Ł_{n+1}$.

ДОКАЗАТЕЛЬСТВО.

I. Если $n \geq 3$ есть простое число, то $K_{n+1} = Ł_{n+1}$ (теорема 2).

II. Если $K_{n+1} = Ł_{n+1}$, то $n \geq 3$ есть простое число. Докажем контрпозицию этого утверждения. Пусть $n \geq 3$ не есть простое число. Тогда из теоремы 1 (необходимость) следует, что $n \notin K_{n+1}$.

Но свойства множества функций $Ł_{n+1}$ такие, что $n \in Ł_{n+1}$ Для любого $n \geq 3$. Следовательно, если $n \geq 3$ не есть простое число, то $K_{n+1} \neq Ł_{n+1}$.

Таким образом, теорема 3 доказана.

7.2. Матричная логика K'_{n+1}

Доказательство теоремы 2 оказалось довольно-таки сложным и поэтому возникает естественный вопрос о более простой характеризации простых чисел посредством логических матриц. Оказывается, это можно сделать за счет ограничения свойств функции $x \to^{K} y$.

Определим матрицу $\mathfrak{M}^{K'}_{n+1}$ следующим образом:

$$\mathfrak{M}^{K'}_{n+1} = < V_{n+1}, \sim, \to^{K'}, \{n\} > \quad (n \geq 3, n \in N),$$

где

$\sim x = n\text{-}x$,

$$x \to^{K'} y = \begin{cases} x, \text{ если } 0 < x < y < n, (x, y) \neq 1 \text{ и } (x + y) \leq n & (i_1) \\ y, \text{ если } 0 < x < y < n, (x, y) \neq 1 \text{ и } (x + y) > n & (i_2) \\ y, \text{ если } 0 < x = y < n & (ii) \\ x \to y, \text{ в остальных случаях} & (iii) \end{cases}$$

где $(x, y) \neq 1$ обозначает, что x и y не являются взаимнопростыми числами, а $x \to y$ есть импликация Лукасевича.

Таким образом, случай (i) в определении функции $x \to^{K} y$ разделился на два подслучая (i_1) и (i_2) в определении функции $x \to^{K'} y$.

Множество всех суперпозиций функций ~x и $x \to^{K'} y$ обозначим посредством K'_{n+1}.

Лемма 1'. *Пусть n есть простое число. Тогда, если x < ~x, то* $x \to^{K'} \sim x = n$.

Доказательство аналогично лемме 1.

Теорема 1'. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $n \in K'_{n+1}$.

Доказательство аналогично теореме 1.

Теорема 2'. *Для любого* $n \geq 3$ *такого, что n есть простое число,* $K'_{n+1} = Ł_{n+1}$.

Нас интересует случай

II. $Ł_{n+1} \subseteq K'_{n+1}$:

$(A')\ x \to^{1'} y = \sim((y \to^{K'} x) \to^{K'} \sim(y \to^{K'} x)) \to^{K'} (x \to^{K'} y)$

$(B')\ x \vee^{1'} y = (x \to^{1'} y) \to^{1'} y$

$(C')\ x \to^{2'} y = \sim y \to^{K'} \sim x$

$(D')\ x \to^{S} y = x \to^{2'} ((y \to^{2'} y) \to^{2'} \sim y$

$(E')\ x \to^{3'} y = \sim y \to^{S} \sim x$

$(F')\ x \to^{4'} y = ((x \to^{K'} y) \to^{3'} (\sim y \to^{K'} \sim x)) \vee^{1'}$

$((\sim y \to^{K'} \sim x) \to^{3'} (x \to^{K'} y)) = x \to y.$

Полное доказательство имеется в *[Карпенко* 1995].
Более того, удалось упростить и это выражение *[Карпенко* 1999]. Здесь приведем полное доказательство.

$(A')\ x \to^{1'} y = \sim((y \to^{K'} x) \to^{K'} \sim(y \to^{K'} x)) \to^{K'} (x \to^{K'} y)$

$(B')\ x \vee^{1'} y = (x \to^{1'} y) \to^{1'} y$

$(C')\ x \to^{2'} y = \sim(\sim x \to^{K'} \sim(x \to^{K'} x)) \to^{K'} y$

$(D')\ x \to^{3'} y = ((x \to^{K'} y) \to^{2'} (\sim y \to^{K'} \sim x)) \vee^{1'}$

$(\sim y \to^{K'} \sim x) \to^{2'} (x \to^{K'} y)) = x \to y.$

ДОКАЗАТЕЛЬСТВО.
Рассмотрение формул *{A')* и *(B')* такое же, как в теореме 2. Перейдем к формуле

$(C')\ x \to^{2'} y = \sim(\sim x \to^{K'} \sim(x \to^{K'} x)) \to^{K'} y.$

Обозначим подформулу $\sim(\sim x \to^{K'} \sim(x \to^{K'} x))$ посредством *X*.
1. y = *n*. Тогда $X \to^{K'} y = n\ (iv)$. Отсюда $x \to^{2'} y = n$.
2. $y = \sim x$ и $x < \sim x$. Тогда X=x. Отсюда $x \to^{2'} y = x \to^{K'} \sim x$ и в силу леммы $1'$, $x \to^{2'} y = n = x \to y$.
3. x = y.
3.1. x = 0. Тогда $X = \sim(n \to^{K'} \sim(0 \to^{K'} 0) = n$. Отсюда $x \to^{2'} y = n \to^{K'} 0 = 0$.
3.2. $0 < x = y < n$. Тогда $X = \sim(\sim x \to^{K'} \sim x) = x$. Отсюда $x \to^{2'} y = x \to^{K'} y = x$.
3.3. x = *n*. Тогда $x \to^{2'} y = X \to^{K'} n = n$.

Таким образом, для случая x = y функция $x \to^{2'} y$ является *идемпотентной* для всех x. Это свойство окажется необходимым при определении импликации Лукасевича x → y в формуле *(D')*.

$(D')\ x \to^{3'} y = ((x \to^{K'} y) \to^{2'} (\sim y \to^{K'} \sim x)) \vee^{1'}$

$(\sim y \to^{K'} \sim x) \to^{2'} (x \to^{K'} y)) = x \to y.$

Пусть $D'_1 = (x \to^{K'} y) \to^{2'} (\sim y \to^{K'} \sim x)$ и $D'_2 = (\sim y \to^{K'} \sim x) \to^{2'} (x \to^{K'} y)$.
1. x<y.
1.1. x = 0 и/или y = *n*. Тогда $x \to^{K'} y = n\ (iv)$. Отсюда $D'_2 = n$ $(C'.1)$ и $D'_1 \vee^{1'} n = n$. Следовательно, $x \to^{3'} y = n = x \to^{K'} y$.
1.2. (x, y) = 1 и/или $(n\text{-}y, n\text{-}x) = 1$. Тогда $x \to^{K'} y = n$ и/или $\sim y \to^{K'} \sim x = n\ (iv)$. Отсюда $D'_1 = n$ и/или $D'_2 = n\ (C'.1)$. Тогда $D'_1 \vee^{1} D'_2 = n$. Следовательно, $x \to^{3'} y = n = x \to y$.
1.3. $(x, y) \neq 1$ и $(n\text{-}y, n\text{-}x) \neq 1$. Имеем два подслучая.
1.3.1. $(x + y) < n$. Тогда в силу определения $x \to^{K'} y\ (i_1)$, $x \to^{K'} y = x$. Очевидно, если $(x + y) < n$, то $(n\text{-}y + n\text{-}x) > n$. Отсюда $\sim y \to^{K'} \sim x = \sim x\ (i_2)$. Поскольку $x < \sim x$, то $D'_1 = x \to^{2'} \sim x = n\ (C'.2)$.
Тогда $n \vee^{1} D'_2 = n$. Следовательно, $x \to^{3'} y = n = x \to y$.
1.3.2. $(x + y) > n$. Тогда в силу определения $x \to^{K'} y = x\ (i_2)$. Очевидно, если $(x + y) > n$, то $(n\text{-}y + n\text{-}x) < n$. Отсюда $\sim y \to^{K'} \sim x = \sim y\ (i_1)$. Поскольку $\sim y < y$, то $D'_2 = \sim y \to^{2'} y = n\ (C'.2)$. Тогда $D'_1 \vee^{1'} n = n$. Следовательно, $x \to^{3'} y = n = x \to y$.
2. x = y.
2.1. x< *n*/2.
2.1.1. x = 0. Тогда $x \to^{K'} y = n\ (iv)$. Отсюда $D'_2 = n\ (C'.1)$ и $D'_1 \vee^{1'} n = n$. Следовательно, $x \to^{3'} y = n = x \to y$.
2.1.2. $x \neq 0$. Тогда $x \to^{K'} y = x$ и $\sim y \to^{K'} \sim x = \sim x\ (iii)$. Отсюда $D'_1 = x \to^{2'} \sim x = n\ (C'.2)$. Тогда $n \vee^{1'} D'_2 = n$. Следовательно, $x \to^{3'} y = n = x \to y$.
2.2. x> *n*/2.
2.2.1. x = *n*. Далее также, как в (D'.2.1.1).
2.2.2. $x \neq n$. Тогда $x \to^{K'} y = x$ и $\sim y \to^{K'} \sim x = \sim x\ (iii)$. Отсюда $D'_2 = \sim x \to^{2'} x = n\ (C'.2)$. Тогда $D'_1 \vee^{1'} n = n$. Следовательно, $x \to^{3'} y = n = x \to y$.

3. x > y. Тогда $x \to^{K'} y = x \to y$ и $\sim y \to^{K'} \sim x = \sim y \to \sim x$ *(iv)*.
Поскольку $x \to y = \sim y \to \sim x$, то
$D'_1 = x \to y$ и $D'_2 = x \to y$ *(C'.3)*.
Тогда $D'_1 \vee^1 D'_2 = x \to y$. Следовательно, $x \to^{3'} y = x \to y$.
Таким образом, для любых x и y, $x \to^{3'} y = x \to y$ и, следовательно, $Ł_{n+1} \subseteq K_{n+1}$. В итоге Теорема 2' доказана.

Теорема 3'. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $Ł_{n+1} = K'_{n+1}$.

Доказательство аналогично теореме 3.
По транзитивности из теоремы 3 и теоремы 3' следует важная
Теорема 4'. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $K_{n+1} = K'_{n+1}$.
Таким образом, для случая, когда *n* есть простое число, функции $x \to^{K} y$ и $x \to^{K'} y$ совпадают. Другими словами, ограничения (i_1) и (i_2) при определении $x \to^{K'} y$ носят лишь вспомогательный характер.
Заметим, что число вхождений функции $\to^{K'}$ в суперпозицию *(A')* - (D^1) всего 167 плюс 113 вхождений функции ~. Возникает вопрос, можно ли заменить эти функции одной, т.е. построить штрих Шеффера для множества функций $\{\sim x, x \to^{K'} y\}$.
(Заметим, что не для каждого множества функций существует штрих Шеффера (см. *[Rose* 1969])

7.3. Штрих Шеффера для простых чисел

Как уже обсуждалось (см. раздел 5.2.3), для логик Лукасевича $Ł_{n+1}$ Дж.Мак-Кинси *[McKinsey* 1936] нашел штрих Шеффера, обозначенный нами как $x \to^{E} y$:

$$x \to^{E} y = x \to ((y \to J_n(y)) \to J_0(y)),$$

т.е. доказана эквивалентность следующих множеств функций

$$\{\sim x, x \to y\} = \{x \to^{E} y\}.$$

Заменим в формуле

$$(C') \; x \to^{2'} y = \sim(\sim x \to^{K'} \sim(x \to^{K'} x)) \to^{K'} y$$

все вхождения переменных на их отрицания, а затем их переименуем. Полученную функцию обозначим посредством $x \to^{s} y$, а саму формулу посредством *(S):*

$$(S) \; x \to^{s} y = \sim(y \to^{K'} \sim(\sim y \to^{K'} \sim y)) \to^{K'} \sim x.$$

Рассмотрим свойства функции $x \to^{s} y$.

1 x = 0. Тогда ~x = *n* и, следовательно, $x \to^{s} y = n$ *(iv)*.
2. 0 < x, y < *n*. Тогда в силу определения $x \to^{K'} y$ *(iii)* и закона снятия двойного отрицания, $x \to^{s} y = \sim y \to^{K'} \sim x$.
3. x = *n*.
3.1. y = *n*. Тогда $x \to^{s} y = n \to^{K'} 0 = 0$ *(iv)*.
3.2. 0<y<*n*. Тогда $x \to^{s} y = \sim y \to^{K'} 0 = y$ *(iv)*.
4. y = 0. Тогда $0 \to^{K'} \sim x = n$ *(iv)*.
Пусть S_{n+1} обозначает множество всех суперпозиций функции $x \to^{s} y$, т е. $[x \to^{s} y] = S_{n+1}$.

Теорема 5. *Для любого* $n \geq 3$ *такого, что* n *есть простое число,* $S_{n+1} = K'_{n+1}$ *[Karpenko* 1994], *[Карпенко* 1995].

ДОКАЗАТЕЛЬСТВО .
(1) $S_{n+1} \subseteq K'_{n+1}$.
Доказательством является формула

$$(S) \; x \to^{s} y = \sim(y \to^{K'} \sim(\sim y \to^{K'} \sim y)) \to^{K'} \sim x.$$

(2) $K'_{n+1} \subseteq S_{n+1}$.
Посредством функции $x \to^{s} y$ надо определить функции ~x и $x \to^{K'} y$.

$$(a) \; \sim x = x \to^{s} x.$$

1. x = 0. Тогда $x \to^{s} x = n$ *(S.1)*.
2. 0 < x < *n*. Тогда $x \to^{s} x = \sim x \to^{K'} \sim x$ *(S.2)*. Отсюда $x \to^{s} x = \sim x$ *(iii)*.
3. x = *n*. Тогда $x \to^{s} x = 0$ *(S.3.1)*.
Таким образом, для любого x, $\sim x = x \to^{s} x$.

$$(b) \; n = \sim(x \to^{s} \sim x) \to^{s} \sim(\sim x \to^{s} x).$$

Обозначим формулу *(b)* посредством *N* и пусть N_1 есть $\sim(x \to^{s} \sim x)$ и N_2 есть $\sim(\sim x \to^{s} x)$.
1. x<*n*/2.
1.1. x = 0. Тогда $N_1 = \sim(0 \to^{s} n)$. Поскольку $0 \to^{s} n = n$ *(S.1)*, то $N_1 = 0$. Отсюда $N = 0 \to^{s} N_2 = n$ *(S.1)*.
1.2. $x \neq 0$. Тогда, используя закон снятия двойного отрицания и в силу определения $x \to^{K'} y$ *(iii)*, $N_1 = \sim(x \to^{K'} \sim x)$. В силу леммы 1′, $x \to^{K'} \sim x = n$. Отсюда $N_1 = 0$ и, следовательно, $N = 0 \to^{s} N_2 = n$ *(S.1)*
2. x > *n*/2.
2.1. *x- n*. Тогда $N_2 = \sim(0 \to^{s} n) = 0$ (см. выше пункт 1.1). Отсюда $N = N_1 \to^{s} 0 = n$ *(S.4)*.

2.2. $x \neq n$. $N_2 = \sim(\sim x \to^{K'} x)$ (см. выше пункт 1.2). В силу леммы 1′, $\sim x \to^{K'} x = n$. Отсюда $N_2 = 0$ и, следовательно,

$N = N_1 \to^s 0 = n$ (*S*.4).

Таким образом, $N = n$ для любых x.

(*c*) $x \to^k y = \sim y \to^s (n \to^s \sim x) = x \to^{K'} y$.

1. x = 0. Тогда $x \to^k y = \sim y \to^s (n \to^s n) = \sim y \to^s 0$ (*S*.3.1).

Поскольку $\sim y \to^s 0 = n$ (*S*.4), то $x \to^k y = n = x \to^{K'} y$.

2. y = *n*. Тогда $x \to^k y = 0 \to^s (n \to^s \sim x) = n$ (S.1). Отсюда

$= x \to^k y = n = x \to^{K'} y$.

3. $0 < x, y < n$. Тогда $x \to^k y = \sim y \to^s \sim x$ (*S*.3.2). Отсюда

$\sim y \to^s \sim x = \sim\sim x \to^{K'} \sim\sim y$ (*S*.2). Следовательно,

$x \to^k y = n = x \to^{K'} y$. В итоге $x \to^{K'} y$ определяется следующим образом:

$$x \to^{K'} y = (y \to^s y) \to^s \{(((x \to^s (x \to^s x)) \to^s (x \to^s (x \to^s x))) \to^s$$
$$(((x \to^s x) \to^s x) \to^s ((x \to^s x) \to^s x))) \to^s (x \to^s x)\}.$$

Таким образом, функция $x \to^s y$ является штрихом Шеффера для K'_{n+1}. Поскольку каждое простое число характеризуется соответствующим предполным классом функций и только этим классам принадлежит константа *n,* т.е. имеется класс тавтологий, то исходя из этого имеет смысл говорить о *штрихе Шеффера для простых чисел.*

В силу теоремы 3, $x \to^s y$ является также штрихом Шеффера для класса функций K_{n+1}. Более того, поскольку при доказательстве теоремы 5 нигде не использовались пункты (i_1) и (i_2) в определении $x \to^{K'} y$, то непосредственное доказательство того, что $x \to^s y$ есть штрих Шеффера для K_{n+1}, в точности совпадает с уже приведенным доказательством: функция $x \to^{K'} y$ заменяется на функцию $x \to^K y$.

Дадим еще несколько определений простого числа в терминах равенства различных классов функций.

Теорема 6. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $S_{n+1} = \mathcal{L}_{n+1}$.

ДОКАЗАТЕЛЬСТВО.

I. Если $n \geq 3$ есть простое число, то $S_{n+1} = \mathcal{L}_{n+1}$. По транзитивности из теоремы 5 и теоремы 2' получаем, что $S_{n+1} = \mathcal{L}_{n+1}$.

II. Если $S_{n+1} = \mathcal{L}_{n+1}$, то $n \geq 3$ есть простое число. Докажем контрпозицию этого утверждения. Пусть $n \geq 3$ не есть простое число. Поскольку функция $x \to^3 y$ определена только посредством функций $\sim x$ и $x \to^{K'} y$, а для этих функций показано, что если *n* не есть простое число, то $n \notin K_{n+1}$ (теорема Г, *необходимость),* то, значит, $n \notin S_{n+1}$.

Но $n \in \mathcal{L}_{n+1}$ для любого $n \geq 3$. Следовательно, если *n* не есть простое число, то $S_{n+1} \neq \mathcal{L}_{n+1}$.

Таким образом, теорема 6 доказана.

По транзитивности из теоремы 6 и теоремы 3' следует

Теорема 7. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $S_{n+1} = K'_{n+1}$.

По транзитивности из теоремы 7 и теоремы 4 следует

Теорема 8. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $S_{n+1} = K_{n+1}$.

Таким образом, одной из характеризаций простого числа является свойство иметь штрих Шеффера в указанном выше смысле.

Наконец, из теоремы 6 и результата Дж.Мак-Кинси о штрихе Шеффера для $\mathcal{L}_{n+1}$ следует

Теорема 9. *Для любого* $n \geq 3$ *n есть простое число т.т.т., когда* $S_{n+1} = E_{n+1}$.

7.3.1. О формуле для простых чисел

Асимптотическое распределение простых чисел, доказываемое аналитическими методами, изложено в монографии К.Прахара *[Прахар* 1967], специально посвященной этому вопросу. О первых 50 миллионах простых чисел очень живо и интересно написано в статье Д.Цагера *[Цагер* 1984]. Здесь мы подойдем к этому вопросу с совершенно другой стороны.

Рассмотрим последовательность формул *(A^1) - (D′),* которая определяет импликацию Лукасевича x → y. Заменим в ней все вхождения ~x и $x \to^{K'} y$ на соответствующие их определения посредством штриха Шеффера $x \to^s y$. Полученную формулу обозначим посредством $(D')^s$. Можно подсчитать, что число вхождений $x \to^s y$ в $(D')^s$ будет 648 042 744 959. (Это число разлагается на следующие простые множители: $53 \times 79 \times 2887 \times 53611$.)

В свою очередь, если произведем соответствующую операцию в последовательности формул *(A)* - (*I*), то получим формулу *астрономической* длины. Тем не менее, обозначим её посредством $(I)^s$.

Обратим внимание, что формулы $(D')^s$ и $(I)^s$ конечной длины (в отличие от определения x → y посредством $x \to^\Sigma y$), хотя доказательство ведется в *n* + 1-значной логике.

Напомним, что доказательство равенства $x \to y = (D')^s$, определящего импликацию Лукасевича в матричной логике K'_{n+1} в терминах штриха Шеффера, имеет место не для всякого $n \in N$, т.е.

не для всего натурального ряда чисел, а только для *последовательности простых чисел* в натуральном ряду. В связи с этим появляется искушение заявить указанные суперпозиции, выраженные формулами $(D')^S$ и $(I)^S$, как бы косвенным образом отображающие сложность закона распределения простых чисел в натуральном ряду. На самом деле сложность заключается в свойствах функций $x \to^K y$ и $x \to^{K'} y$. И как мы увидим в следующей главе, наиболее сложной окажется характеризация четных чисел. Однако логики K_{n+1} и K'_{n+1} указывают нам формулы, которые в алгебрологическом смысле порождают все простые числа.

Попытки найти *формулу,* с помощью которой вычислялись бы (или порождались) все простые числа, имеют длинную историю. Известно, что не существует полинома без констант, который принимает значения только в простых числах (см., например, *[Ribenboim* 1996]). Тем не менее найдены полиномы от многих переменных с целыми коэффициентами, такие, что множество простых чисел совпадает с множеством натуральных значений, приписываемых переменным. Наилучший результат здесь принадлежит Ю.В.Матиясевичу *[Матиясевич* 1977], который нашел полином из 10 переменных.

В связи с этим продолжается дискуссия о том, что считать формулой для простых чисел (см. в особенности *[Wilf* 1982], а также *[Ribenboim* 1997]). Если можно не использовать знак суммы, факториал, минимизирующую функцию в наших формулах, тогда в действительности существуют формулы для простых чисел. Разумная интерпретация слова «формула» выглядит очень просто: «Машина Тьюринга, которая останавливается на всех входах». При такой интерпретации определенно имеются останавливающиеся машины Тьюринга, которые вычисляют *n*-е простое число. Однако никто не знает как вычислить это *n*-е простое число в полиномиальное время, т.е. за *log n.* Это остается открытой проблемой.

В некотором смысле, а именно в алгебро-логическом, все наши тавтологии (число которых счетно) логик K_{n+1} и K'_{n+1} являются формулами для простых чисел. Видимо, наиболее короткая формула, обозначенная посредством *U,* как раз и использована нами при доказательстве теоремы 1 (достаточность). Эта формула является тавтологией в K_{n+1} при всех натуральных значениях *x* и *y* т.т.т., когда *n* есть простое число. Алгебро-логические операции $\sim x$ и $x \to^K y$ можно заменить на одну-единственную - штрих Шеффера $x \to^s y$. Правда, в целом эта функция имеет довольно-таки сложную природу и кроме операций арифметического сложения, вычитания и *min(x, y)* включает еще проверку чисел на взаимную

простоту. А эта операция уже не является полиномиальной. Таким же свойством обладают и характеризующие теоремы 3, 3^1, 4', 6, 7, 8 и 9. Однако, оказывается, от алгебро-логической формулы, «выражающей» простые числа, можно перейти к такой же формуле, но порождающей *классы* простых чисел. А это уже нечто совсем другое.

7.4. Закон порождения классов простых чисел

Поскольку теорема 4 говорит о том, что для случая, когда $n \geq 3$, такого, что *n* есть простое число, ограничения (i_1) и (i_2) при определении функции $x \to^{K'} y$ излишни, то можно заменить в теореме 2' (II) функцию $x \to^{K'} y$ на функцию $x \to^K y$. Обозначим новую последовательность формул посредством $(A^*) - (D^*)$. Нетрудно показать, что тогда формула (D*):

$$x \to^* y = ((x \to^K y) \to^2 (\sim y \to^K \sim x)) \vee^1$$

$$(\sim y \to^K \sim x) \to^2 (x \to^K y))$$

определяет импликацию Лукасевича x →y только для первых пяти нечетных чисел: 3, 5, 7, 11 и 13. Однако если x < y и $(x, y) \neq 1$, $(n\text{-}y, n\text{-}x) \neq 1$, то в общем случае $x \to^* y \neq x \to y$. Например, пусть *n* =17, x = 2 и y= 12. Тогда

$x \to^K y = 12$, $\sim y \to^K \sim x = 15$, $12 \to^2 15 = 15$,

$15 \to^2 12 = 14$, $15 \vee^1 14 = 15$. Таким образом, $x \to^* y = 15$, в то время как $x \to y = 17$. Можно показать, что итерация $\mathcal{D}_i$ $(i = 1, 2, 3, \ldots)$ формулы (D*) будет задавать классы простых чисел, для которых формула $\mathcal{D}_i$ определяет x →y. Пусть

$$A_0 = ((x \to^K y) \to^2 (\sim y \to^K \sim x)) \text{ и}$$

$$B_0 = ((\sim y \to^K \sim x) \to^2 ((x \to^K y)).$$

Тогда

$$\mathcal{D}_0 = A_0 \vee^1 B_0,$$

$$\mathcal{D}_1 = (A_0 \to^2 B_0) \vee^1 (B_0 \to^2 A_0),$$

$$\mathcal{D}_2 = ((A_0 \to^2 B_0) \to^2 (B_0 \to^2 A_0)) \vee^1 ((B_0 \to^2 A_0) \to^2 (A_0 \to^2 B_0))$$

и так далее.

Таким образом, смысл итерации состоит в том, что берется исходная формула $\mathcal{D}_0$, в ней осуществляется операция замены дизъюнкции $\vee^1$ на импликацию $\to^2$ (эту операцию обозначим посредством:

$[\rightarrow^2 / \vee^1])$, затем над полученной формулой производится операция обращения (REV), т.е. импликация записывается в обратную сторону, и, наконец, обе формулы соединяются дизъюнктивно. Заметим, что дизъюнкцию $\vee^1$ в силу формулы

$$(E)\ \ x \vee y = (x \vee^K y) \vee^1 (y \vee^K x) = max(x, y)$$

можно заменить на обычную дизъюнкцию $\vee$, что упрощает вычисления. Тогда в общем случае запись итерации выглядит так:

$$\mathcal{D}_i = ([\rightarrow^2/ \vee]\mathcal{D}_{i-1}) \vee (\mathrm{REV}([\rightarrow^2/ \vee]\mathcal{D}_{i-1})).$$

Обозначим посредством P_i класс простых чисел, при которых $\mathcal{D}_i = x \rightarrow y$. В силу идемпотентности операции $\rightarrow^2$ замена дизъюнкции $\vee$ на $\rightarrow^2$ сохраняет значения обоих членов дизъюнкции $\mathcal{D}_{i-1}$, когда они равны, при переходе к $\mathcal{D}_i$. Отсюда следует, что класс P_{i-1} содержится в P_i. Тогда имеем

$$P_0 = \{3, 5, 7, 11, 13\},$$

$$P_1 = P_0 \cup \{17, 19\},$$

$$P_2 = P_1 \cup \{23, 29, 31, 41, 43, 53, 59, 61\}.$$

С помощью компьютерной программы, написанной В.И.Шалаком в 1995 г., можно вычислить другие P_i. Например,

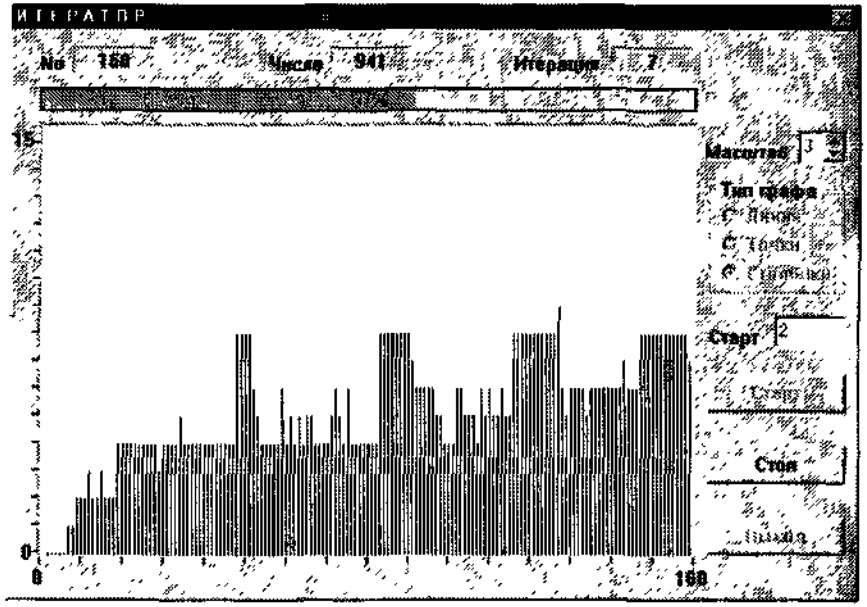
$$P_3 = P_2 \cup \{37, 47, 109\}.$$

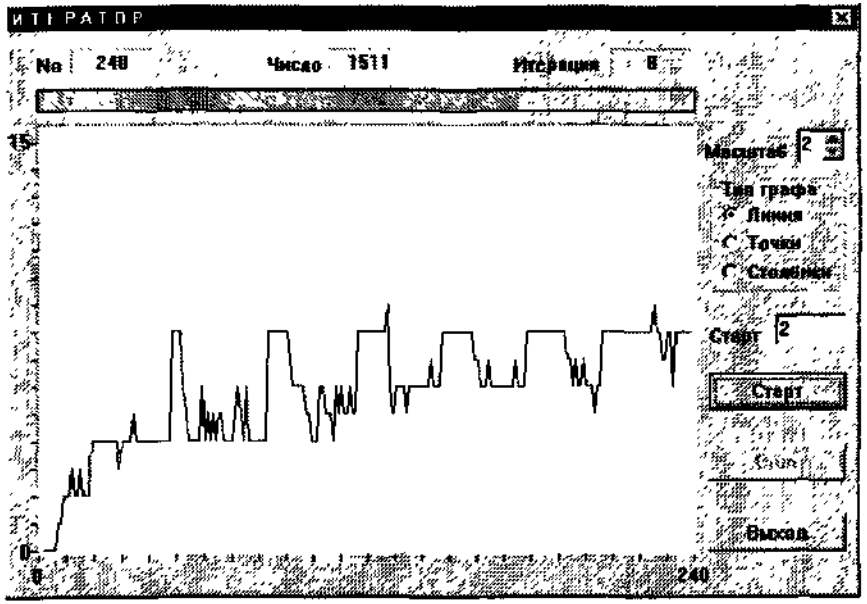
Класс P_4 содержит новые простые числа в количестве 51; класс P_5 содержит 21 новое простое число.

Таким образом, для каждого n импликации Лукасевича $x \rightarrow y$ соответствует свой новый класс простых чисел. В результате получаем разбиение множества простых чисел на классы эквивалентности относительно числа итераций. Это разбиение напрямую связано со свойствами импликации Лукасевича.

По существу формула $\mathcal{D}_i$ является *законом порождения простых чисел [Карпенко* 1995], *[Karpenko* 1996], *[Karpenko* 1997], а точнее, законом порождения классов простых чисел. Подчеркнем, что в силу теоремы 8 этот закон может описываться итерацией только одной-единственной функции, а именно штриха Шеффера $x \rightarrow^5 y$.

Для наглядности приведем два графика для определенного числа простых чисел. По вертикали показано число итераций, по горизонтали - простые числа.





Заметим, что вычислять простые числа по формуле $\mathcal{D}_i$ весьма громоздко и не эффективно, тем более, что для этого существует огромное число различных алгоритмов (см. обзор *[Василенко* 1988]). В данном случае нас интересует разбиение простых чисел на классы P_i И программа В.И.Шалака выполняет именно эту задачу, т.е. $\mathcal{D}_i$ вычисляется только для случаев, когда $n = p$. Поэтому введем функцию i, которая по каждому простому числу дает число итераций $i(p)$.
Можно упростить исходную формулу (D*), заменив в ней вхождения функции $x \to^2 y$ на x →y и при этом рассматривая только случай $0 < x < y < n$. Правда, на компьютерный процесс вычисления это влияет незначительно. Значение имело бы, если максимально ограничить процесс перебора случаев, когда
$(x, y) \neq 1$ и $(n\text{-}y, n\text{-}x) \neq 1$, рост которых экспоненциален. То есть проблема состоит в поиске наиболее короткого *пути* (может быть единственного !) от $\mathcal{D}_i$ к $\mathcal{D}_j$, где $i < j$.
Но это весьма трудоемкая задача.
Мы показали, что итерация $\mathcal{D}_i$ порождает классы простых чисел, но встает принципиальный вопрос: порождаются ли *все* простые числа?
На этот вопрос дает ответ
Теорема 10. *Каждое простое число (кроме 2) содержится в некотором классе* P_i *[Карпенко* 1997].

Пусть x < y и $(x, y) \neq 1$, $(n\text{-}y, n\text{-}x) \neq 1$. Свойства функций $x \to^2 y$ и $x \vee y$ таковы, что с ростом числа i в $\mathcal{D}_i$, т. е. с увеличением числа итераций, исходные значения x и y также растут. Это следует из того, что функция $x \to^2 y$ для случая $x < y$ выбирает большее значение, $x \vee y$ есть *max(xy),* а для случая $x > y$ функция $x \to^2 y$ есть функция Лукасевича x →y, т. е. $x \to^2 y = p\text{-}x+y$, а значит y растет. Рост значений x и y *не* может продолжаться бесконечно (в силу конечного числа значений, определяемых числом *p)* и не может уменьшаться, как только что было показано, но может «зацикливаться», т.е. начиная с некоторого i при всех дальнейших итерациях $\mathcal{D}_i = z$, где $z \neq p$. Это может произойти, если

$(a)\ x \to^2 y = x$ при $x > y$

и если также выполняется условие

$(b)\ (x, y) \neq 1$, тогда $y \to^2 x = x$.

Отсюда следует, что существуют такие p, которые не попадают ни в один класс P_i. Покажем, что это не так, поскольку условия *(a)* и (6) несовместимы между собой.
В общем случае условие *(a)* имеет место, когда $x = p\text{-}k$ *и* $y = p\text{-}2k$. Тогда $x \to^2 y = (p\text{-}k \to^2 p\text{-}2k) = p\text{-}(p\text{-}k)+p\text{-}2k = p\text{-}k$.
Покажем, что $(p\text{-}k, p\text{-}2k) = 1$, т.е. $p\text{-}k$, и $p\text{-}2k$ взаимно-простые числа.
Допустим обратное, т.е. $d \mid p\text{-}k$ и $d \mid p\text{-}2k$, где $d \neq 1$. Тогда из II(с.о.д.) следует, что $d \mid ((p\text{-}k)\text{-}(p\text{-}2k))$, т.е. $d \mid k$. Поскольку $d \mid p\text{-}k$, то из I(с.о.д) следует, что $d \mid (p\text{-}k+k)$, т.е. $d \mid p$. Последнее противоречит свойству быть простым числом. Таким образом, $(p\text{-}k, p\text{-}2k) = 1$.
Тогда $y \to^2 x \neq x$ и, значит, при выполнении условия *(a)* условие *(b)* не выполняется. Отсюда следует, что для любого нечетного простого числа p за конечное число итераций i найдется класс P_i.
Таким образом, теорема 10 доказана.
Гипотеза. *Каждый класс* P_i *конечен.*
Обратим внимание на «нерегулярность» заполнения классов P_i. Так, простое число 223 попадает только в класс P_8, тогда как уже класс P_5 заканчивается простым числом 757. С другой стороны, имеем следующую последовательность мощностей классов P_i: 5, 2, 8, 3, 51, 21, 54, 19, ... Класс P_8 содержит уже больше 250 простых чисел. Все это несомненно отражает невероятную сложность *закона* распределения простых чисел в натуральном ряду, про который Эйлер сказал: «...у нас есть основания считать, что это тайна, в которую человеческий разум никогда не проникнет» (цит. по: *[Ayoub* 1963]).

8. Характеризация классов натуральных чисел логическими матрицами Лукасевича

Известно, что для диофантовых множеств [Матиясевич 1972] существуют полиномы от многих переменных с целыми коэффициентами, такие, что множество всех натуральных значений, принимаемых ими при натуральных значениях переменных, есть в точности исходные диофантовы множества. Так был найден полином, множеством значений которого является множество всех простых чисел [Матиясевич 1977]. По некоторой аналогии с этим можно охарактеризовать разные подмножества множества истинностных значений $V_{n+1} = \{0, 1, 2, \ldots, n\}$, используя логические матрицы или, что то же самое, алгебры определенного вида, а именно простые числа (что сделано в предыдущей главе), степени простых чисел, четные числа, нечетные числа. Обращает на себя внимание сложность характеризации четных чисел. Предпринимается попытка связать это со знаменитой проблемой Гольдбаха о разложении четных чисел на сумму двух простых.

8.1. Простые числа

В предыдущей главе был охарактеризован класс простых чисел следующим образом: построена матричная логика K'_{n+1} такая, что K'_{n+1} имеет класс тавтологий т.т.т., когда *n* есть простое число. Более того, было показано, что $K'_{n+1} = Ł_{n+1}$ т.т.т., когда *n* есть простое число, где $Ł_{n+1}$ есть *n*+1-значная логика Лукасевича *[Lukasiewicz & Tarski* 1930].

Решающим пунктом при доказательстве функциональной эквивалентности множества функций K'_{n+1} и $Ł_{n+1}$ явилось нахождение алгебро-логического полинома, состоящего из вхождений ~ и $\to^{K'}$, такого, что посредством него определяется импликация Лукасевича → (см. выше раздел 7). Для сравнения с последующими характеризациями приведем еще раз определение логической матрицы $\mathfrak{M}^{K'}_{n+1}$, её свойства и сам этот полином:

$$\mathfrak{M}^{K'}_{n+1} = < V_{n+1}, \sim, \to^{K'}, \{n\} > \quad (n \geq 3, n \in N),$$

где

$$\sim x = n\text{-}x,$$

$$x \to^{K'} y = \begin{cases} x, \text{ если } 0 < x < y < n, (x, y) \neq 1 \text{ и } (x + y) \leq n & (i_1) \\ y, \text{ если } 0 < x < y < n, (x, y) \neq 1 \text{ и } (x + y) > n & (i_2) \\ y, \text{ если } 0 < x = y < n & (ii) \\ x \to y, \text{ в остальных случаях} & (iii), \end{cases}$$

где $(x, y) \neq 1$ обозначает, что *x* и *y* не являются взаимнопростыми числами, а *x* →y есть импликация Лукасевича.

Множество всех суперпозиций функций ~*x* и $x \to^{K'} y$ обозначим посредством K'_{n+1}.

Лемма 1'. *Пусть n есть простое число. Тогда, если* $x < \sim x$, *то* $x \to^{K'} \sim x = n$.

Доказательство аналогично лемме 1 (гл. 7).

Теорема 1'. *Для любого* $n \geq 3$ *n есть простое число т.т т., когда* $n \in K'_{n+1}$.

Доказательство аналогично теореме 1 (гл. 7).

Теорема2′. *Для любого* $n \geq 3$ *такого, что n есть простое число,* $K'_{n+1} = Ł_{n+1}$.

Нас интересует наличие следующего полинома.

II. $Ł_{n+1} \subseteq K'_{n+1}$:

$$(A') \quad x \to^{1'} y = \sim((y \to^{K'} x) \to^{K'} \sim(y \to^{K'} x)) \to^{K'} (x \to^{K'} y)$$

$$(B') \quad x \vee^{1'} y = (x \to^{1'} y) \to^{1'} y$$

$$(C') \quad x \to^{2'} y = \sim(\sim x \to^{K'} \sim(x \to^{K'} x)) \to^{K'} y$$

$$(D') \quad x \to^{3'} y = ((x \to^{K'} y) \to^{2'} (\sim y \to^{K'} \sim x)) \vee^{1'}$$

$$(\sim y \to^{K'} \sim x) \to^{2'} (x \to^{K'} y)) = x \to y.$$

Из теоремы *1′*, теоремы *2′* и свойств $Ł_{n+1}$ следует

Теорема 3'. *Для любого* $n \geq 3$ *n есть простое число тогда и только тогда, когда* $K'_{n+1} = Ł_{n+1}$.

Теперь перейдем к подобной характеризации других подмножеств натурального ряда.

8.2. Степень простого числа

При обсуждении результатов о характеризации простых чисел во время доклада в ВИНИТИ 18 декабря 1981 г. В.К.Финн поставил

проблему о погружении степени простого числа в n+1-значные логики Лукасевича, т.е. когда $n = p^{\beta}$, где p - простое число, а β- натуральное число, $\beta \geq 1$. В связи с этим обратим внимание на следующий результат в *[Бочвар & Финн* 1972, теорема 4]. *каждая тождественно неравная* 0 *функция* $f \in Ł_{n+1}$ *имеет I-J-совершенную дизъюнктивную нормальную форму т.т.т., когда* $n = p^{\beta}$. (Определение I- и *J*-функций см. в разделе 5.3). Отсюда возникает идея о естественном обобщении логики K_{n+1} на случай $n = p^{\beta}$.

Проблема Финна была специально сформулирована в *[Карпенко* 1995,], а в *[Карпенко* 1998] была построена матричная логика F_{n+1} и высказана гипотеза, что данная проблема имеет решение. В *[Карпенко* 1999] указан искомый полином без доказательства, которое приведем здесь с более четким определением функции $x \to^{F} y$.

Для решения вышеупомянутой проблемы предлагается ограничить условие, когда $(x, y) \neq 1$ в пунктах (i_1) и (i_2) определения фунции $x \to^{K'} y$: $(i_1^F \; u \; i_2^F)$ *среди общих делителей x и y не существует такого делителя d, отличного от 1, который сам или его степень являются единственными делителелями n.*

В противном случае $x \to^{K'} y = n$. В свою очередь, это ограничение распространяется также на случай, когда $0 < x = y < n$. При этом $x + y = n$. Это позволяет охарактеризовать числа вида 2^n. Данное ограничение обозначим посредством (ii^F).

Так определенную функцию обозначим посредством $x \to^{F} y$, а множество всех суперпозиций функций ~x и $x \to^{F} y$ обозначим посредством F_{n+1}.

Лемма 1^F. *Пусть n есть степень простого числа, т.е.* $n = p^{\beta}$. *Тогда, если x < ~x, то* $x \to^{F} \sim x = n$.

Если (x, *n*-x) = 1, то $x \to^{F} \sim x = n$.

Пусть НОД(x, *n-x) = d,* где *d* отлично от 1 и *n.* Так как *d* является простым числом (см. *[Бухштаб* 1960]), то допустим, что *d = p*,* где *p** отлично от *p* в условии леммы. Тогда из I(с.о.д.) следует, что $d^* | (x+n-x)$, т.е. $d^* | n$, что противоречит условию леммы. (I(с.о.д.) если числа *x* и *y* делятся на *z,* то их сумма *x + y* делится на *z.)*

Отсюда *p* = p* и, следовательно, $x \to^{F} \sim x = n$.

Теорема 4. *Для любого* $n \geq 2$ *n есть степень простого числа* p^{β} *т.тт., когда* $p^{\beta} \in F_{n+1}$.

Доказательство аналогично теореме 1 (гл. 7).

Теорема 5. *Для любого* $n \geq 3$, *такого, что n есть степень простого числа,* $F_{n+1} = Ł_{n+1}$.

I. $F_{n+1} \subseteq Ł_{n+1}$.

Доказательство следует из критерия Р.Мак-Нотона *[McNaughton* 1951] об определимости функций в матрицах Лукасевича. Напомним: функция $f(\frac{x_1}{n}, \ldots, \frac{x_k}{n}) = \frac{x}{n}$ определима в матрице для $Ł_{n+1}$, если и только если НОД есть $(x_1, \ldots, x_k, n)$ делитель x (НОД - наибольший общий делитель).

II. $Ł_{n+1} \subseteq F_{n+1}$.

(A^F) $x \to^1 y = \sim((y \to^F x) \to^F \sim(y \to^F x)) \to^F (x \to^F y)$

(B^F) $x \vee^1 y = (x \to^1 y) \to^1 y$

(C^F) $x \to^2 y = \sim((\sim x \to^F \sim x) \to^F \sim(x \to^F x)) \to^F y$

(D^F) $x \to^3 y = ((x \to^F y) \to^2 (\sim y \to^F \sim x)) \vee^1$
$((\sim y \to^F \sim x) \to^2 (x \to^F y)) = x \to y.$

В формуле (A^F) нас интересует случай
2. x = y.
2.1. (x, *n)* - 1. Тогда доказательство аналогично пункту *(A.2) в* теореме 2 из раздела 7.1 с соответствующей заменой леммы 1 на лемму 1^F.
2.2. $(x, n) \neq 1$.

2.2.1. $0 < x < n/2$. Тогда $x \to^F \sim x = n$ $(i_1^F \; u \; i_2^F)$. В силу определения $x \to^F y$ *(iii),* $x \to^F y = n$. Отсюда $\sim(x \to^F \sim x) = 0$ и, следовательно, $x \to^1 y = n$.

2.2.2. $n/2 < x < n$. Тогда $x \to^1 y = (2x - n) \to^F x$, где $(2x - n) < n$ (см. пункт (A.2.2) в теореме 2 из раздела 7.1). Отсюда в силу определения $x \to^F y$ $(i_1^F \; u \; i_2^F)$, $\sim(x \to^F \sim x) \to^F x = n$. Следовательно, $x \to^1 y = n$.

2.3. $(x, n) \neq 1$ и $x + x = n$. Тогда $x \to^F y = n$ (ii^F)и, следовательно, $x \to^1 y = n$.

Таким образом, $x \to^F y$ всегда принимает значение *n,* когда x= y, т.е. как и импликация Лукасевича $x \to y$.

Обратим внимание, что формула (C′) из теоремы (2'), заменена здесь на формулу *(C^F).* Это позволяет верифицировать случай $n = 2^n$. Проверим формулу

(C^F) $x \to^2 y = \sim((\sim x \to^F \sim x) \to^F \sim(x \to^F x)) \to^F y.$

Здесь надо рассмотреть пункт 3, учитывая ограничение (ii^F).

Обозначим подформулу $\sim((\sim x \to^F \sim x) \to^F \sim(x \to^F x))$ посредством X^F.

3. x = y.
3.1. x = 0. Тогда $X^F = \sim((n \to^F n) \to^F \sim(0 \to^F 0)) = n$. Отсюда $x \to^2 y = n \to^F 0 = 0$.
3.2. 0<x = y <*n*.
3.2.1. $x + y \neq n$. Тогда $X^F = \sim(\sim x \to^F \sim x) = x$. Отсюда $x \to^2 y = x \to^F y = x$.
3.2.2. x + y = *n*.
3.2.2.1. Пусть ограничение (ii^F) не выполняется. Тогда, как в случае 3.2.1.
3.2.2.2. Пусть ограничение (ii^F) выполняется. Тогда $X^F = \sim(n \to^F (\sim n)) = n$. Отсюда $x \to^2 y = n \to^F y = y$. Заметим, что если бы оставалась формула (C′), то имели бы:
$X^F = \sim(\sim x \to^F 0) = \sim x$. Отсюда $x \to^2 y = \sim x \to^F x$, т.е. функция $x \to^2 y$ не являлась бы идемпотентной.
3.3. x = *n*. Тогда $x \to^2 y = X^F \to^F n = n$.
Из теоремы 4, теоремы 5 и свойств $Ł_{n+1}$ следует
Теорема 6. *Для любого* $n \geq 3$ *n есть степень простого числа тогда и только тогда, когда* $F_{n+1} = Ł_{n+1}$.
См. доказательство теоремы 3 в гл. 7.

8.3. Чётные числа

В *[Карпенко* 1998] была поставлена проблема о характеризации четных чисел посредством логических матриц Лукасевича в смысле леммы 1 из гл. 7. Вопрос о соответствующем алгебро-логическом полиноме даже не ставился. Проблема эта давно интересовала Карпенко, но непонятно было, как подойти к её решению. В *[Карпенко* 1999] было предложено следующее решение, которое с некоторыми уточнениями и полным доказательством изложим здесь.
Рассмотрим логическую матрицу

$$\mathfrak{M}^e_{n+1} = < V_{n+1}, \sim, \to^e, \{n\} > \quad (n \geq 3, n \in N),$$

где

$$\sim x = n\text{-}x,$$

$$x \to^e y = \begin{cases} x, \text{ если } 0 < x < y < n \text{ и } x + y < n & (i) \\ x, \text{ если } 0 < x < y < n,\ x + y = n \text{ и } x, y \text{ различной четности} & (ii) \\ y, \text{ если } 0 < x < y < n \text{ и } x + y > n & (iii) \\ y, \text{ если } 0 < x = y < n \text{ и } x + y \neq n & (iv) \\ x \to y, \text{ в остальных случаях} & (v) \end{cases}$$

Множество всех суперпозиций функций ~x и $x \to^e y$ обозначим посредством $\mathcal{E}_{n+1}$.
Лемма 1^e. *Пусть n есть чётное число. Тогда, если* $x < \sim x$, *то* $x \to^e \sim x = n$.
Поскольку *n* есть чётное число, то x и ~x являются числами одинаковой четности и в сумме дают *n*. Отсюда пункт (*ii*) в определении $x \to^e y$ не выполняется и, значит, $x \to^e y = n$ *(v)*.
Теорема 6. *Для любого* $n \geq 2$ *n есть четное число т.т.т., когда* $n \in \mathcal{E}_{n+1}$.
1. ***Достаточность***. Если *n* есть чётное число, то $n \in \mathcal{E}_{n+1}$. Пусть *n* есть четное число. Тогда существует такая формула U (см. теорему 1 в гл. 7):

$$\sim((x \to^e y) \to^e \sim(x \to^e y)) \to^e (\sim(x \to^e y) \to^e (x \to^e y)),$$

что U = *n*. Рассмотрим подформулы $U_1 = (x \to^e y) \to^e \sim(x \to^e y)$ и $U_2 = \sim(x \to^e y) \to^e (x \to^e y)$.
Очевидно, что для тех случаев, когда $x \to^K y = 0$ или $x \to^K y = n$, $U = n$. Пусть $x \to^e y < n/2$. Тогда в силу леммы 1^e, $U_1 = n$, $\sim U_1 = 0$ и, следовательно, $U = 0 \to^e U_2$. Пусть $x \to^e y = n/2$ или $x \to^e y = n$. Тогда в силу леммы 1^e , $U_2 = n$ и, следовательно, $U = \sim U_1 \to^e n = n$.
2. ***Необходимость***. Если $n \in \mathcal{E}_{n+1}$, то *n* есть четное число. Докажем контрпозицию этого утверждения: если *n* не есть четное число, то $n \notin \mathcal{E}_{n+1}$. Пусть *n* есть нечётное число и 0 < x, y < *n*. Для того чтобы $x \to^e y$ принимало значение *n*, x и y должны быть одинаковой четности и в сумме давать *n*. Но тогда *n* будет чётным числом, что противоречит нашему допущению о нечётности *n*.
Теорема 7. *Для любого* $n \geq 2$ *такого, что n есть чётное число,* $\mathcal{E}_{n+1} = Ł_{n+1}$.
I. $\mathcal{E}_{n+1} \subseteq Ł_{n+1}$.
Доказательство следует из критерия Р.Мак-Нотона об определимости функций в матрицах Лукасевича.

II. $Ł_{n+1} \subseteq E_{n+1}$.

(A^e) $x \to^1 y = \sim((y \to^e x) \to^e \sim(y \to^e x)) \to^e (x \to^e y)$,

(B^e) $x \to^2 y = \sim((y \to^e x) \to^e \sim(y \to^e x)) \to^e (\sim y \to^1 \sim x)$,

(C^e) $x \vee^1 y = (x \to^2 y) \to^2 y$,

(D^e) $x \to^3 y = \sim((\sim x \to^e \sim x) \to^e \sim(x \to^e x)) \to^e y$

(E^e) $x \to^4 y = ((x \to^e y) \to^3 (\sim y \to^e \sim x)) \vee^1$
$((\sim y \to^e \sim x) \to^3 (x \to^e y)) = x \to y$.

В формуле (A^e) нас интересует условие
2. $0<x = y<n$ и $x> n/2$. Тогда в общем случае $(2x - n) \to^e x \neq n$.
Например, пусть $n = 10$ и $x = 6$. Тогда $x \to^1 y = 2 \to^e 6 = 2$.
Поэтому для того чтобы $x \to^1 y = n$ при любых $x = y$, вводится формула (B^e). Тогда, при этом условии, $\sim y \to^1 \sim x = n$ и, следовательно, $x \to^2 y = n$.
Из теоремы 6, теоремы 7 и свойств $Ł_{n+1}$ следует
Теорема 8. *Для любого $n \geq 2$ n есть чётное число тогда и только тогда, когда $E_{n+1} = Ł_{n+1}$.*

8.4. Нечётные числа

Наконец, дадим характеризацию нечётных чисел (см. *[Карпенко* 1999]. Это легко сделать, учитывая, что логические матрицы Лукасевича имеют неподвижную точку, относительно отрицания ~x. Пусть

$\mathfrak{M}^{\circ}_{n+1} = < V_{n+1}, \sim, \to^{\circ}, \{n\} >$,

где

$\sim x = n\text{-}x$,

$$x \to^{\circ} y = \begin{cases} n, \text{ если } x < y & (i) \\ y, \text{ если } 0 < x = y < n \text{ и } x + y = n & (ii) \\ x \to y, \text{ в остальных случаях} & (iii), \end{cases}$$

Множество всех суперпозиций функций ~x и $x \to^{\circ} y$ обозначим посредством O_{n+1}.
Теорема 9. *Для любого $n \geq 2$ есть нечетное число т.т.т., когда $n \in O_{n+1}$.*

(1) *Достаточность.* Если *n* есть нечетное число, то $n \in O_{n+1}$. Пусть *n* есть нечетное число. Тогда существует такая формула U°, а именно

$(x \to^{\circ} y) \to^{\circ} (x \to^{\circ} y)$,

что U° = *n*.
1. Пусть $(x \to^{\circ} y) = 0$ и/или $(x \to^{\circ} y) = n$. Тогда U° = *n*.
2. Пусть $(x \to^{\circ} y) = z$ и $0 < z < n$. В силу нашего допущения о нечетности и, $z+z \neq n$. Отсюда $z \to^{\circ} z = n$. Следовательно, U° = *n*.
(2) *Необходимость.* Если $n \in O_{n+1}$, то *n* есть нечетное число. Докажем контрпозицию этого утверждения: если *n* не есть нечетное число, то $n \notin O_{n+1}$. Пусть *n* есть четное число. Посредством D обозначим множество элементов вида x+x = *n*. Покажем, что множество D замкнуто относительно ~x и $x \to^{\circ} y$.
Пусть $x \in D$. Тогда ~x = x, т.е. относительно отрицания данная матрица имеет неподвижную точку. Следовательно $\sim x \in D$.
Пусть x = y и x+y = *n*. Тогда из определения $x \to^{\circ} y$ следует, что $x \to^{\circ} x = x$, т.е. для этого случая операция $\to^{\circ}$ идемпотентна. Следовательно $x \to^{\circ} x \in D$.
Теорема 10. *Для любого $n \geq 2$, такого, что n есть нечетное число, $O_{n+1} = Ł_{n+1}$*
I. $O_{n+1} \subseteq Ł_{n+1}$.
Доказательство следует из критерия Мак-Нотона о выразимости функций в матрицах Лукасевича.
II. $Ł_{n+1} \subseteq O_{n+1}$.

$x \to y = x \to^{\circ} y$.

Из теоремы 9, теоремы 10 и свойств $Ł_{n+1}$ следует
Теорема 11. *Для любого $n \geq 2$ n есть нечётное число тогда и только тогда, когда $O_{n+1} = Ł_{n+1}$.*

Несколько замечаний (в том числе и о проблеме Гольдбаха)

Конечно, учитывая результат Мак-Кинси о штрихе Шеффера для $Ł_{n+1}$ (см. раздел 5.2.3) и построение штриха Шеффера для простых чисел (см. раздел 7.3), есть все основания считать, что существуют соответствующие штрихи Шеффера для степени простых чисел, четных чисел и нечетных чисел. Вопрос этот чисто технический. Более важной темой является вопрос о погружении арифметики в n+1-значные логики Лукасевича $Ł_{n+1}$. Именно так и был поставлен вопрос

В.К.Финном в связи с характбризацией степени простого числа: какая часть арифметики может быть погружена в $Ł_{n+1}$?
На особые размышления наводит характеризация матрицами Лукасевича четных чисел. Загадочным выглядит уже то, что из всех рассмотренных подмножеств натурального ряда эта проблема оказалась наиболее сложной. Обратим внимание, что при доказательстве соответствующих теорем здесь неявным образом использовался элементарный арифметический факт о разложении четных чисел на сумму двух четных или сумму двух нечетных чисел. Это подводит нас к другому разложению четных чисел, к проблеме Гольдбаха о разложении четных чисел ≥ 4 на сумму двух простых (см., например монографию *[Wang* 1984]). Если предположение Гольдбаха верно, тогда для каждого числа *m* найдутся простые числа *p* и *q* такие, что

$$\varphi(p) + \varphi(q) = 2m,$$

где φ(*x*) есть функция Эйлера *[Guy* 1994].
Эдмунд Ландау на международном конгрессе математиков в Кембридже в 1912 г. заявил, что проблема Гольдбаха недоступна для современного состояния науки. Недоступна она и сейчас. Как заметил Г.Харди *[Hardy* 1999]: «Сравнительно легко сделать умное предположение; в действительности имеются теоремы, подобные "теореме Гольдбаха", которые никогда не были доказаны и которые любой дурак может понять».
(Может быть именно поэтому за решение данной проблемы Британским издательством «Faber and Faber» в 2000 г. объявлена премия в 1 000 000 $ Информацию об этом и условия конкурса можно найти на сайте http.//www.mscs.dal.ca/~dilcher/Goldbach/index html.)
В 1885 г. A.Desboves верифицировал предположение Гольдбаха для 10000. С появлением мощных и сверхмощных компьютеров границы проверяемых чисел стали быстро расширяться и в 1998 г. J.-M.Deshouillers, H.J.J. te Riele и Y. Saouter установили верифицируемость предположения Гольдбаха до 4×10^{14}.
И все-таки существенного продвижения в общем решении этой проблемы не видно. Тем не менее, используя методы, изложенные в настоящей главе, можно наметить следующий алгебро-логический подход к рассмотрению этой проблемы (см. *[Карпенко* 1989], *[Karpenko* 1989]). Заметим, что в обеих указанных работах в качестве исходного условия выдвигалось построение матричной логики для четных чисел, т.е. логики $Ł_{n+1}$. Далее стратегия выглядит так (в качестве примера).
На пункт (*ii*) в определении функции $x \to^{\varepsilon} y$ накладывается

следующее ограничение: *x и y не являются простыми числами.*
Аналогичное ограничение налагается на пункт (*i*v). В противном случае, $x \to^{e} y = n$. Так определенную функцию обозначим посредством $x \to^{G} y$. Однако в этом случае при доказательстве аналога теоремы 6 (достаточность) возникают серьезные затруднения, в то время как условие необходимости доказывается слишком тривиально. На самом деле, при определении функции подобной $x \to^{G} y$ должен соблюдаться некоторый баланс, позволяющий представить константу типа формулы U (или совсем другую) и в то же время соблюсти условие необходимости. Это как раз и удавалось достигнуть при характеризации указанных подмножеств натурального ряда.
В завершение этой главы можно теперь со всей определенностью сказать, что *n*+1-значные логики Лукасевича $Ł_{n+1}$ имеют *чисто* арифметическую интерпретацию и поэтому все, что изложено в гл. 4, указывает на некоторые общие факты, свойственные широкому классу пропозициональных логических систем, или на изощренность человеческого ума.

Таблицы чисел

Таблица 1

Степени кардинальной полноты $\gamma(Ł_n)$

(см. раздел 3.2.2)

n =	1										
n =	2	=>	γ(n) =	2	n =	51	=>	γ(n) =	10		
n =	3	=>	γ(n) =	3	n =	52	=>	γ(n) =	6		
n =	4	=>	γ(n) =	3	n =	53	=>	γ(n) =	10		
n =	5	=>	γ(n) =	4	n =	54	=>	γ(n) =	3		
n =	6	=>	γ(n) =	3	n =	55	=>	γ(n) =	15		
n =	7	=>	γ(n) =	6	n =	56	=>	γ(n) =	6		
n =	8	=>	γ(n) =	3	n =	57	=>	γ(n) =	15		
n =	9	=>	γ(n) =	5	n =	58	=>	γ(n) =	6		
n =	10	=>	γ(n) =	4	n =	59	=>	γ(n) =	6		
n =	11	=>	γ(n) =	6	n =	60	=>	γ(n) =	3		
n =	12	=>	γ(n) =	3	n =	61	=>	γ(n) =	50		
n =	13	=>	γ(n) =	10	n =	62	=>	γ(n) =	3		
n =	14	=>	γ(n) =	3	n =	63	=>	γ(n) =	6		
n =	15	=>	γ(n) =	6	n =	64	=>	γ(n) =	10		
n =	16	=>	γ(n) =	6	n =	65	=>	γ(n) =	8		
n =	17	=>	γ(n) =	6	n =	66	=>	γ(n) =	6		
n =	18	=>	γ(n) =	3	n =	67	=>	γ(n) =	20		
n =	19	=>	γ(n) =	10	n =	68	=>	γ(n) =	3		
n =	20	=>	γ(n) =	3	n =	69	=>	γ(n) =	10		
n =	21	=>	γ(n) =	10	n =	70	=>	γ(n) =	6		
n =	22	=>	γ(n) =	6	n =	71	=>	γ(n) =	20		
n =	23	=>	γ(n) =	6	n =	72	=>	γ(n) =	3		
n =	24	=>	γ(n) =	3	n =	73	=>	γ(n) =	35		
n =	25	=>	γ(n) =	15	n =	74	=>	γ(n) =	3		
n =	26	=>	γ(n) =	4	n =	75	=>	γ(n) =	6		
n =	27	=>	γ(n) =	6	n =	76	=>	γ(n) =	10		
n =	28	=>	γ(n) =	5	n =	77	=>	γ(n) =	10		
n =	29	=>	γ(n) =	10	n =	78	=>	γ(n) =	6		
n =	30	=>	γ(n) =	3	n =	79	=>	γ(n) =	20		
n =	31	=>	γ(n) =	20	n =	80	=>	γ(n) =	3		
n =	32	=>	γ(n) =	3	n =	81	=>	γ(n) =	21		
n =	33	=>	γ(n) =	7	n =	82	=>	γ(n) =	6		
n =	34	=>	γ(n) =	6	n =	83	=>	γ(n) =	6		
n =	35	=>	γ(n) =	6	n =	84	=>	γ(n) =	3		
n =	36	=>	γ(n) =	6	n =	85	=>	γ(n) =	50		
n =	37	=>	γ(n) =	20	n =	86	=>	γ(n) =	6		
n =	38	=>	γ(n) =	3	n =	87	=>	γ(n) =	6		
n =	39	=>	γ(n) =	6	n =	88	=>	γ(n) =	6		
n =	40	=>	γ(n) =	6	n =	89	=>	γ(n) =	15		
n =	41	=>	γ(n) =	15	n =	90	=>	γ(n) =	3		
n =	42	=>	γ(n) =	3	n =	91	=>	γ(n) =	50		
n =	43	=>	γ(n) =	20	n =	92	=>	γ(n) =	6		
n =	44	=>	γ(n) =	3	n =	93	=>	γ(n) =	10		
n =	45	=>	γ(n) =	10	n =	94	=>	γ(n) =	6		
n =	46	=>	γ(n) =	10	n =	95	=>	γ(n) =	6		
n =	47	=>	γ(n) =	6	n =	96	=>	γ(n) =	6		
n =	48	=>	γ(n) =	3	n =	97	=>	γ(n) =	28		
n =	49	=>	γ(n) =	21	n =	98	=>	γ(n) =	3		
n =	50	=>	γ(n) =	4	n =	99	=>	γ(n) =	10		
					n =	100	=>	γ(n) =	10		

n = 101 => γ(n) = 20
n = 102 => γ(n) = 3
n = 103 => γ(n) = 20
n = 104 => γ(n) = 3
n = 105 => γ(n) = 15
n = 106 => γ(n) = 20
n = 107 => γ(n) = 6
n = 108 => γ(n) = 3
n = 109 => γ(n) = 35
n = 110 => γ(n) = 3
n = 111 => γ(n) = 20
n = 112 => γ(n) = 6
n = 113 => γ(n) = 21
n = 114 => γ(n) = 3
n = 115 => γ(n) = 20
n = 116 => γ(n) = 6
n = 117 => γ(n) = 10
n = 118 => γ(n) = 10
n = 119 => γ(n) = 6
n = 120 => γ(n) = 6
n = 121 => γ(n) = 105
n = 122 => γ(n) = 4
n = 123 => γ(n) = 6
n = 124 => γ(n) = 6
n = 125 => γ(n) = 10
n = 126 => γ(n) = 5
n = 127 => γ(n) = 50
n = 128 => γ(n) = 3
n = 129 => γ(n) = 9
n = 130 => γ(n) = 6
n = 131 => γ(n) = 20
n = 132 => γ(n) = 3
n = 133 => γ(n) = 50
n = 134 => γ(n) = 6
n = 135 => γ(n) = 6
n = 136 => γ(n) = 15
n = 137 => γ(n) = 15
n = 138 => γ(n) = 3
n = 139 => γ(n) = 20
n = 140 => γ(n) = 3
n = 141 => γ(n) = 50
n = 142 => γ(n) = 6
n = 143 => γ(n) = 6
n = 144 => γ(n) = 6
n = 145 => γ(n) = 56
n = 146 => γ(n) = 6
n = 147 => γ(n) = 6
n = 148 => γ(n) = 10
n = 149 => γ(n) = 10
n = 150 => γ(n) = 3

n = 151 => γ(n) = 50
n = 152 => γ(n) = 3
n = 153 => γ(n) = 15
n = 154 => γ(n) = 10
n = 155 => γ(n) = 20
n = 156 => γ(n) = 6
n = 157 => γ(n) = 50
n = 158 => γ(n) = 3
n = 159 => γ(n) = 6
n = 160 => γ(n) = 6
n = 161 => γ(n) = 28
n = 162 => γ(n) = 6
n = 163 => γ(n) = 21
n = 164 => γ(n) = 3
n = 165 => γ(n) = 10
n = 166 => γ(n) = 20
n = 167 => γ(n) = 6
n = 168 => γ(n) = 3
n = 169 => γ(n) = 105
n = 170 => γ(n) = 4
n = 171 => γ(n) = 20
n = 172 => γ(n) = 10
n = 173 => γ(n) = 10
n = 174 => γ(n) = 3
n = 175 => γ(n) = 20
n = 176 => γ(n) = 10
n = 177 => γ(n) = 21
n = 178 => γ(n) = 6
n = 179 => γ(n) = 6
n = 180 => γ(n) = 3
n = 181 => γ(n) = 175
n = 182 => γ(n) = 3
n = 183 => γ(n) = 20
n = 184 => γ(n) = 6
n = 185 => γ(n) = 15
n = 186 => γ(n) = 6
n = 187 => γ(n) = 20
n = 188 => γ(n) = 6
n = 189 => γ(n) = 10
n = 190 => γ(n) = 15
n = 191 => γ(n) = 20
n = 192 => γ(n) = 3
n = 193 => γ(n) = 36
n = 194 => γ(n) = 3
n = 195 => γ(n) = 6
n = 196 => γ(n) = 20
n = 197 => γ(n) = 20
n = 198 => γ(n) = 3
n = 199 => γ(n) = 50
n = 200 => γ(n) = 3

Таблица 2

Значения обратной функции Эйлера $\varphi^{-1}(m)$

(см. раздел 6.4)

1 : 1, 2
2 : 3, 4, 6
4 : 5, 8, 10, 12
6 : 7, 9, 14, 18
8 : 15, 16, 20, 24, 30
10 : 11, 22
12 : 13, 21, 26, 28, 36, 42
16 : 17, 32, 34, 40, 48, 60
18 : 19, 27, 38, 54
20 : 25, 33, 44, 50, 66
22 : 23, 46
24 : 35, 39, 45, 52, 56, 70, 72, 78, 84, 90
28 : 29, 58
30 : 31, 62
32 : 51, 64, 68, 80, 96, 102, 120
36 : 37, 57, 63, 74, 76, 108, 114, 126
40 : 41, 55, 75, 82, 88, 100, 110, 132, 150
42 : 43, 49, 86, 98
44 : 69, 92, 138
46 : 47, 94
48 : 65, 104, 105, 112, 130, 140, 144, 156, 168, 180, 210
52 : 53, 106
54 : 81, 162
56 : 87, 116, 174
58 . 59, 118
60 : 61, 77, 93, 99, 122, 124, 154, 186, 198
64 : 85, 128, 136, 160, 170, 192, 204, 240
66 : 67, 134
70 : 71, 142
72 : 73, 91, 95, 111, 117, 135, 146, 148, 152, 182, 190, 216, 222, 228, 234, 252, 270
78 : 79, 158
80 : 123, 164, 165, 176, 200, 220, 246, 264, 300, 330
82 : 83, 166
84 : 129, 147, 172, 196, 258, 294
88 : 89, 115, 178, 184, 230, 276
92 : 141, 188, 282
96 : 97, 119, 153, 194, 195, 208, 224, 238, 260, 280, 288, 306, 312, 336, 360, 390, 420
100 : 101, 125, 202, 250

102 : 103, 206
104 : 159, 212, 318
106 : 107, 214
108 : 109, 133, 171, 189, 218, 266, 324, 342, 378
110 : 121, 242
112 : 113, 145, 226, 232, 290, 348
116 : 177, 236, 354
120 : 143, 155, 175, 183, 225, 231, 244, 248, 286, 308, 310, 350, 366, 372, 396, 450, 462
126 : 127, 254
128 : 255, 256, 272, 320, 340, 384, 408, 480, 510
130 : 131, 262
132 : 161, 201, 207, 268, 322, 402, 414
136 : 137, 274
138 : 139, 278
140 : 213, 284, 426
144 : 185, 219, 273, 285, 292, 296, 304, 315, 364, 370, 380, 432, 438, 444, 456, 468, 504, 540, 546, 570, 630
148 : 149, 298
150 : 151, 302
156 : 157, 169, 237, 314, 316, 338, 474
160 : 187, 205, 328, 352, 374, 400, 410, 440, 492, 528, 600, 660
162 : 163, 243, 326, 486
164 : 249, 332, 498
166 : 167, 334
168 : 203, 215, 245, 261, 344, 392, 406, 430, 490, 516, 522, 588
172 : 173, 346
176 : 267, 345, 356, 368, 460, 534, 552, 690
178 : 179, 358
180 : 181, 209, 217, 279, 297, 362, 418, 434, 558, 594
184 : 235, 376, 470, 564
190 : 191, 382
192 : 93, 221, 291, 357, 386, 388, 416, 442, 448, 476, 520, 560, 576, 582, 612, 624, 672, 714, 720, 780, 840
196 : 197, 394
198 : 199, 398
200 : 275, 303, 375, 404, 500, 550, 606, 750
204 : 309, 412, 618
208 : 265, 424, 530, 636
210 : 211, 422
212 : 321, 428, 642
216 : 247, 259, 327, 333, 351, 399, 405, 436, 494, 518, 532, 648, 654, 666, 684, 702, 756, 798, 810
220 : 253, 363, 484, 506, 726

222 : 223, 446
224 : 339, 435, 452, 464, 580, 678, 696, 870
226 : 227, 454
228 : 229, 458
232 : 233, 295, 466, 472, 590, 708
238 : 239, 478
240 : 241, 287, 305, 325, 369, 385, 429, 465, 482, 488, 495, 496, 525, 572, 574, 610, 616, 620, 650, 700, 732, 738, 744, 770, 792, 858, 900, 924, 930, 990, 1050
250 : 251, 502
252 : 301, 381, 387, 441, 508, 602, 762, 774, 882
256 : 257, 512, 514, 544, 640, 680, 768, 816, 960, 1020
260 : 393, 524, 786
262 : 263, 526
264 : 299, 335, 483, 536, 598, 644, 670, 804, 828, 966
268 : 269, 538
270 : 271, 542
272 : 289, 411, 548, 578, 822
276 : 277, 329, 417, 423, 554, 556, 658, 834, 846
280 : 281, 319, 355, 562, 568, 638, 710, 852
282 : 283, 566
288 : 23, 365, 455, 459, 555, 584, 585, 592, 608, 646, 728, 730, 740, 760, 864, 876, 888, 910, 912, 918, 936, 1008, 1080, 1092, 1110, 1140, 1170, 1260
292 : 293, 586
294 : 343, 686
296 : 447, 596, 894
300 : 341, 453, 604, 682, 906
306 : 307, 614
310 : 311, 622
312 : 313, 371, 395, 471, 477, 507, 626, 628, 632, 676, 742, 790, 942, 948, 954, 1014
316 : 317, 634
320 : 425, 561, 615, 656, 704, 748, 800, 820, 850, 880, 984, 1056, 1122, 1200, 1230, 1320
324 : 489, 513, 567, 652, 972, 978, 1026, 1134
328 : 415, 664, 830, 996
330 : 331, 662
332 : 501, 668, 1002
336 : 337, 377, 609, 645, 674, 688, 735, 754, 784, 812, 860, 980, 1032, 1044, 1176, 1218, 1290, 1470
342 : 361, 722
344 : 519, 692, 1038
346 : 347, 694
348 : 349, 413, 531, 698, 826, 1062
352 : 353, 391, 445, 706, 712, 736, 782, 890, 920, 1068, 1104, 1380
356 : 537, 716, 1074

Литература

1. *Булос Дж., Джеффри Р.* Вычислимость и логика. М.: Мир, 1994.

2. *Мендельсон Э.* Введение в математическую логику. М.: Наука, 1984.

3. *Лавров И. А, Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. М.: Физ.-мат. литература, 1995.

4. *Роджерс Х.* Теория рекурсивных функций и эффективная вычислимость. М., Мир, 1972.

5. *Столл Р.* Множества, логика, аксиоматические теории. М.: Просвещение, 1968.

6. *Успенский В. А., Верещагин Н. К., Плиско В. Е.* Вводный курс математической логики. М.: МГУ, 1991

7. Справочная книга по математической логике / Под ред. Дж. Барвайза. Часть 1, Теория моделей. М.: Наука, 1982.

8. *Гейтинг А.* Интуиционизм. М., Мир, 1965.

9. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. М.: Мир, 1983.

10. *Клини С. К.* Математическая логика. М.: Мир, 1973.

11. *Линдон Р.* Заметки по логике. М.: Мир, 1968.

12. *Успенский В. А.* Лекции о вычислимых функциях. М., Физматгиз, 1960.

13. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.

14. *Фейс Р.* Модальная логика. М.: Наука

15. *Van Benthem J.* Essays in Logical Semantics. Studies in Linguistics and Philosophy. Dordrecht: D.Reidel Publishing Company, 1986.

16. *Van Dalen D.* Logic and Structure. Universitext. Springer-Verlag, 1994

17. *Gamut L. T. F.* Logic, Language, and Meaning. University of Chicago Press, Chicago, 1991

18. *McCawley J. D.* Everything that linguists have always wanted to know about logic. Chicago: University of Chicago Press. 1981.

19. Карпенко А.С. Логика Лукасевича и простые числа. М.: Наука. 2000