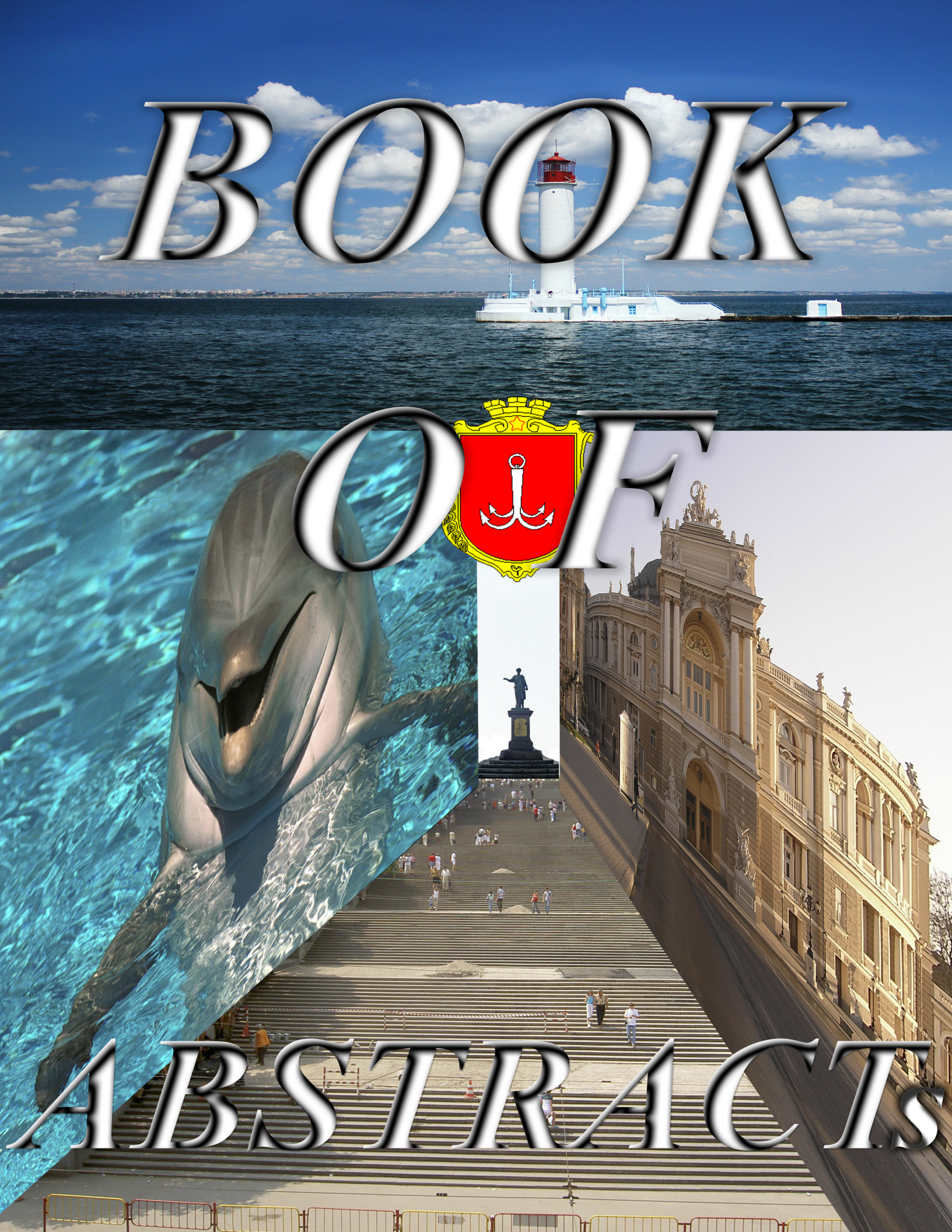


BOOK OF ABSTRACTS

INSTITUTE OF MATHEMATICS OF NAS OF UKRAINE
KIEV TARAS SHEVCHENKO NATIONAL UNIVERSITY
I.I. MECHNIKOV ODESSA NATIONAL UNIVERSITY
ODESSA NATIONAL POLYTECHNIC UNIVERSITY

Book of abstracts

of the

International Scientific Conference

"Computer Algebra and Information Technology"

August 20-26, 2012
I.I. Mechnikov Odessa National University
Odessa, Ukraine

УДК 511.33+519.2+681.3+621.3
ББК 22,32

Матеріали статей опубліковані в авторській редакції

Тези доповідей Міжнародної наукової конференції "Комп'ютерна алгебра та інформаційні технології", м. Одеса, 20-26 серпня 2012р. - Одеса: ОНУ, 2012.- 80 с.

Збірка містить матеріали доповідей Міжнародної конференції з сучасних проблем комп'ютерної алгебри та інформаційних технологій за основними напрямками: алгоритми комп'ютерної алгебри, аналітична теорія чисел та її застосування, штучний інтелект, нейронні мережі, прикладна теорія груп, генерування псевдовипадкових чисел, теорія автоматів, мережі Петрі.

Book of abstracts of the International scientific conference "Computer Algebra and Information Technology", Odessa, August 20-26, 2012р. - Odessa: ONU, 2012.- p.80

This composite book contains the articles of talks of International conference by the modern problems of computer algebra and information technology based on the following areas: Algorithms of Computer Algebra, Analytic Number Theory and its Applications, Artificial Intelligence, Neuron Networks, Applied Group Theory, Pseudorandom Numbers generation, Automata Theory, Petri Nets.

УДК 511.33+519.2+681.3+621.3
ББК 22,32

Contents

Semigroups of transformations defined by branchless automata and slowmoving automata of finite type

A. S. Antonenko

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: aantonenko@mail.ru)

We consider finite automata which input and output alphabet coincide and transformation of finite or infinite words defined by them. We denote such automata by quadruples $A = (X, Q, \pi, \lambda)$, where X is the finite alphabet (where $|X| \geq 2$), Q is the finite nonempty set of states, $\pi : X \times Q \to Q$ is the transition functions and $\lambda : X \times Q \to X$ is the output function.

Definition 1. [1] We say that q is a *state without branches* or a *branchless state* if for all $x_1, x_2 \in X$ the equality $\pi(x_1, q) = \pi(x_2, q)$ holds.

In other words a state is a brachless one if a transition from this state is independent of input symbol. We say that a finite automaton is a branchless automaton if all it states are branchless ones. Branchless automata generate only finite semigroups. Let $T_X = \{f | f : X \to X\}$ be the semigroup of all transformation of the alphabet X (the full symmetric semigroup).

Theorem 1. *A semigroup S is isomorphic to some semigroup, generated by a finite automaton without branches over the alphabet X ($|X| = m$), if and only if S is isomorphic to a semigroup $S' \leq (T_X)^k$ for some natural k such that there exists a natural number l such that $1 \leq l \leq k$ and*

$$v_1 v_2 \dots v_k \in S' \Rightarrow v_2 \dots v_k v_l \in S' \tag{1}$$

where $v_i \in T_X$ for $1 \leq i \leq k$.

That means that we can describe transformation semigroups generated by branchless automata in terms of vector semigroups with the special "cyclic" property. Note that there exists subsemigroups S' of $(T_X)^k$ that are not isomorphic to any subsemigroup of $(T_X)^l$ with the "cyclic" property 1.

The semigroup generated by all branchless automaton transformations is not finite-generated. Moreover, using the automaton from the Theorem in Gecheg[2] which is in fact a branchless automaton one can prove that the semigroup generated by all branchless automaton transformations can not be a subsemigroup of any finite-generated semigroup of finite automaton transformations.

Let us consider transformations defined by automata over two-symbol alphabet with the following two properties [1]:

1) for each state $q \in Q$ of an automaton, there exist at most one symbol $x \in X$ such that $\pi(x, q) \neq q$ (slowmoving automata);

2) there are no cycles except of loops in the Moore diagram of an automaton (automata of finite type).

We find a generating set $\left\{\alpha_i, \beta_i, \delta_i : i = \overline{0, \infty}\right\}$ of the semigroup SSl_2 generated by all slowmoving automata of finite type over two-symbol alphabet.

Lemma 1. *For an arbitrary integers i, j such that $0 \leq i < j$, the next relation holds*

$$\delta_i = \beta_i \beta_{i+1} \cdots \beta_{j-1} \delta_j$$

This implies that $\left\{\alpha_i, \beta_i, \delta_i : i = \overline{0, \infty}\right\}$ is not the minimal generating set of SSl_2. We prove that SSl_2 has no minimal generating set.

References

[1] Antonenko A. S., Berkovich E. L., Groups and semigroups defined by some classes of Mealy automata, Acta Cybernetica 2007, 18, pp.23-46.

[2] F. Gecheg, On a group of one-to-one transformations specified by finite automata, Kibernetika, No. 1 (1965) (in Russian).

Синтез примитивных матриц над конечными полями Галуа и их приложения

А.Я. Белецкий[1] и Е.А. Белецкий[2]

[1] National Aviation University, Department of Radioelectronics, str. Komarova 1, 03680 Kiev, Ukraine (E-mail: abelnau@ukr.net)

[2] National Aviation University, Department of Radioelectronics, str. Komarova 1, 03680 Kiev, Ukraine (E-mail: abelnau@ukr.net)

Пусть $A = (a_{i,j})$ является положительной невырожденной матрицей порядка $n > 1$ над полем целых чисел таких, что $a_{i,j} \in GF(p)$ для всех $i, j = \overline{1, n}$, и $E = (\delta_{i,j})$, где $\delta_{i,j}$ - символ Кронекера, есть единичная матрица того же порядка, что и A. Матрица A невырожденная в поле $GF(p)$, если ее определитель $detA$ по модулю p не равен нулю, т.е. $detA \pmod p \in \overline{1, p-1}$, где p - простое число. Операция возведения матрицы A в некоторую степень t выполняется в кольце вычетов по модулю p, при этом каждый элемент матрицы A^t приводится к неотрицательному остатку по модулю p. Последовательность степеней матрицы A, начиная с нулевой степени, для которой $A^0 = E$, образует циклическую группу порядка e. Матрицу A будем называть *примитивной*, если наименьшее натуральное e, при котором $A^e = E$, удовлетворяет соотношению $e = p^n - 1$.

Основная проблема, обсуждаемая в докладе, состоит в разработке алгоритмов построения *обобщенных примитивных матриц Галуа*(G) *и Фибоначчи*(F) n-го порядка над полем $GF(p)$, однозначно определяющих структуру соответствующих n-разрядных линейных регистров сдвига (ЛРС) с линейными обратными связями (ЛРСЛОС) максимального периода. Классические G и F матрицы (назовем их *простыми* матрицами), принадлежащие подмножеству матриц Фро-

бениуса, можно записать в виде:

$$G = \begin{pmatrix} u_{n-1} & u_{n-2} & u_{n-3} & \cdots & u_1 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}; F = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 & u_1 \\ 0 & 1 & \cdots & 0 & 0 & u_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 & u_{n-2} \\ 0 & 0 & \cdots & 0 & 1 & u_{n-1} \end{pmatrix}, \quad (1)$$

где u_i, $i = \overline{1, n-1}$, - коэффициенты двоичного примитивного полинома степени n.

Матрицы G и F связаны оператором *правостороннего транспонирования* $\perp$ (транспонирования относительно вспомогательной диагонали), т.е. $G \overset{\perp}{\longleftrightarrow} F$. Матрицы (1), которые назовем *базовыми* матрицами, лежат в основе построения классических ЛРСЛОС максимального периода. Базовым матрицам Галуа и Фибоначчи ставятся в соответствие *сопряженные* G^* и F^* матрицы, определяемые соотношением $M^* = 1 \cdot M \cdot 1$, где 1 - оператор (матрица) инверсной перестановки, в которой на вспомогательной диагонали располагаются единицы, а в остальных элементах - нули.

Обобщенные примитивные G матрицы формируются по правилу *диагонального заполнения*, суть которого состоит в следующем. Пусть ω_m есть примитивный элемент поля $GF(p^n)$, представленный в векторной (числовой) форме в виде полинома над $GF(p)$ степени m, $m < 1 < n$, и f_n - произвольный неприводимый полином (НП) степени n (совсем не обязательно примитивный). Поместим образующий элемент ω_m в правом углу нижней строки обобщенной матрицы G, приписав ей (строке) номер 1. Элементы первой строки, расположенные левее ω, заполняются нулями. Последующие строки матрицы G (по направлению снизу вверх) образуются циклическим сдвигом справа налево предыдущих строк. Если при этом левый элемент сдвигаемой строки равен 1, то выполняется обычный сдвиг строки на один разряд влево, а в правый элемент освободившийся строки записывается 0. Разрядность подобных строк становится на единицу больше порядка матрицы. Векторы, отвечающие таким

строкам, приводятся к остатку по модулю НП f_n. Тем самым данные строки матрицы также становятся n-битными. Пример обобщенных базовых (G и F) и сопряженных (G^* и F^*) примитивных матриц четвертого порядка с элементами над $GF(3)$ приведен ниже.

$$G = \begin{pmatrix} 1\,1\,1\,2 \\ 1\,0\,2\,1 \\ 2\,2\,2\,2 \\ 1\,1\,0\,2 \end{pmatrix}; \quad F = \begin{pmatrix} 2\,2\,1\,2 \\ 0\,2\,2\,1 \\ 1\,2\,0\,1 \\ 1\,2\,1\,1 \end{pmatrix}; \qquad (2)$$

$$G^* = \begin{pmatrix} 2\,0\,1\,1 \\ 2\,2\,2\,2 \\ 1\,2\,0\,1 \\ 2\,1\,1\,1 \end{pmatrix}; \quad F^* = \begin{pmatrix} 1\,1\,2\,1 \\ 1\,0\,2\,1 \\ 1\,2\,2\,0 \\ 2\,1\,2\,2 \end{pmatrix}.$$

В качестве примера на Рис. 1 и 2 показаны структурные схемы обобщенных ЛРС Галуа и Фибоначчи, обратные связи в котором заданы матрицами G и F^* системы (2) соответственно.

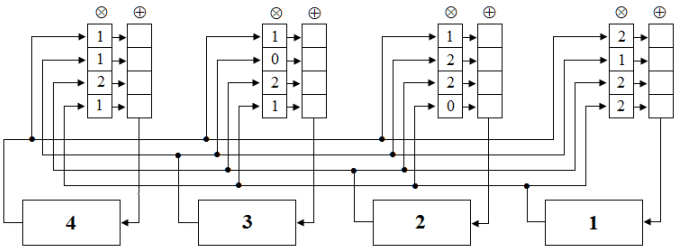


Figure1. Структурная схема обобщенного базового ЛРС Галуа над $GF(3)$

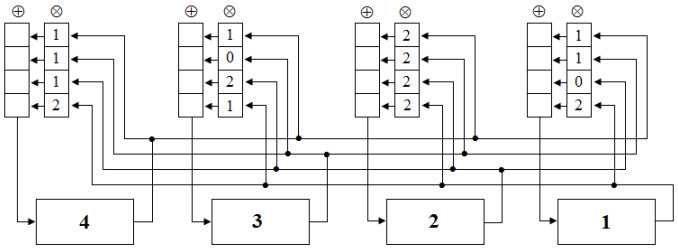


Figure2. Структурная схема обобщенного сопряженного ЛРС Фибоначчи над $GF(3)$

К основным результатам исследований по обозначенной тематике следует отнести разработку алгоритмов синтеза обобщенных базовых и сопряженных матриц Галуа и Фибоначчи, элементы которых принадлежат простому полю $GF(p)$ характеристики $p \geq 2$. Данные

матрицы обладают замечательными свойствами, такими как примитивность и коммутативность, что дало возможность построить на их основе обобщенные ЛРСЛОС максимального периода, а также предложить матричные аналоги криптографического протокола Диффи-Хеллмана. Структурные схемы обобщенных ЛРС оказались как однородными, так и инвариантными к порядкам регистров n и характеристикам p поля $GF(p^n)$.

Обратимые модулярные преобразования на основе обобщенных кодов Грея

А.Я. Белецкий[1] и А.А. Белецкий[2]

[1] National Aviation University, Department of Radioelectronics, str. Komarova 1, 03680 Kiev, Ukraine (E-mail: abelnau@ukr.net)

[2] National Aviation University, Department of Radioelectronics, str. Komarova 1, 03680 Kiev, Ukraine (E-mail: abelnau@ukr.net)

Теория обратимых модулярных преобразований (ОМП) является разделом дискретной математики. В качестве примеров ОМП можно привести операции сложения по mod 2 (операция XOR), прямой и обратной нелинейной подстановки (операция S-box) и другие. Отдельное подмножество ОМП составляют преобразования, в основу которых положены обобщенные коды Грея.

Коды Грея (КГ), предложенные в середине XX века в ответ на запросы инженерной практики относительно построения оптимальных по критерию минимума ошибки неоднозначности преобразователей типа "угол код", на заре своего появления привлекли к себе внимание не только исследователей математиков, но и широкого круга разработчиков разнообразной аппаратуры. Отличительная особенность кодов Грея состоит в том, что в двоичном пространстве (или в двоичной системе счисления) при переходе от изображения одного числа к изображению соседнего старшего или соседнего

младшего числа происходит изменение цифр (1 на 0, или наоборот) только в одном разряде числа. Коды Грея находят широкое применение в различных областях науки и техники. В частности, КГ используются для построения и минимизации ошибок скоростных аналого-цифровых и цифро-аналоговых преобразователей, в теории генетических алгоритмов, при синтезе квазиэквидистантных кодов, в криптографии, теории помехоустойчивого кодирования и пр.

За более чем пятидесятилетнюю историю своего развития теория кодов Грея претерпела незначительные изменения. По-видимому, оказались вне поля зрения, как математиков, так и разработчиков аппаратуры возможности построения кодов, *инверсных* по направлению формирования *классическим* кодам Грея. В известной схеме процесс формирования прямых и обратных кодов Грея развивается по направлению слева направо; при этом старший (левый) разряд преобразуемой кодовой комбинации сохраняется неизменным. Вместе с тем можно построить схему преобразования, в общем, m-ичных равномерных кодов, обратную по направлению классическому (*левостороннему*) преобразованию Грея. В таком классе преобразований, который назван *правосторонним*, при прямом и обратном преобразованиях сохраняется неизменным значение младшего (правого) разряда преобразуемого числа.

Операторы прямого и обратного преобразований Грея левосторонних, для краткости обозначаемые цифрами 2 и 3, как и соответствующие им операторы 4 и 5 правосторонних преобразования Грея, представляют собой правосторонне симметрические матрицы, т.е. матрицы, симметричные относительно вспомогательной диагонали. Нижеследующие соотношения иллюстрируют перечисленные выше операторы четвертого порядка для двоичных кодов Грея.

$$2 = \begin{bmatrix} 1\,1\,0\,0 \\ 0\,1\,1\,0 \\ 0\,0\,1\,1 \\ 0\,0\,0\,1 \end{bmatrix}; \quad 3 = \begin{bmatrix} 1\,1\,1\,1 \\ 0\,1\,1\,1 \\ 0\,0\,1\,1 \\ 0\,0\,0\,1 \end{bmatrix}; \quad 4 = \begin{bmatrix} 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 0\,1\,1\,0 \\ 0\,0\,1\,1 \end{bmatrix}; \quad 5 = \begin{bmatrix} 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 0\,1\,1\,0 \\ 0\,0\,1\,1 \end{bmatrix}.$$

Как следует из приведенных выражений, матрицы 2 и 4, а также матрицы 3 и 5, связаны операцией транспонирования, т.е. $4 = 2^T$,

$5 = 3^T$. Совокупность операторов лево- и правостороннего преобразований Грея (как прямого, так и обратного), совместно с оператором инверсной перестановки (представляющим собой квадратную $(0,1)$-матрицу, в которой элементы вспомогательной диагонали равны единице, а остальные - нулю) и единичной матрицей образуют полную группу *простых* операторов (кодов) Грея. Произведение некоторого набора простых операторов Грея в кольце вычетов по $\bmod 2$ приводит к формированию комбинированных или *составных* кодов Грея. Применение как простых, так и составных кодов Грея привело к конструктивным результатам в задачах определения структуры и взаимосвязи дискретных симметричных систем Виленкина-Крестенсона функций (ВКФ), частным случаем которых являются системы дискретных экспоненциальных функций и системы функций Уолша. И, тем не менее, не для всех порядков систем ВКФ удается "связать" с помощью упомянутых выше преобразований Грея полное множество симметричных систем ВКФ. Возникает так называемая проблема *кластеризации*, которая проявляется в том, что, для примера, только 126 из 448 симметричных систем функций Уолша 16-го порядка оказалось возможным синтезировать с помощью простых и составных кодов Грея. Обозначенную проблему кластеризации удалось разрешить с помощью так называемых обобщенных преобразований Грея. К важнейшим результатам теории ОМП, базирующихся на обобщенных преобразованиях Грея, относятся:

- решена задача синтеза полного множества симметричных систем ВКФ (включая системы ДЭФ и Уолша), получены их структуры и установлены правила взаимосвязи этих систем;
- получены оценки числа симметрических систем ВКФ во всем диапазоне изменения параметров m и n, определяющих порядок N матриц преобразования систем ВКФ $N = m^n$, где m - основание системы счисления (модуль), а n - длина преобразуемой m-ичной кодовой комбинации;
- разработаны прямые методы синтеза фундаментальных инвариантных систем Уолша, формируемых перестановками строк (ба-

зисных функций) систем Уолша-Пели множеством простых операторов Грея;

- доказано, что каждой симметричной системе ВКФ m-рационального порядка $N = m^n$ соответствует единственная однозначно связанная с ней индикаторная матрица n-го порядка. Индикаторной матрицей систем ВКФ является правосторонне симметрическая матрица n-го порядка (необходимые условия), невырожденная в кольце вычетов по $\mathrm{mod} m$ (достаточные условия);
- доказано, что факторизация матриц преобразования систем ВКФ (в том числе систем ДЭФ и Уолша), обычно проводившаяся при решении задач быстрых преобразований Фурье (БПФ), является избыточной, поскольку в алгоритмах БПФ в различных базисах структура дерева преобразования сохраняется неизменной (и повторяет известную схему Кули-Тьюки), а смена базиса достигается элементарными перестановками номеров отсчетов сигнала на входе процессора БПФ, причем правила перестановки определяются индикаторными матрицами систем ВКФ;
- решены прямая и обратная задачи взаимосвязи систем функций Уолша и соответствующих им индикаторных матриц систем;
- разработаны методы синтеза гарантированно невырожденных примитивных двоичных матриц, последовательность степеней которых в кольце вычетов по $\mathrm{mod}\ 2$ образует последовательность максимальной длины. Такие матрицы послужили основой построения достаточно простой и удобной для применения в криптографических приложениях односторонней функции.

В докладе обсуждаются различные направления применения преобразований Грея: в криптографии, теории и практике дискретного спектрального анализа сигналов и изображений, структурном синтезе пересчетных схем и др.

Построение и нумерация комбинаторных объектов на основе позиционных систем счисления

А. А. Борисенко

Сумский государственный университет, кафедра электроники и компьютерной техники, ул. Римского-Корсакова 2, 40007 Сумы, Украина (E-mail: 5352008@ukr.net)

Построения комбинаторных объектов относится к третьей задаче комбинаторики, возникающей после решения двух предшествующих ей задач - задачи на существование и задачи на перечисление, решение которых широко используется в дискретной математике и теории кодирования. Решение задачи на построение особенно актуально для решения задач комбинаторной оптимизации, например, построения перестановок в задаче коммивояжера. Имеется немало научной литературы, где, наряду с первой и второй задачами комбинаторики, решается также и третья ее задача, например, в [1, 2, 3]. Однако, все же, до настоящего времени отсутствует общий подход, который бы позволял с единых позиций решать задачи на построение, к которым следует отнести и задачи нумерации комбинаторных объектов, как их простейшую разновидность.

В данной работе предлагается рассмотреть один из возможных общих подходов к построению и нумерации, комбинаторных объектов, использующий позиционные комбинаторные системы счисления. К наиболее известным и простым системам счисления такого вида следует отнести факториальные системы, генерирующие перестановки [2]. Существуют также фибоначчиевые системы счисления, формирующие числа Фибоначчи [4]. Наряду с этими системами автором были также предложены двоичные биномиальные системы счисления и их такие разновидности, как линейно-циклические и матричные системы для генерации равновесных кодов и близких к ним конфигураций [2]. Также были разработаны многозначные биномиальные системы счисления, генерирующие сочетания и полифакториальные системы для формирования размещений [3, 7].

Однако этим перечнем число комбинаторных систем счисления не ограничивается. В работах автора было показано, что теоретически число таких систем счисления может быть неограниченно большим и порождаться они могут различными классами комбинаторных объектов. Каждый из таких классов характеризуется своей оригинальной структурой. Такую же структуру должна иметь и соответствующая этому классу система счисления. Тогда диапазон чисел, представляемых в комбинаторной системе счисления, будет равен количеству комбинаторных объектов, содержащихся в порождающем ее классе.

Совпадение структур систем счисления и структур, соответствующих им классов комбинаторных объектов, приводит к идее построения комбинаторных объектов заданного класса на основе позиционных систем счисления. Сам комбинаторный объект - это, по сути, завуалированное комбинаторное число. Это значит, что, имея комбинаторное число, можно по соответствующему алгоритму перейти от него к более сложному комбинаторному объекту и обратно - от этого объекта к числу. В первом случае в комбинаторное число вводится избыточная информация, а во втором - она устраняется.

Однако самое интересное состоит в том, что каждому классу комбинаторных чисел соответствует неограниченное количество различных классов комбинаторных объектов, обладающих такой же структурой, что и структура комбинаторной системы счисления. Это позволяет строить на основе одной комбинаторной системы счисления соответственно неограниченное количество разных по внешней форме классов комбинаторных объектов, но, при этом, имеющих одинаковую с ней структуру. Поэтому, например, одна и та же двоичная биномиальная система счисления позволяет строить композиции, универсальный промышленный код, равновесный код с различными ограничениями и прочие менее известные классы комбинаторных объектов. При этом для всех этих классов характерным свойством остается идентичность их структур.

Задача построения комбинаторных систем счисления отнюдь не простая и пока еще не поддается полной формализации до уровня компьютерных алгоритмов, так как требует сложных неформаль-

ных рассуждений, но сам путь или принцип их построения понятен. К сожалению, в настоящее время теория комбинаторных систем счисления находится в зачаточном состоянии и в ней имеется множество нерешенных задач, например, не разработана обобщенная арифметика, которая бы годилась для любой комбинаторной системы счисления. Поэтому разработка арифметик производится на уровне конкретных систем счисления, что приводит к немалым трудностям.

Существенным свойством, при использовании комбинаторных систем счисления, является также и то, что их нумерационные функции могут, как, впрочем, и функции всех остальных позиционных систем счисления, нумеровать свои числа, производя в общем случае, при этом, сжатие информации. Это свойство, применяя двоичные биномиальные системы счисления, можно использовать и для сжатия обычных двоичных кодов в принципе неограниченной длины. Особенно эффектно выглядит это сжатие при его реализации в виде соответствующих оригинальных цифровых устройств и систем, что дает возможность резко поднять быстродействие алгоритмов сжатия и в конечном итоге получать на них патенты.

В этом случае сначала происходит переход от сжимаемого комбинаторного объекта к числу соответствующей системы счисления, а затем от этой системы к номеру. Обратная процедура перехода от номера к комбинаторной системе счисления и далее к соответствующему комбинаторному объекту, позволяет, в силу ввода избыточной информации, использовать его для помехоустойчивого кодирования.

Таким образом, использование комбинаторных систем счисления дает возможность разработать общий метод построения и нумерации различных классов комбинаторных объектов, которые могут находить применение как при построении математических теорий, где требуются такие операции, так и при кодировании информации, а также при проектировании соответствующих технических систем и устройств.

References

[1] Рейнгольд Э, Нивергельт Ю, Део Н., Комбинаторные алгоритмы. Теория и практика / Пер. с англ. Изд. "Мир". - М.: 1980. - 476с.

[2] Липский В., Комбинаторика для программистов: Пер. с польск. - М.: Мир, 1988. - 213с.

[3] Андерсон Джеймс, Дискретная математика и комбинаторика - Discrete Mathematics with Combinatorics. - М.: "Вильямс", 2006. - 960с.

[4] Стахов А. П., Фибоначчиевые двоичные позиционные системы счисления: Кодирование и передача дискретных сообщений в системах связи. - М.: Наука, 1976. - 179с.

[5] Борисенко А.А., Введение в теорию биномиального счета: Монография / Борисенко А.А.- Сумы: ИТД "Университетская книга", 2004. - 88с.

[6] Борисенко А. А., Об одной системе счисления с биномиальным основанием. - Рук. деп. В ВИНИТИ, 1982. - №874-82. - 8с.

[7] Борисенко А.А., Полифакториальная система счисления // Вестник СумГУ, 2011. - №2, с.60-64.

Permutation generation and numeration based on factorial numbers in computer systems

A. A. Borysenko[1], A. E. Goryachev[2], and V. V. Siriachenko[3]

[1] Sumy State University, Department of Electronics and Computer Technology, str. Rimskogo-Korsakova 2, 40007 Sumy, Ukraine (E-mail: electron@sumdu.edu.ua)

[2] Sumy State University, Department of Electronics and Computer Technology, str. Rimskogo-Korsakova 2, 40007 Sumy, Ukraine (E-mail: electron@sumdu.edu.ua)

[3] Sumy State University, Department of Electronics and Computer Technology, str. Rimskogo-Korsakova 2, 40007 Sumy, Ukraine (E-mail: electron@sumdu.edu.ua)

The permutations are widely used in practice for solving various problems, among which are the combinatorial optimization problems, a data transfer and protection from unauthorized access. The existing specialized methods for obtaining long length permutations have such a common drawback as increased complexity of the algorithms for generating permutations, which is not acceptable for solving various tasks [1].

More effective method of solving the problem of permutations generation is to use factorial numbers [2]. The advantages of permutations generation based on factorial numbers are in a fairly high speed of the method, as well as the simplicity of the software and hardware implementation of the algorithm [3].

Converting numbers from a uniform number system to the factorial number system is done in next order:

The first step is to divide the number being converted by 1. The modulo in this case will create a number of zero digit. Further, at each step of converting the quotient of the previous division is divided into a number one more than the previous divisor, that is, is a sequential number division by 1, 2, 3, etc. The division continues for as long as the quotient is less than the divisor. Then this quotient is written and after it from right to left the earlier obtained modulus are written. This sequence creates the desired number.

Consider the permutation algorithm based on the factorial numbers. To find a correlation between the number in factorial number system

and permutation, the first digit of factorial number should remain unchanged, and is considered as the first element of the permutation. The next digit of factorial number is compared with the first element of the permutation, and if it is equal to or greater then it has to be increased by 1, and if not, leaved unchanged. In both cases the second element of the permutation is obtained. Further, in general, first the comparison of factorial number digit with the smallest already found permutation element is performed. If this digit is t is equal to or greater than smallest element, then it is incremented by one. Otherwise it becomes another element of the permutation. An increased digit of factorial number is further compared with the smallest already found element of the permutation without the element against which the comparison has already taken place, and then the cycle repeats as long as there is no already found permutation element. Next, the next digit of factorial number is selected and using the rules above a new element of the permutation is formed, and so continues until the last digit of factorial number.

Factorial numbers are also used to solve the problem of permutations numeration. This requires finding corresponding factorial number for specified permutation and then using the factorial function to get the uniform number of permutation.

Transition to the factorial number from permutation is done as follows: the first digit of factorial number is a first element of the permutation. Next digit of factorial number will be equal to the second element of the permutation, if this element is smaller than the first element of a permutation, or reduced by 1 if it is more of this element. Clearly, they can not be equal. Accordingly, value of each element of the permutation is decreased by the number of previous permutation elements smaller than this element.

Converting factorial numbers into uniform numbers is performed by substituting the factorial numbers in numeric (numbering) function for the factorial number system [2]. Thus, all of the featured in this function multiplications and additions are performed.

References

[1] Reingold E. Combinatorial algorithms: theory and practice / E. Reingold, J. Nievergeld, N. Deo - M.: "Mir", 1980. - 477 p.

[2] Borysenko A.A. Electronic system of permutations generation based on a factorial numbers / A.A. Borysenko, , I.A. Kulik, A.E. Goryachev // Visnyk SumDU Technical sciences. - 2007. - №1. - p. 183 - 188.

[3] Borysenko A.A. Generation of Permutations Based Upon Factorial Numbers / A.A. Borysenko, V.V. Kalashnikov, I.A. Kulik, A.E. Goryachev // Eighth International Conference on Intelligent Systems Design and Applications. Kaohiung, Taiwan, 2008. - p. 57 - 61.

Information system of assessment and certification of civil servants

A. B. Kyrushko[1] and V. M. Chaplyha[2]

[1] Company "Investment Capital Ukraine", str. B. Khmelnytsky 19-21, 01030 Kiev, Ukraine (E-mail: Andrew-bk@yandex.ua)

[2] Lviv Banking Institute of BU of NBU, av. Chornovola 61, 79058 Lviv, Ukraine (E-mail: 4vyach@gmail.com)

Automation of the annual assessment and appraisal of civil servants in accordance with the law and departmental regulations is an important way of operational planning and analysis of the work of civil servants, guarantee of unbiased and fair evaluation of their performance, increasing motivation, strengthening of strategic orientation, coordination of actions and performance, promotion of transparency and validity of the passage of the civil service.

The algorithm of electronic support to the annual assessment of public servants includes the following: update of the individual plans of

civil servants taking into account strategic goals of the department, designation of a list of civil servants to be evaluated, announcement of the terms of annual evaluation, generation and distribution of logins and passwords to undergo evaluation, evaluation of the individual plan, performance evaluation, assessment of achievement of strategic objectives by the defined criteria, assessment of competence profiles of leadership, evaluation of the human resource management potential, processing and analysis of assessment results by means of neural networks, publication of the results, training for current or higher level position (in case of the corresponding decision), monitoring of the implementation of decisions taken after the evaluation.

The developed models and the algorithm underlie the web-based information system "Annual evaluation of civil servants of the National Bank of Ukraine". The system has client server architecture, implemented in the environment Net and works in OS MS Windows XP and Windows 7. The system is modular and consists of interactive subsystems: Reference educational and knowledge testing, evaluation of civil servants, mining of evaluation data using neural network technology.

Reference educational subsystem contains the following modules: "Legislative acts", "Regulations of Civil Service of Ukraine", "Regulations of the National Bank of Ukraine", "Encyclopaedic referential dictionary of Ukrainian legislative terms" and " Knowledge testing". Informational and methodological support of the modules is integrated and complies with active legislation.

Subsystem of civil servants' appraisal consists of two software components: AWS of Administration (AdminTool executive program) and AWS of Evaluation process (client's part with the access through web-interface). AWS of Administration allows creating and editing a multi-hierarchical structure of the organization's departments, keeping records of employees and changing their characteristics according to a career history. It is possible to create job positions and edit their parameters according to the staff schedule along with setting out the level of the hierarchy for different positions, and the ability to edit the user access parameters to the client's part of the system and recover lost passwords.

The built-in analytical module makes it possible to calculate the over-

all performance indicators of individual employees or units based on statistical data and expert rules and correlate them with others. Provided there is a large volume of historical data the analyst can predict the outcome of a specialist's appraisal and then evaluate how the actual result corresponds to the expected one. The basis of the analytical module is the neuron fuzzy system, which is capable of working simultaneously with both clear and fuzzy input variables.

Модели и информационные системы непрерывного аудита в банке

С.Т. Иванишин[1] и В.В. Чаплыга[2]

[1] Университет банковского дела НБУ, ул. Андреевская 1, 04070 Киев, Украина (E-mail: ist@bank.gov.ua)

[2] Львовский институт банковского дела УБС НБУ, пр. Чорновила 61, 79058 Львов, Украина (E-mail: 4vyach@gmail.com)

В современных условиях финансовой нестабильности и внедрения превентивных мероприятий по отношению к возможной второй волне кризисных явлений в экономике, резко возрастают требования к оперативности и эффективности системы управления в банковской сфере, особенно, управления рисками и процессами внутреннего контроля, которые требует соответствующего повышения оперативности и эффективности внутреннего аудита, усиление его роли как независимого и профессионального органа по предоставлению гарантий и консультаций, направленных на повышение эффективности деятельности банка.

Волатильность внешней среды обусловливает переход к менеджменту банком и принятия решений заинтересованными лицами в режиме реального времени, которое требует от аудиторов изобретать новые способы для внедрения постоянного мониторинга, сбора и анализа аудиторских доказательств. Это, с одной стороны, и ши-

рокое применение в банковском секторе корпоративных информационных систем и бурное развитие информационных технологий, особенно, новых хранилищ данных и методов интеллектуального анализа данных, с другой стороны, способствуют построению систем непрерывного аудита и мониторинга.

Перспективными технологиями для создания систем непрерывного аудита и мониторинга является хранилища и витрины данных, интеллектуальный анализ данных, экспертные системы, интеллектуальные агенты, нечеткие нейронные сети, сценарный подход и функции доверия при оценке рисков, учет в режиме реального времени, XBRL/XML и другие.

В условиях функционирования в банке корпоративных ERP систем и автоматизированных банковских систем с хранилищами данных возможно создание системы непрерывного аудита и мониторинга на базе встроенных модулей в корпоративные информационные системы. Эти модули должны обеспечивать: автоматизацию мониторинга и анализа управления бизнес-процессами, широкие возможности интеллектуального анализа данных, мультиплатформенность и кросс-платформенность анализа, автоматизированный аудит системы внутреннего контроля, возможность осуществления мониторинга управления и мониторинга транзакций, поддержку корпоративных IT систем, витрины информации и отчетности.

При применении витрин данных аудита не требуется наличие корпоративных хранилищ данных. Собранные с разных бизнес-единиц данные физически хранятся на сервере внутреннего аудита для быстрого доступа. При этом корпоративные данные не дублируются, а хранятся только те, что отобраны для аудита. Интегрированный с витриной данных модуль аудита имеет следующие характеристики: встроенный механизм запросов, анализа и отчетности в рамках единого интерфейса пользователя, способность легко экспортировать результаты запросов в общие электронные таблицы и базы данных, 3D-визуализация данных для дальнейшего исследования и интеллектуального анализа, многомерные базы данных и агрегация данных, моделирование и мощный статистический анализ, использование интеллектуальных агентов, использование нечетких нейронных

сетей для обработки неструктурированной информации в условиях неопределенности, скрытых связей и зашумленности, использование XML и веб-сервисов.

В докладе рассматривается архитектура системы непрерывного аудита, которая содержит: витрину данных для определенных ролей пользователей; подсистему анализа и отчетности для получения и передачи результатов в витрину данных; аналитический модуль индикаторов риска или параметров управления с настройкой на тестирование или аудит; хранилище данных аудита с интегрированными модулями в соответствии с ключевыми индикаторами риска; модуль импорта, сопоставления и загрузки данных; а также существующая инфраструктура интегрированных или отдельных корпоративных IT систем, из которых поступает входная информация для системы непрерывного аудита.

Представлена реализованная Система автоматизации внутреннего аудиту (CABA) в банке с отдельными функциями непрерывного аудита.

On Average Value Of Pillai's Function In An Arithmetic Progression

Z. Yu. Dadayan

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: dadayan.zy@gmail.com)

In 1933 S. Pillai [3] introduced an arithmetic function defined on natural numbers

$$g(n) = \sum_{k=1}^{n} (k,\, n), \tag{1}$$

where $(k,\, n)$ is the greatest common divisor of k and n. This function is common in asymptotic problems of numbers theory. Beginning from Broughan's work [2] in the last 10 years has been built asymp-

totic formulas for the summation functions associated with $g(n)$ or its generalizations [1, 2].

Our task was to obtain an asymptotic formula for the sum

$$G(x;\, a,\, q) := \sum_{\substack{n\equiv a(mod\ q)\\ n\leq x}} g(n), \tag{2}$$

where $(a,\, q) = 1$ and q grow with x.

O. Bordelles [1] showed that Pillai's function is a Dirichlet convolution

$$g = \varphi * Id,$$

where φ – Euler's function and Id – identity function. So for $Res > 2$ we have

$$\sum_{n=1}^{\infty} \frac{g(n)}{n^s} = \frac{\zeta^2(s-1)}{\zeta(s)}.$$

Hence, for $Res > 2$

$$\sum_{\substack{n=1\\ n\equiv l(mod\ q)}}^{\infty} \frac{g(n)}{n^s} = \sum_{l_3\in\mathbb{Z}_q^*} f(s;\, l_3,\, q)\times \\ \times \sum_{\substack{l_1,l_2\in\mathbb{Z}_q^*\\ l_1l_2\equiv a\bar{l}_3(mod\ q)}} q^{-2(s-1)}\zeta\left(s-1,\, \frac{l_1}{q}\right)\zeta\left(s-1,\, \frac{l_2}{q}\right), \tag{3}$$

where $\mathbb{Z}_q^* = \{a \in \mathbb{Z}_q |\, (a,\, q) = 1\},\ l_3\bar{l}_3 \equiv 1\,(mod\, q),\ \zeta(s,\, q)$ – Hurwitz zeta function, and function $f(s;\, l_3,\, q)$ defined by series

$$f(s;\, l_3,\, q) = \sum_{\substack{n=1\\ n\equiv l_3(mod\ q)}}^{\infty} \frac{\mu(n)}{n^s}, \tag{4}$$

where $\mu(n)$ – Mobius function.

Introduce the notation:

$$\Psi(s,\, \alpha) = \sum_{\substack{n=-\infty\\ n\neq 0}}^{+\infty} \frac{b_n}{|n|^s}, \tag{5}$$

where $b_n = -ie^{2\pi in\alpha} sign(n)$, a $sign(t)$ is a sign of real $t \neq 0$.

Lemma 1. *Let* $\alpha \in (0,\,1)$ *and let* $\Phi(s,\,\alpha) = \zeta(s,\,\alpha) - \zeta(s,\,1-\alpha)$. *Then, in the whole complex* $s-$ *plane we have the functional equation*

$$\pi^{-\frac{s}{2}}\Gamma\left(\frac{1+s}{2}\right)\Phi(s,\,\alpha) = \pi^{-\frac{1-s}{2}}\Gamma\left(1-\frac{s}{2}\right)\Psi(1-s,\,\alpha).$$

The following lemma is an analogue of the second moment of Riemann zeta function.

Lemma 2. *Let* $l,\ q$ *– natural,* $T > T_0 \geq 2$. *Then if* $T \to \infty$ *we have the asymptotic estimate*

$$\int_{T_0}^{T_0+T}\left|q^{-\frac{1}{2}-it}\zeta\left(\frac{1}{2}+it;\ \frac{l}{q}\right) - \frac{1}{l^{\frac{1}{2}+it}}\right|^2 dt \ll \frac{T\log^2(qT)}{q}$$

with constant in the symbol "O*" depending only on* T_0.

We proved a theorem.

Theorem 1. *Let* $a,\ q \in \mathbb{N},\ (a,\,q) = 1$. *Then we have an asymptotic formula*

$$G(x;\,a,\,q) := \sum_{\substack{n\equiv a(mod\ q)\\ n\leq x}} g(n) = \frac{x^2\log x}{q}c_1(q) + \frac{x^2}{q}c_0(q) + O\left(x^{\frac{3}{2}}\,(\log x)^{\frac{3}{2}}\right), \tag{6}$$

where

$$c_1(q) = \prod_{p|q}\left(1-\frac{1}{p}\right)\left(1-\frac{1}{p^2}\right),$$

$$c_0(q) = -\frac{\zeta'(2)}{\zeta^2(2)}\prod_{p|q}\left(1-\frac{1}{p^2}\right)^{-1} + \frac{1}{\zeta(2)}\prod_{p|q}\frac{\log p}{p^2-1}$$

and constant in symbol "O*" does not depend on* $x,\ a,\ q$.

Conclusion. An asymptotic formula resulting in the main theorem (6) is nontrivial for all $q = o\left(x^{\frac{1}{2}}\,(\ logx)^{-\frac{1}{2}}\right)$. Furthermore, we note that this theorem is easy extended to the case $(a,\,q) > 1$.

References

[1] O. Bordelles, A note on the average order of the gcd-sum function, J. Integer Sequences, 10 (2007), Article 07.3.3.

[2] K. A. Broughan, The gcd-sum function, J. Integer Sequences, 4 (2001), Article 01.2.2.

[3] S. S. Pillai, On an arithmetic functions, J. Annamalai Univ., 2 (1937), 243-248.

Algorithms for computing optimal coefficients

Nikolay M. Dobrovolskiy[1], Larisa P. Dobrovolskaya[2], Nikolay N. Dobrovolskiy[3], Nadegda K. Ogorodnichuk[4], and Evgenii D. Rebrov[5]

[1] L.N. Tolstoy Tula State Pedagogical University, pr. Lenina 125, 300026 Tula, Russia (E-mail: dobrovol@tspu.tula.ru)

[2] Institute of Economics and Management, str. Veresaeva 10, 300041 Tula, Russia

[3] Tula State University, pr. Lenina 92, 300012 Tula, Russia

[4] L.N. Tolstoy Tula State Pedagogical University, pr. Lenina 125, 300026 Tula, Russia

[5] Moscow State Pedagogical University, str. M. Pirogovskaya 1, str. 1, 119991 Moscow, Russia

A general algorithm for computing optimal coefficients such that

$$T_{N,A}(a_1,\ldots,a_s) \le T_{N,A} \tag{1}$$

is given in Bocharova[1] for any module N. Here $T_{N,A}(a_1,\ldots,a_s)$ is a generalized logarithmic measure of quality and $T_{N,A}$ is an average generalized logarithmic measure of quality of optimal coefficients modulo N with a constant A.

Theorem 1. *We have*

$$T_{N,A} = A^s \cdot (N-1) + \sum_{p|N} \sum_{\nu=1}^{\nu_p} \left(1 - \frac{1}{p}\right) p^{\nu} \left(\left(A - \frac{2p \ln p}{(p-1)p^{\nu}} \right)^s - A^s \right).$$

Further we define an admissible sequence of prime numbers and a special module. After this we construct an $O(N)$ algorithm to compute values of $a_1, \ldots, a_s$ from reduced residue system modulo $N = p_1 p_2 \ldots p_k$ where $p_1, p_2, \ldots, p_k$ is an admissible sequence of prime numbers such that inequality (1) holds true. This algorithm is based on general algorithm from Bocharova[1].

Let p be a fixed odd prime number greater than 3. For example, any odd prime number greater than 3 and not greater than 100. We say that a monotonically increasing sequence $p_1, p_2, \ldots, p_k$ is an admissible sequence of prime numbers of length k for prime number p if

$$p_1 = p, \quad \frac{6p_1 \ldots p_{j-1}}{6^{j-1}} < p_j < \frac{12p_1 \ldots p_{j-1}}{6^{j-1}} \quad (2 \le j \le k).$$

From Bertrand's postulate we have that for any odd prime number $p > 3$ there exists an admissible sequence of prime numbers of any given length k.

By direct computations using the table of prime numbers it is easy to obtain that for $p = 5$ an admissible sequence of prime numbers of length 6 is $5, 7, 11, 17, 53, 307$. This admissible sequence of prime numbers defines a module $N = 106\,493\,695$.

For any given prime number p there exist several admissible sequences of prime numbers of length k. Among them there are the one minimal admissible sequence $p'_1, p'_2, \ldots, p'_k$ and the one maximal admissible sequence $p''_1, p''_2, \ldots, p''_k$ such that

$$p'_j < p_j < p''_j \quad (3 \le j \le k)$$

for any admissible sequences $p_1, p_2, \ldots, p_k$ for the same p. Thus we obtain the minimal module $N'_{p,k} = p'_1 \cdot p'_2 \cdot \ldots \cdot p'_k$ and the maximal module $N''_{p,k} = p''_1 \cdot p''_2 \cdot \ldots \cdot p''_k$.

For example, for $p = 5$ the minimal admissible sequence of length 6 is $5, 7, 11, 13, 23, 89$, the maximal admissible sequence is $5, 7, 11, 19, 67, 751$ and $N'_{5,6} = 5 \cdot 7 \cdot 11 \cdot 13 \cdot 23 \cdot 89 = 10\,245\,235$, $N''_{5,6} = 5 \cdot 7 \cdot 11 \cdot 19 \cdot 67 \cdot 751 = 368\,068\,855$ respectively.

Let us assume that we have a table of values for the function $\ln\left(2 \sin\left(\pi \left\{\frac{k}{N}\right\}\right)\right)$. This requires $C \cdot N$ operations. Also let us assume that we have all divisors of $N = p_1 \cdot \ldots \cdot p_k$ (there are 2^k of them) and

values of Mobius and Euler functions for all those divisors. This requires no more than $O(\sqrt{N})$ additional operations and no more than $O(\sqrt{N})$ bytes of memory for storage.

Thus the preparation phase of our algorithm requires $O(N)$ elementary operations and $O(N)$ bytes of memory for storage of auxiliary tables.

Theorem 2. *Let $K^*(N)$ be the number of elementary operations to compute the optimal coefficients $a_1, \ldots, a_s$ modulo N of an index s; then*

$$K^*(N) \le s^2 C \cdot N \cdot \left(\frac{7}{2} \cdot p + 32\frac{2}{5}\right) = 5 \cdot C \cdot s^2 \cdot N \cdot \left(\frac{7}{10} \cdot p + 6\frac{12}{25}\right)$$

and hence special modules N have an index 0 with a constant $\frac{7}{10} \cdot p + 6\frac{12}{25}$. There C is the maximum number of elementary operations to compute and use one factor of the form

$$A - 2\ln\left(2\sin\left(\pi\left\{\frac{z_{j\nu}t}{p_\nu}\right\}\right)\right).$$

The following program finds such p-points optimal parallelepipedal net for a given P-points net S that their product is a $P \cdot p$-points optimal parallelepipedal net. An important feature of this program is that an array with the resulting net consists of p subarrays with a modified original net in each of them. We use this feature when we construct an algorithm for numerical integration with stopping rule.

$$\begin{array}{l}
HTS(S,p,s) := \left| \begin{array}{l}
a_{s-1} \leftarrow 1,\ b_0 \leftarrow 1,\ a_0 \leftarrow 1, P \leftarrow rows(S) \\
for\ j \in 1..s-1 \\
\quad \left| \begin{array}{l}
R \leftarrow 3^{j+1} \\
for\ c \in 1..p-1 \\
\quad \left| \begin{array}{l}
b_j \leftarrow c \\
r \leftarrow \frac{3^{j+1}}{p \cdot P} \left[\sum\limits_{n=0}^{P-1} \sum\limits_{k=0}^{p-1} \left[\prod\limits_{l=0}^{j} \left[1 - 2\left(S_{n,l} + \frac{b_l \cdot k}{p} - floor\left(S_{n,l} + \frac{b_l \cdot k}{p} \right) \right) \right] \right]^2 \right] \\
a_j \leftarrow c,\ R \leftarrow r \quad if\ r < R
\end{array} \right. \\
b_j \leftarrow a_j
\end{array} \right. \\
S_{P \cdot p-1, s-1} \leftarrow 0 \\
for\ k \in 1..p-1 \\
\quad for\ n \in 0..P-1 \\
\quad\quad for\ j \in 0..s-1 \\
\quad\quad\quad S_{P \cdot k+n, j} \leftarrow S_{n,j} + \frac{a_j \cdot k}{p} - floor\left(S_{n,j} + \frac{a_j \cdot k}{p} \right) \\
S
\end{array} \right.
\end{array}$$

Finally, we note that N. M. Korobov pointed out the fact that the time complexity to compute optimal coefficients ($O(N)$) is comparable with the time complexity of an quadrature formula and hence is acceptable.

References

[1] L. P. Bocharova (Dobrovolskaya). Algorithms for finding optimal coefficients. Chebyshev Sbornik 8(1):4–109, 2007.

The Steiner's problem. The realization of the modified genetic algorithm based on the consistent inclusion of additional vertices inward the minimum spanning tree

O. A. Gerenko[1], T. A. Gerenko[2], and A. A. Skosarev[3]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: Gerenko_Olga@ukr.net)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: Gerenko_Olga@ukr.net)

[3] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: Gerenko_Olga@ukr.net)

The Steiner's problem belongs to a class of problems, for which, considered by many contemporary researchers, the efficient algorithms, presumably, so never will be found The approximation algorithms of solution are quite often used in various applications of the search problem of the shortest networks. Among them are the designing of integrated electronic circuits, the construction of an evolutionary tree for the group of species and the minimization of material consumption to create telephone lines, pipelines and highways.

The mathematical methods are limited to quantity of points for which the algorithm solves Steiner's problem in a reasonable time. For the growing number of points in the design of the network topology should be used all kinds of heuristic methods for solving this problem. One of such methods is the use of genetic algorithms.

Rectilinear Steiner's tree (RST) is used in telecommunications in the

route optimization of multimedia data transfer. The offered genetic algorithm uses a genetic encoding of the trees in which the symbols from binary and nonbinary alphabets occupy alternative positions. The algorithm realizes the operator of a crossover from [1]. The problem of constructing the minimum cost RST has been shown to be NP-complete [4]. The RST problem has been studied extensively and many heuristic algorithms have been proposed. Special cases of the RST problem have been shown to have polynomial time algorithms [3], [4].

To receive an exact type of algorithm, it is necessary to solve some tasks originally.

1. The encoding of chromosomes.

On n points exists $2^{n-1}n^{n-2}$ Rectilinear Steiner's tree. It means the following: $n-1$ the binary symbol sequences with $n-2$ by symbols, which were chosen from the alphabet of n symbols. The following is the algorithm for the encoding of a certain order of vertices $v_1, v_2, \ldots, v_n$.

1. We initialize the variable i is equal to 1. We initialize the vertex degrees.
2. We specify the first vertex v_0 (according to order). Its degree is equal to 1. (v_0, a_i) is an edge in the spanning tree. Reduce the degree of v_0 and a_i, and increase i.
3. We repeat a step 2 while degree of each vertex isn't equal to 0, except for two vertices, whose degrees are equal to 1. The last edge in the spanning tree connects these two vertices.

A string that encodes the RST, includes binary symbols. These symbols define for each edge of the spanning tree that point from the two possible points which is included in the tree of the point, the value of 0 indicates that it the left Steiner's point, and a value of 1 - the right Steiner's point. The first binary symbol indicates that Steiner's point from the first edge, which is defined by the chromosome. The last binary symbol, which is the last symbol in the chromosome, indicates that Steiner's point, which is associated with the edge of the spanning tree, which is chosen by the algorithm at least.

2. The objective function.

The objective function - is the total length of the edges of RST. The

problem of the algorithm - to minimize the objective function.

3. Crossover operator.

There are two principle crossover operator for this problem: the descendants should contain the structural features (the edges, the Steiner's points) of both parents, the descendants should save all the edges that are common to both parents. [2] The resulting crossover operator generates one descendant from two parents. The operator copies into the descendant all the pairs of straight edges, which comply with the edges of the spanning tree, which the parents divide between each other.

If the edges of the spanning tree differ only in one Steiner's point, then the operator includes these edges of the spanning tree and chooses the Steiner's point arbitrarily. Then, choose the pairs of straight edges arbitrarily from two parents as long as he finishes the formation of a descendant of RST. The operator discards any pairs of edges that create cycles in the descendants. Because each parent encodes RST, then for the construction of a descendant of RST always take the edges of their two parents. There is no need to insert the arbitrary edges.

4. The mutation operator

The mutation operator modifies each position independently, with probability at five percent. Then, the mutation changes the symbol, the new value is chosen from the non-old of values. The algorithm applies two operators separately, and each receives one descendant. The genetic algorithm, uses strings with the mixed encoding with the alternative binary and non-binary symbols to represent elements of the solution problem.

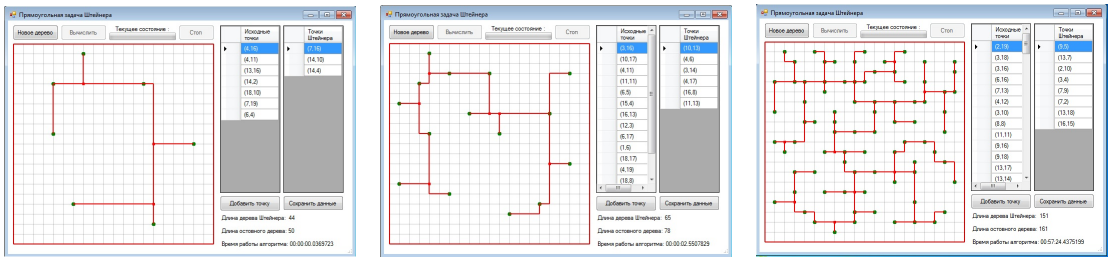


Figure1. Result of work of the program for 7 points,15 points,80 points

Average results of work of the program for different quantity of points are given in table 1.

	Quantity of starting points in Steiner's task					
	before 10	10-20	20-30	30-50	50-80	80-...
Operating time	0 seconds	1-6 seconds	7-40 seconds	40 c - 9 minutes	9 minutes - 1 hour	>1 hour

Table1. Results of work of the program

The program realized in C#. Experimental studies have shown the advantage of using genetic search for the construction of RST compared to iterative methods. On the basis of the data was determined that the experimental time complexity of the algorithm is approximately $O(\alpha\ n3)$, where the α-factor, n-number of vertices of the graph. Adoption of new and modified methods to accelerate the search procedure for finding optimal RST.

References

[1] Емельянов В. В., Курейчик В. В., Курейчик В. М., Теория и практика эволюционного моделирования - М: Физматлит, 2003. - с. 432.

[2] Рутковская Д., Пилиньский М., Рутковский Л., Нейронные сети, генетические алгоритмы и нечеткие системы = Sieci neuronowe, algorytmy genetyczne i systemy rozmyte - 2-е изд.. - М: Горячая линия-Телеком, 2008. - с. 452.

[3] Aho A.V., Garey M.R. and Hwang F.K., "Rectilinear Steiner trees: Efficient special-case algorithm" Networks, vol. 7, pp. 37-58, 1977.

[4] 131 Cohoon J.P., Richards D.S. and Salowe J.S., "A linear-time Steiner tree routing algorithm for terminals on the boundary of a rectangle" in Dig. Technical Papers, ICCAD-88, pp. 402-405, Nov. 1988.

Arithmetic and Computer Algebra of Calabi-Yau - type Varieties

N. M. Glazunov

National Aviation University, Kiev, Ukraine (E-mail: glanm@yahoo.com)

We shall briefly review the applications of computer algebra in arithmetic of Calabi-Yau - type varieties [1, 2]. These include computation of modular symbols and $L-$functions of elliptic curves, arithmetic structures of $K3$ surfaces over fields of finite characteristics, formal groups of Calabi-Yau - type varieties and their $\zeta-$functions, computation of Shafarevich-Tate group of abelian varieties A over the field of algebraic functions with algebraically closed residue field and the $p-$component of the Shafarevich-Tate group of A.

By Calabi-Yau - type variety (CY variety) we understand algebraic variety X over complex numbers with zero canonical class K_X. The one-dimensional CY varieties are elliptic curves and two-dimensional CY varieties are two-dimensional abelian surfaces and $K3$ surfaces.

In arithmetic applications it is desirable to conside these varieties over fields of finite characteristics, over local, quasi-local and quasi-global fields. We also present data structures and algorithms to computation in quasi-global fields.

This is the extension of computer algebral part of my papers [3, 1, 5, 6].

References

[1] Yau S.T., Nadis S. The shape of inner space. String theory and the geometry of the universe's hidden dimensions, New York, Basic Books, 2010, 355 p. .

[2] Shafarevich I.R., Rudakov A.I. $K3$ surfaces over fields of finite characteristics (In Russian). *Itogi nauki i texniki. Ser. Modern Math. Problems*, M.: VINITI, 1981, N 18. P.115-207.

[3] Glazunov N.M. Calabi-Yau manifolds, algebraic geometry and computer algebra methods, *Int. Conf. on Criptography,* Lvov, LPI, 2012, P.138-139.

[4] Glazunov N.M., On Validated Numerics, Category Theory and Computer Algebra Framework for Simulation and Computation in Theoretical Physics, *Nuclear Inst. and Methods in Physics, A, vol.502, Nos.2-3,* 2003, 654-656.

[5] Glazunov N.M., Category Theory Aspects of Complex Manifolds and Problems of Mirror Symmetry (in Russian,) *Problemy Programmirovaniya*. No. 3-4, Kiev, 2002, 104-110.

[6] Glazunov N.M. Some algebraic geometric aspects of Calabi-Yau manifolds and their zeta functions, *Int. Workshop "Computer Algebra and its Applications to Physics (CAAP-2001)"*, Dubna, Russia, 2001. P.18.

Symmetries and Simplification of Tensors

N. M. Glazunov[1] and E. V. Shaxova[2]

[1] National Aviation University, Kiev, Ukraine (E-mail: glanm@yahoo.com)

[2] National Aviation University, Kiev, Ukraine (E-mail: glanm@yahoo.com)

Tensor and tensor polynomials (tensor objects) are used in differential geometry, geometric analysis, celestial mechanics, general relativity and under design of complex systems. The first author have investigated aspects of these constructions in papers [1, 2, 3] (see the papers and references their in). Under investigation, simplification and application of the tensor objects it is desirable to recognize and compute their symmetries.

We consider monoterm symmetries, multiterm symmetries, indexed objects with symmetries, investigate the symmetries and discuss algorithms that put tensor objects to their canonical forms.

Applications to connections on sheaves [2], Riemann and Ricci tensors will be done.

References

[1] N.M. Glazunov, On Validated Numerics, Category Theory and Computer Algebra Framework for Simulation and Computation in Theoretical Physics, *Nuclear Inst. and Methods in Physics, A, vol.502, Nos.2-3,* 2003, 654-656.

[2] N.M. Glazunov, On algebraic geometric and computer algebra aspects of mirror symmetry, *Proc. of the Int. Conf. "Computer Algebra and its Applications to Physics." Dubna-2001.* Joint Institute of Nuclear Research, Dubna, 2002, 104-113.

[3] Glazunov N., Mirror Symmetry: Algebraic Geometric and Special Lagrangian Fibrations Aspects, *Proceedinds of Fourth International Conference "Symmetry in Nonlinear Mathematical Physics" (12–18 July, 2001, Kyiv), Editors A.G. Nikitin and V.M. Boyko*, Kyiv, Insitute of Mathematics, 2002, V.43, Part 2, 623–628.

Responsibilities Matrix

D. Ivanov[1] and F. Novikov[2]

[1] I.I. St. Petersburg State Polytechnical University, Department of applied mathematics Polytechnicheskaya, 29, 195251, St. Petersburg, Russia (E-mail: fedornovikov@rambler.ru)

[2] I.I. St. Petersburg State Polytechnical University, Department of applied mathematics Polytechnicheskaya, 29, 195251, St. Petersburg, Russia (E-mail: fedornovikov@rambler.ru)

Introduction.

Software development is complex challenge [1]. To get a success one needs to predefine a sequence of actions to be done. This predefined sequence of actions is called a process. Let us introduce some terms to be

more specific. We say that the process is necessary one, if it might gain desired result. We say that the process sufficient one, if it guarantees desired result.

There are several necessary processes, known in software engineering [2], each of which has its advantages, but none of them could be referred to as a sufficient process. Software development is a bit like the medicine: necessary processes are established, but final result depends on physician's experience and intuition too much. Make note, that it is not the case in civil engineering, where we do have reliable sufficient processes for solving most problems.

In this paper we analyze the reasons why sufficient software development process hadn't been constructed yet and propose the approach for its construction.

Why software development processes are necessary but not sufficient ones?

Software development process includes a lot of aspects: management, technology, psychology, finances, and so on. Of course, there are several companies and persons who can solve all these problems - otherwise we would not have some excellent software products we do have. On the other hand everyone can easily enumerate software pieces of pour quality, low performance and unreliable behavior. The reason of these drawbacks is not an evil will, but the lack of sufficient software development process. Successful software development and disappointing failures are coexisting at the moment.

In our opinion, the principal cause of this phenomenon is that successful developers till now have no adequate means to formalize and share their expertise to others. This is true especially for technical aspects of development process. Software development experience is not alienable. Once again, make note that in civil engineering it is not the case. Civil engineers have elaborated formalized drawings of various kinds and that is enough to reuse good technical solutions many times.

Thus, to construct sufficient software development process we need to elaborate methods for reuse of successful software development experience. In other words we need software engineering to be engineering in

real life, not only in title.

Current trends in programming methodologies

Let us turn to programming methodologies which constituent the basis for development processes at the moment. First of all, two things catch the eye:

- all methodologies operate with terms "role" and "artifact";
- all methodologies avoid to consider technical aspects of development.

Let us try to find relationship between these observations.

All developers do play certain roles in development process. Each role acts in certain area of responsibility/competence. The result of role's activity is an artifact. For example, an analyst composes a requirements specification and a programmer writes a program code. "Analyst" and "programmers" are roles, and "requirements specification" and "program code" are artifacts.

All methodologies explicitly define relationships between roles and artifacts. For example, waterfall methodology postulates that analyst must prepare requirements specification before the programmer may write the first line of code, and agile methodology says that requirements specification refinement and code development should run in parallel.

But neither waterfall, no agile says what exactly should put analyst into requirements specification and programmer into program code - these technical questions is to be answered by the developer himself. Methodology does not give you any hints - only previous own experience and advices of colleagues could help you, if you lucky to have experience or colleagues.

Thus we see that current methodologies describe the set of roles, involved in the development process, enumerate the set of artifacts, produced by these roles, and put some relationships between roles and artifacts. This is necessary but not sufficient.

Roles and artifacts - an inside look

Most people consider role to be synonym of position title. Common practice of software development project management is to decide what

roles would be involved and to assign persons to the project team. Then project team members would design artifacts according to their competence.

Our approach differs in direction. First of all it should be defined what artifacts are to be produced, and then suitable performers should be found for the roles. Artifacts are primary and roles are secondary. Role is the confirmed responsibility for artifacts of certain kind, no less and no more.

But what is an artifact? Artifact is the result of development process - it is obvious. Requirements specification, program code, UML model - all these are artifacts. The point is that although the artifacts have different outward forms, they have a common inner nature. Let us explain it recalling some observations. Developer produces artifacts according his or her competence and responsibility. Responsibility it is an ability to response (to answer) a queries (a questions). Developer answers the question he or she is competent to response and put the answer into artifact in the certain form.

Thus the artifact of any kind is the set of answers to the certain questions.

Question Driven Development

What is of most importance in development process? We agreed that roles have low importance. Artifacts are depend on answers - having no answers one would have no artifacts. Thus what is the most important is the set of questions one has to answer during development process. What does the system do? How does the system work? What does the system consist of? These are technical questions and these should be answered somehow, otherwise development would fail.

The main point of our approach to sufficient development process is to find the proper place for most important thing - the set of questions to be answered.

The process based on explicit allocation of the set of questions which are to be answered we call Question Driven Development, or QDD for short.

Caution: QDD is not a predefined checklist - it is absolutely impossible to invent universal checklist suitable for any situation. Questions are differing from project to project, and developers have to put specific questions and answer them for each project. The very art of computer programming is to select best questions in every particular case. Our goal is to suggest structure to capture questions of development process.

Responsibility Matrix

Software development is a complex problem in its nature, and this complexity cannot be avoided for free [1]. There might be hundreds and thousands of questions in the project and developers have to manage them all. Of course some classification should be used in this case. We use two orthogonal classifications. First classification is based upon question scope (usage, structure, behavior) and second classification is based upon level of details (system, module, element). Thus we get matrix, which columns are marked with scope of question, and rows are marked with level of details (fig. 1).

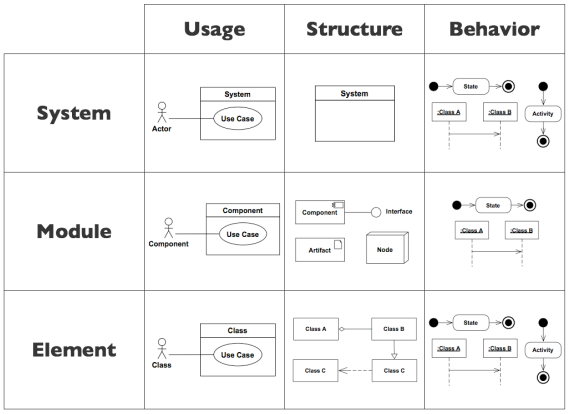


Figure1. Responsibilities matrix filled with UML constructs

We put in matrix cells UML constructions because the UML is the only language at the moment which had proved the ability to capture all technical questions and answers.

Conclusion

Any technical question in couple with its answer fits one of the re-

sponsibility matrix cells. Thus responsibility matrix is convenient and sufficient structure for software development, at least in technical aspects.

References

[1] Brooks F. P. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition), 1995.
[2] Rambough J., Blaha M. Object - Oriented Modeling And Design With Uml, (2nd Edition), Pearson Education, 2007.
[3] Novikov F., Ivanov D. Modeling with UML (in Russian), Nauka i Tekhnika, 2010.

E6 Sockets in Linux

M. A. Kharsun[1] and D. A. Zaitsev[2]

[1] International Humanitarian University, str. Fontaskaya Doroga 33, 65009 Odessa, Ukraine (E-mail: mikefromsky@gmail.com)
[2] International Humanitarian University, str. Fontaskaya Doroga 33, 65009 Odessa, Ukraine (E-mail: zsoftua@yahoo.com)

1. Introduction

In Ukraine, a national stack of networking protocols E6 [1] was developed. The goal of the present work is to present the software implementation of E6 stack via OS Linux socket interface [2]. The main advantage of using sockets is a standard and consistent API for each protocol, implemented via this interface. Basic functions of sockets are contained in libc library, the main header file is sys/socket.h; besides for specific address/protocol families, own header files are added, netinet/in.h in TCP/IP, for example. For stack E6, the file e6.h was created.

2. Interface of the loadable kernel module E6-socket

Almost each OS Linux protocol stack is implemented as a loadable kernel module. To build a kernel module, a special way of compilation and linking with make command is used to access the current kernel symbol tables; the required by command make files makefile and KBuild are the following:

```
Makefile::
obj-m  := e6.o
KERNELDIR := /lib/modules/2.6.31.5-desktop-1mnb/build
all:: $(MAKE) -C $(KERNELDIR) M=‘pwd‘ modules
KBuild::
obj-m  := e6.o
```

E6-socket module parameters are: name of E6 device in OS Linux devices name format E6-devname, E6 device address (put in place of the factory MAC-address) E6-devaddr. An example of loading module command is the following:

```
insmod e6.ko E6_devname=eth0 E6_devaddr=000000000001
```

For E6 address family of Linux sokets the following description from e6.h header file is used

```
typedef uint8_t e6_addr_t[E6_ADDR_LEN];
struct e6_addr
{
        e6_addr_t s_addr;
};
typedef uint16_t e6_port_t;
struct sockaddr_e6
{
__SOCKADDR_COMMON (se6_);
     e6_port_t se6_port; /* E6 port number.  */
     struct e6_addr se6_addr; /* E6 address.  */
     unsigned char se6_zero[sizeof (struct sockaddr)
__SOCKADDR_COMMON_SIZE - sizeof (e6_port_t)
                       - sizeof (struct e6_addr)];
```

```
};
```

Thus E6 socket address sockaddr-e6 consists of E6 host address of 6 octets and E6 port number of 2 octets. System of addresses and ports is independent of other protocol families.

3. Algorithms of E6-socket module basic functions

The sending message function has the form

```
static int e6_sendmsg(
                        struct kiocb *iocb,
                        struct socket *sock,
                        struct msghdr *msg,
                        size_t len
                      )
```

where iocb is the block of input/output control, msg is a message header, len is the message length. Initially, memory is allocated for the packet (frame) buffer

```
skb = sock_alloc_send_skb( sk, len + sizeof(struct e6_hdr),
                           msg->msg_flags & MSG_DONTWAIT, &err);
```

Copying information from the user's buffer to skb block is performed by the memcpy-fromiovec command using the message header msg parameters

```
skb_reserve(skb, sizeof(struct e6_hdr));
err = memcpy_fromiovec(skb_put(skb,len),
                           msg->msg_iov, len);
```

initial part of buffer is reserved by skb-reserve command for subsequent placement of e6-hdr header. Recall that, e6-hdr is a combined network and data-link layers header. Then the packet (frame) header is formed

```
hdr = (struct e6_hdr *)skb_mac_header(skb);
hdr->de6a = daddr;
hdr->se6a = addr;
hdr->type = htons(ETH_P_E6);
hdr->de6p = dport;
hdr->se6p = port;
```

The output device and the priority of the operation is specified

```
skb->dev = e6_dev;
skb->priority = sk->sk_priority;
```

The actual launch of the operation on the device is performed by the sequence of commands

```
dev_queue_xmit(skb);
dev_put(dev);
```

Command dev-queue-xmit inserts the frame buffer skb to the device queue; command dev-put forces the device driver to get started in case of absence of active operation and does not involve any changes otherwise; the driver will check the queue at the end of the active operation.

4. Conclusions

The source files of the kernel module and test applications with brief instructions on how to compile, install and run them are put on the website http://daze.ho.ua and can be used as prototypes for the programmers to develop their own protocol stacks.

References

[1] D.A. Zaitsev, S.I. Bolshakov, E6 Addressing Scheme and Network Architecture. Journal of Advanced Computer Science and Technology, 1 (1) (2012) 18-31.

[2] D.A. Zaitsev, M.A. Kharsun, Implementing Stack E6 via OS Linux Sockets. Journal of Advanced Computer Science and Technology, 1 (3) (2012) 116-133.

Groups with perfect order subsets

S. V. Konyagin

Steklov Institute of Mathematics, 8 Gubkin Street, 119991 Moscow, Russia
(E-mail: konyagin@mi.ras.ru)

If n is a positive integer, p is a prime, $a \geq 0$ is an integer, p^a divides n but p^{a+1} does not divide n, then we write $p^a||n$. Consider the multiplicative function

$$f(n) = \prod_{p^a||n} (p^a - 1).$$

A finite group G is said to have *Perfect Order Subsets* if for every d, the number of elements of G of order d (if there are any) divides $|G|$. This notion was introduced by E. Finch and L. Jones[1]. In the case of finite Abelian groups, the authors reduced the problem of which groups have this property to the case of groups of the form $G = \prod_{i=1}^{k} (\mathbb{Z}/p_i\mathbb{Z})^{a_i}$, where p_i are primes and $a_i \geq 1$. For these groups, it follows from results in [1] that G has *Perfect Order Subsets* if and only if $f(n)|n$.

We prove the following results.

Theorem 1. *If $f(n)|n$ and $n > 2$, then $3|n$.*

Theorem 2. *If $n \in \mathbb{N}$ and $f(n)|n$, then $n/f(n) \leq 85$.*

This is joint work with Kevin Ford and Florian Luca.

The research of Sergei Konyagin was partially supported by Russian Fund for Basic Research, Grant N. 11-01-00329.

References

[1] C. Finch, L. Jones. A curious connection between Fermat numbers and finite groups, Amer. Math. Monthly **109**: 517–524, 2002.

Artificial neural network training by hybrid method based on ant colony algorithm

Eugene V. Kotlyarov[1] and Tatyana I. Petrushina[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: jimis@ua.fm)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: tatyana.petrushina@gmail.com)

The hybrid method of artificial neural network training on the basis of back propagation algorithm and ant colony algorithm is considered. The base of the method is the modified ant colony algorithm which is used for more effective choice of priority ways of movement on the neural network and flexible network scales correction.

The rapid information technologies development involves essential need of efficiency increase of training process for artificial neural networks [1, 2], which are widely used for solving different problems today. Today the classical training methods [3, 4] which have played the most important role in the development of scientific and applied researches need modification. For the neural network training efficiency increase the hybrid algorithm on the basis of ant colony algorithm and back propagation algorithm is suggested.

Process of multilayered perceptron training is considered by the hybrid algorithm. Perceptron consists of three layers: input, one hidden and output. The number of neurons on the layers is n, m and p respectively. Let's define $u = (u_1, u_2, ..., u_n)$ - is input signals vector, $v = (v_1, v_2, ..., v_m)$ - output signals vector of hidden layer, $e = (e_1, e_2, ..., e_p)$ - output signal vector. Matrices $W = \|w_{ij}\|$ dimension $[n * m]$ and $A = \|a_{ij}\|$ dimension $[m * p]$ are used for definition of network scales values between input and hidden layer and between hidden and output layer respectively. Vectors $W_0 = \|w_{0i}\|$ dimension $[1*m]$ and $A_0 = \|a_{0i}\|$ dimension $[1*p]$ are used for definition of network scales offsets on hidden and output layers respectively. The initial values for weights matrices elements and for offset vectors elements are set random numbers from

interval $[-1; 1]$. Signal propagation in the network occurs in the same way as in classic back propagation algorithm. The target signals vector is $c = (c_1, c_2, ..., c_p)$. Mistakes vector count by formula $err = c - e$.

The ant colony algorithm modified by the authors is used for priority ways choosing in neural network while training. The ants number is counted using formula

$$Ants = B * ceil\left(\frac{max(m, p)}{n}\right) * n,$$

where $B \in \mathbb{N}$ - is user parameter to increase or decrease number of ants. Such a number of ants is used to visit all the nodes.

$$ceil(x) = \begin{cases} [x] + 1, x \notin Z, \\ [x], x \in Z, \end{cases}$$

where $[x]$ is the whole part of number x. Ants are uniformly allocated on n of input nodes. Each ant visits 3 nodes on one node on each layer. Ants use pheromones to communicate with each other and to choose the priority ways. Ants mark the visited ways by pheromones. On each way part one ant sprays some pheromones, which the following ants use for priority way choosing.

The following method is used at least once to ensure the visit of all nodes of the hidden and target layers. At the first step the choice is made only among those ways which haven't been visited yet by any ant if such ways exist, or on all ways. At the second and subsequent steps the choice is made among the ways marked with the maximum quantity of the pheromones. If there are some available ways ant chooses the way randomly.

Pheromones redistribution in network ways begins when all the ants finish their travels. Pheromones quantity which each ant dispose on the two edges visited by it depends on mistake value in target node to which ant came. Pheromones quantity which each ant disposes on the edges visited by it is calculated for output node i to which ant came by the formula:

$$pher_i = \frac{Q}{|err_i|}, i = 1, 2, ..., p,$$

where Q - algorithm parameter which is set in each samples database, the pheromone value of one ant. When the error value equals zero ($err_i = 0$) the $pher_i$ value is defined as a number bigger than maximum of the counted values. In the suggested modified algorithm, as in the classical one, the reception of "pheromones evaporation" is used to reduce probability of visiting nodes with great mistakes and to increase probability of visiting nodes with small mistakes. Matrices $\Phi = \|\varphi_{ij}\|$ and $\Psi = \|\psi_{ij}\|$ are used for definition of the pheromone level values in network ways between input and hidden and hidden and output layers respectively. The initial values for pheromones matrices elements are set to zero. Pheromones updating in network is made by formulas:

$$\varphi^*_{ij} = (\varphi_{ij} + pher_k) * (1 - \rho),$$
$$\psi^*_{jk} = (\psi_{jk} + pher_k) * (1 - \rho),$$

where ρ - pheromone evaporation coefficient, $\rho \in [0; 1)$, i, j, k - are fixed ant visited nodes on input, hidden and output layers respectively.

Ant travels repeat until on two sequential iterations the ants' ways change. The algorithm begins to update the network weights and offsets when the ants stop their traveling.

heromones accounting function introduced by authors for more flexible accounting of pheromones quantity when updating network weights $\eta = \eta(x)$ with following properties $\eta \in ([0; 1])$, is strictly monotonously decreasing $\eta(1) = 1$, $\eta(0) = \eta_0$, $\eta_0 > 1$. Define the sum of Φ and Ψ matrices elements as:

$$\Phi_0 = \sum_{i=1}^{n} \sum_{j=1}^{m} \varphi_{ij}, \qquad \Psi_0 = \sum_{i=1}^{m} \sum_{j=1}^{p} \psi_{ij}.$$

Network weights correction coefficients are calculated by:

$$\delta_{ij} = (c_i - e_i)e_i(1 - e_i)\eta\left(\frac{\psi_{ij}}{\Psi_0}\right),\ i = 1, 2, ..., m,\ j = 1, 2, ..., p$$

- for hidden/output layer ways,

$$\delta'_{ij} = \sum_{k=1}^{p} a_{jk}\delta_{jk} \cdot v_i(1 - v_i)\eta\left(\frac{\varphi_{ij}}{\Phi_0}\right),\ i = 1, 2, ..., n,\ j = 1, 2, ..., m$$

- for input/hidden layer ways.

Network weights and offsets are updating by formulas:

$$a_{ij}^* = a_{ij} + \theta\delta_{ij}e_j,\ i = 1, 2, ..., m,\ j = 1, 2, ..., p,$$

$$w_{ij}^* = w_{ij} + \theta\delta'_{ij}u_i,\ i = 1, 2, ..., n,\ j = 1, 2, ..., m,$$

$$a_{0i}^* = a_{0i} + \theta a_{0i} \max_{j=1,2,...,m} \delta_{ji},\ i = 1, 2, ..., p,$$

$$w_{0i}^* = w_{0i} + \theta w_{0i}\frac{1}{n}\sum_{j=1}^{n}\delta'_{ji},\ i = 1, 2, ..., m,$$

where θ - is network training coefficient influencing training speed.

Training artificial neural network hybrid algorithm based on ant colony algorithm and back propagation algorithm is offered. The modified ant colony algorithm is used for more effective choice of the movement ways on neural network and for network weights change by the pheromones. Experiments results on various databases show acceleration of neural network training process in comparison with classical back propagation algorithm.

References

[1] S. Haykin. Neural Networks. A Comprehensive Foundation. Pearson Education 2nd edition 2004.

[2] Anil K. Jain, Jianchang Mao, K.M. Mohiuddin Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, March/1996, pp. 31-44. Translated from the original English version and reprinted with permission. (IEEE).

[3] Werbos P. J., Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.

[4] Rumelhart D.E., Hinton G.E., Williams R.J., Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318 - 362. Cambridge, MA, MIT Press. 1986.

Joint distribution of the Riemann zeta-function

Antanas Laurinčikas

Vilnius University, Naugarduko str. 24, LT-03225 Vilnius, Lithuania (E-mail: antanas.laurincikas@mif.vu.lt)

It is well known that the Riemann zeta-function $\zeta(s)$ has a limit distribution in the sense of weakly convergent probability measures. More precisely, let $\mathcal{B}(S)$ denote the class of Borel sets of the space S. Then for $\sigma > \frac{1}{2}$,

$$\frac{1}{T}\mathrm{meas}\left\{t \in [0,T],\ \zeta\left(\sigma + it\right) \in A\right\}, \quad A \in \mathcal{B}(\mathbb{C}),$$

converges weakly to the explicitly given probability measure P_σ on $(\mathbb{C}, \mathcal{B}(\mathbb{C}))$ as $T \to \infty$.

If $\sigma = \frac{1}{2}$, then a normalization is needed. In this case,

$$\frac{1}{T}\mathrm{meas}\left\{t \in [0,T],\ \zeta\left(\frac{1}{2} + it\right)^{(2^{-1}\log\log T)^{-\frac{1}{2}}} \in A\right\}, \quad A \in \mathcal{B}(\mathbb{C}),$$

converges weakly to lognormal probability measure on $(\mathbb{C}, \mathcal{B}(\mathbb{C}))$.

Similar results are also true for $|\zeta(s)|$.

In the report, we discuss the limit theorems for $(|\zeta(s)|, \zeta(s))$ when $\sigma > \frac{1}{2}$ and $\sigma = \frac{1}{2}$. For this, we apply the method of the m-characteristic transforms of probability measures on $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ [1], where $\mathbb{X} = \mathbb{R} \times \mathbb{C}$. Moreover, we investigate the dependence between $|\zeta(s)|$ and $\zeta(s)$ in terms of m-characteristic transforms.

Suppose that the functions $f_1(t)$ and $f_2(t)$ are defined on $\mathbb{R}$ with values in $\mathbb{R}$ and $\mathbb{C}$, respectively, and that, as $T \to \infty$,

$$\frac{1}{T}\mathrm{meas}\left\{t \in [0,T],\ f_1(t) \in A\right\}, \quad A \in \mathcal{B}(\mathbb{R}),$$

$$\frac{1}{T}\mathrm{meas}\left\{t \in [0,T],\ f_2(t) \in A\right\}, \quad A \in \mathcal{B}(\mathbb{C}),$$

and

$$\frac{1}{T}\mathrm{meas}\left\{t \in [0,T],\ (f_1(t), f_2(t)) \in A\right\}, \quad A \in \mathcal{B}(\mathbb{X}),$$

converges m-weakly to P_{f_1}, P_{f_2} and $P_{(f_1,f_2)}$, respectively. Denote by $v_{f_1,m}(\tau)$, $m=0,1$, $v_{f_2}(\tau,k)$ and $\left(v_{f_1,m}(\tau), v_{f_2}(\tau,k), v_{(f_1,f_2),m}(\tau_1,\tau_2,k),\right.$ $\left.m=0,1\right)$ the characteristic transforms and m-characteristic transforms of the measures P_{f_1}, P_{f_2} and $P_{(f_1,f_2)}$, respectively. Let

$$V_m(f_1(t),f_2(t)) \stackrel{def}{=} \sup_{\substack{\tau_1,\tau_2\in\mathbb{R}\\ k\in\mathbb{Z}}} \left|v_{(f_1,f_2),m}(\tau_1,\tau_2,k) - v_{f_1,m}(\tau_1)v_{f_2}(\tau_2,k)\right|,$$

$m=0,1$.

Then, for example, we obtain that

$$V_m\left(\left|\zeta\left(\frac{1}{2}+it\right)\right|^{(2^{-1}\log\log T)^{-\frac{1}{2}}}, \zeta\left(\frac{1}{2}+it\right)^{(2^{-1}\log\log T)^{-\frac{1}{2}}}\right) = 1,$$

$m=0,1$.

References

[1] A. Laurinčikas, R. Macaitienė. The characteristic transforms on $\mathbb{R}\times\mathbb{C}$. Integral Transforms and Special Functions 19(1):11-22, 2008.

Exponential divisors function over Gaussian integers

A. V. Lelechenko

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: 1@dxdy.ru)

We introduce the exponential divisors function over the Gaussian integers $\mathbb{Z}[i]$ in two different ways and study their asymptotic behaviour.

Let $\tau^{(e)}$ (so called *exponential divisors function* [1]) be a multiplicative function such that $\tau^{(e)}(p^\alpha) = \tau(\alpha)$, where $\tau(n)$ denotes the number of divisors of n. Properties of $\tau^{(e)}$ and especially its asymptotic behaviour were widely studied last years (for example, [2]).

We are going to introduce the exponential divisors function over the Gaussian integers $\mathbb{Z}[i]$. Let us denote $\mathfrak{t}^{(e)}\colon \mathbb{Z}[i] \to \mathbb{N}$ such that it is multiplicative over $\mathbb{Z}[i]$ and for prime Gaussian integer $\mathfrak{p}$ we have $\mathfrak{t}^{(e)}(\mathfrak{p}^\alpha) = \tau(\alpha)$. Here τ stands for the usual divisors function $\tau\colon \mathbb{N} \to \mathbb{N}$.

One can also define modified exponential divisors function $\mathfrak{t}_*^{(e)}(\mathfrak{p}^\alpha) = \mathfrak{t}(\alpha)$, where $\mathfrak{t}$ stands for the function $\mathfrak{t}\colon \mathbb{Z}[i] \to \mathbb{N}$, counting Gaussian integer divisors.

We have proved the following theorems.

Theorem 1.

$$\limsup_{N(\alpha)\to\infty} \frac{\log \mathfrak{t}^{(e)}(\alpha) \log\log N(\alpha)}{\log N(\alpha)} = \frac{\log 2}{4},$$

$$\limsup_{N(\alpha)\to\infty} \frac{\log \mathfrak{t}_*^{(e)}(\alpha) \log\log N(\alpha)}{\log N(\alpha)} = \frac{\log 2}{2}.$$

Theorem 2. *Let $F(s)$ be Dirichlet series for $\mathfrak{t}^{(e)}$. Then*

$$F(s) = Z(s)Z(2s)Z^{-1}(5s)G(s),$$

where $G(s)$ is regular for $\Re s > 1/6$ and Z denotes Hecke zeta-function.

Theorem 3. *Let $F_*(s)$ be Dirichlet series for $\mathfrak{t}_*^{(e)}$. Then*

$$F_*(s) = Z(s)Z^3(2s)Z^{-2}(3s)Z(4s)Z(5s)H(s),$$

where $H(s)$ is regular for $\Re s > 1/6$.

Theorem 4.

$$\sum_{N(\alpha)\le x} \mathfrak{t}^{(e)}(\alpha) = Cx + O(x^{1/2+\varepsilon}),$$

where C is the computable constant.

Theorem 5.

$$\sum_{N(\alpha)\le x} \mathfrak{t}_*^{(e)}(\alpha) = Dx + O(x^{1/2+\varepsilon}),$$

where D is the computable constant.

References

[1] Subbarao M. V., On some arithmetic convolutions. The theory of arithmetical functions, Lecture Notes in Mathematics **251,** Springer Verlag, 1972. — 247–271.

[2] Wu J. Probleme de diviseurs exponentiels et entiers exponentiellement sans facteur carre. // J. Theor. Nombres Bordeaux **7.** — 1995. — P. 133–141.

Means of automatic detection of functional dependencies

I. M. Lisitsyna[1] and O. A. Ponyatovsky[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: irina.lisitsyna@gmail.com)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: yagdee@gmail.com)

As is known, essential requirements to information stored in a database are referential integrity and consistency. In the context of relational databases, meeting these requirements needs normalized relations.

A normalization process is a formal method that can be described using the notion of a dependency such as a functional dependency [1]. However, functional dependencies are explained through a domain, thus there is no way to automate their detection at database design phase. That's exactly why modern data modeling software, such as Sybase PowerDesigner or IBM Rational Software Architect, lack normalization means.

In this document, we describe the diagramming tool for building graph of functional dependencies for relations in a relational database, whereby used to understand what normal form relationship uses.

A functional dependency can be determined from:

- a database schema, i.e. from its relationships, attributes, primary and alternate keys, unique indexes;
- a database's contents, using a statistical approach to generate probabilistic functional dependencies. The point at issue is that normalization of existing database needs to be re-evaluated.

We use triplet $\langle X, A, p\rangle$ as a probabilistic functional dependency [3], where X is a subset of attributes of relationship R, A is its attribute, p is the probability that a trivial $X \to A$ exists.

It is proposed that probability p for every possible combination X, A of relationship R is rated, unless a functional dependency was detected earlier, or if a set X was found to be a superset of some candidate keys. We fit into model relations that exist with a probability greater than a certain parameter p_0, and analyze together with relations formed previously.

In order to find a numeric probability p of existence for a functional dependency of attribute A on a set X of attributes, the following algorithm is proposed:

- across all unique values V_X of a set X of attributes, we learn a value V_A of attribute A that is the most widespread;

- we calculate a probability that functional dependency $X \to A$ exists among tuples having V_X value of X attributes: $p(X \to A, V_X) = \frac{|V_X, V_A|}{|V_X|}$, where $|V_X, V_A|$ is a number of tuples having values V_X corresponding to V_A values, $|V_X|$ - a number of tuples having V_X value of X attributes;
- we calculate probability value p as an average among all the probabilities, calculated in the previous step. Formally, if D_X is a set of unique values of a set X of attributes, then probability of existence of a functional dependency can be given by:

$$p(X \to A, R) = \frac{\sum\limits_{V_X \in D_X} p(X \to A, V_X)}{|D_X|}.$$

References

[1] C. J. Date. An Introduction to Database Systems, pp.1328, 2005.

[2] Y. Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies // The Computer Journal, Vol. 42(2), p. 100-111.

[3] Daisy Zhe Wang, Xin Luna Dong, Anish Das Sarma, Franklin M. Halevy A. Functional dependency generation and applications in Pay-As-You-Go data integration systems // Conference: International Workshop on the Web and Databases - WebDB, 2009.

[4] A. Troelsen. Pro C# 2010 and the .NET 4 Platform, pp.1392, 2011.

A Multidimensional Limit Theorem for Zeta Functions of Newforms

Renata Macaitienė[1]

Šiauliai University, Višinskio str. 19, 77156 Šiauliai, Lithuania (E-mail: renata.macaitiene@mi.su.lt)

For $j = 1, ..., r$, let $F_j(z)$ be a normalized newform of weight κ_j and level l_j with the Fourier series expansion

$$F_j(z) = \sum_{m=1}^{\infty} c_j(m) \mathrm{e}^{2\pi imz}, \quad c(1) = 1.$$

We consider the asymptotic behaviour of the collection $(\zeta(s_1, F_1), ..., \zeta(s_r, F_r))$, $s_j = \sigma_j + it_j$, of zeta-functions attached to the forms F_j, and defined by

$$\zeta(s_j, F_j) = \sum_{m=1}^{\infty} \frac{c(m)}{m^{s_j}}, \quad \sigma_j > \frac{\kappa_j + 1}{2}, \quad j = 1, ..., r.$$

Denote by $meas\{A\}$ the Lebesgue measure of a measurable set $A \subset \mathbb{R}$, let $D_j = \{s \in \mathbb{C} : \sigma > \frac{\kappa}{2}\}$, and let $H(D_j)$ stand for the spaces of analytic functions on D_j equipped with the topology of uniform convergence on compacta. Define $H^r = H^r(D_1, ..., D_r) = H(D_1) \times ... \times H(D_r)$. Then we have the following statement.

Theorem 1. *On $(H^r, \mathcal{B}(H^r))$, there exists an explicitly given probability measure P such that*

$$\frac{1}{T} meas \left\{\tau \in [0, T] : (\zeta(s_1 + i\tau, F_1), ..., \zeta(s_1 + i\tau, F_r)) \in A\right\}, \ A \in \mathcal{B}(H^r),$$

converges weakly to P as $T \to \infty$.

Here $\mathcal{B}(H^r)$ denotes the Borel σ-field of the space H^r.

[1] Partially supported by the European Commission within the Seventh Framework Programme FP/2007-2013 project INTEGER (Institutional Transformation for Effecting Gender Equality in Research) under Grant Agreement No. 266638

Determination of correspondence between the projections of universal entities on various subject domains

Eugene V. Malakhov[1] and Maria G. Glava[2]

[1] Odessa National Polytechnic University, 1, Shevchenko Ave., 65044 Odessa, Ukraine (E-mail: opmev@mail.ru)

[2] Odessa National Polytechnic University, 1, Shevchenko Ave., 65044 Odessa, Ukraine (E-mail: mikulinska@mail.ru)

The focus of the study is the problem of uniting SD (subject domain) models on the base of a comparison of the relevant properties of their objects as universal entities' projections on these SDs. The properties are grouped according to measuring scales types of their values and specific comparison techniques are applied to each of the groups. The properties of the ordinal and numeric types are requested to be compared by statistical methods and them of the nominal type by neural networks. While comparing the properties of the ordinal type in order to let converge the correspondence of instances and to equalize their number, it is requested to add blank instances to one or another projection of an entity (an object).

Under nowadays' difficult economic conditions, it is often required to restructure the companies: by merging several companies into a larger enterprise or by dividing an enterprise into smaller-scale ones. A new structure shall require significantly changing the company's various units, which will result in significant changing of the information system (IS). In most cases, it is not reasonable to develop a new IS because of the danger to lose the data having been piled over the entire enterprise life period. The way out should be modernization, merger or division of existing ISs.

Let us consider the union of ISs, which primarily requires uniting the data stored in each of them. Each of the merged ISs shall be a subject domain (DD), which, if combined into a single SD, shall be called a subject subdomain (SSD). Design of any database is based on building

a SD model. Thus, the set task shall be reduced to integrating the SSD models [1].

In order to avoid the data redundancy and inconsistency, it is necessary to identify similar objects in different SSDs as projections of the same universal entity [2]. Therefor, it is required to distinguish each object's most relevant properties that describe a specific SSD. Then, it is required to build the universal entity's projection of a higher-order SD, by comparing the properties. The aggregate of such projections of the corresponding universal entities will be a SD of the following level.

The enlarged algorithm comparing the properties of objects (entities) is shown in Fig. 1.

The relevance of the properties is required to be determined by Shannon's formula for calculating the amount of information for events with different probabilities, which shall be determined by experts. The properties are ranked by amount of information. The properties with amount of information are less than a specified threshold, should be discarded.

Techniques for comparing the properties of selected entities differ depending on their types. Therefore, it is required to group the properties into three sets, without changing their rank: ordinal, nominal, numeric. Then it is required to sort the instances.

It is required to start comparing the properties by comparing the ordinal properties. It is assumed that, after determining the relevance, all remaining properties, even of this type, shall bear a serious meaning loading (i.e. they shouldn't be just a counters). Therefore, the probability is large that the discrepancy between values of the two compared properties of different objects assumes skipping or loss of instances of the corresponding object. To fix this fact, it is required to add blank instance into a specific projection of the entity (object), that will let to converge the correspondence of instances and equalize their number. Then, using any statistical method (e.g. correlation) it is required to compare the properties. If the result is unsatisfactory, it is necessary to produce the same operations with other ordinal properties. If the similarity is not found, the further searching will not reasonable, and it can be concluded that the compared objects are actually projections of different universal entities on various SSDs.

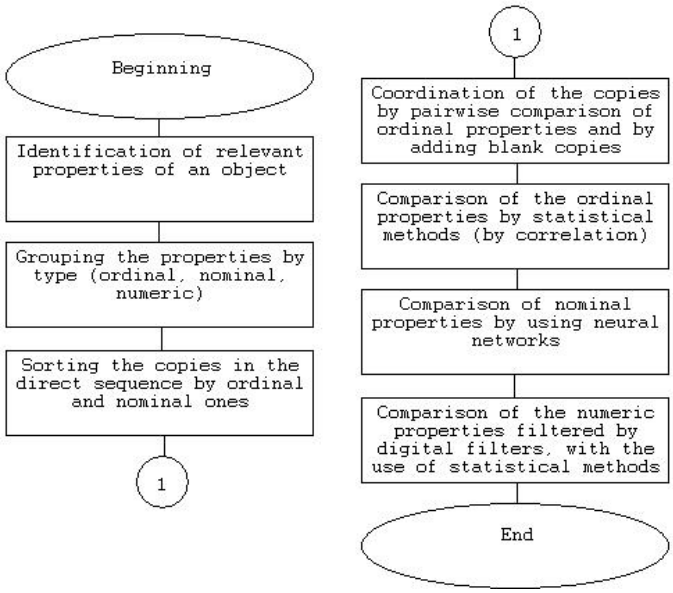


Figure1. The enlarged algorithm comparing the properties of objects (entities)

The properties of the nominal type are proposed to compare by using neural networks. Subject to coincidence of the amount of relevant nominal properties, it is required to train a neural network on the base of the nominal properties of the one among the objects for which the similarity is sought and then to verify the correspondence of the second object. If the number of properties does not coincide or the first comparison has yielded a negative result, it is required to train a neural network for one of the properties and then to compare with each of the nominal properties of the potentially similar projection of the entity. If the maximum similarity is found, it is need to make re-ranking of nominal properties and then sorting of instances taking into account the new order of nominal properties. The maximum number of these operations repetitions can match with exhaustive search of nominal properties.

The numeric properties of an entity are proposed be compared by using statistical methods. To this end the digital filters based, for example, on wavelet or discrete cosine transform, have be applied previously to numeric properties, which then will be compared by any statistical method.

After comparing all the relevant properties, the correlation of the similar properties to their total amount should be considered. After

comparing with a given threshold, it is concluded whether the given objects are projections of the same universal entity.

References

[1] Malakhov E. V., Glava M. G. The conceptual DB scheme design is based on union of the subject subdomains models, Materials of seventh international scientifically-practical conference "Modern technologies of operation of business and an opportunity of use of information systems: a condition, problems, prospects" (from March, 30-th, to March, 31-st, 2012), Odessa I.I. Mechnikov national university.- P. 219-220 (in Russian).

[2] Malakhov E. V. Manipulation of the subject subdomains metamodels, Eastern-european journal of enterprise technologies, Kharkov, 5/3 (29), pp. 6-10, 2007, (in Russian).

The formal pragmatics-semantics modeling for ontology development of Pragmatic Web

I. E. Mazurok

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: igor@mazurok.com)

Development of modern knowledge based information systems requires the solution of several problems associated with semantic and pragmatic analysis and synthesis. Such problems appear not only in systems with natural language interface, but also in the process of formation, analysis and conclusions in the construction of content of the knowledge based information system. Consider the typical problems en-countered with this context in the systems that store and process information in textual or audio form [1], [2].

Cause emerging problems lies in the area of the properties of cognitive structures formed in the content of the intelligent system [1], [3]. The development of pragmatic and semantic systems ontology [4] is one of the key challenges for the identification and formation of models of the cognitive structures. A study of pragmatic characteristics and development of ontologies using a fully algorithmic methods have fundamental limitations because of intervention of the subjective factor. The task is especially difficult when the domain of values of semantic functions are subjective emotional category (emotional semantics of music [2]).

It is proposed to use a large amount accumulated in the computers of discourse material - narrative texts, scores of musical works to meet the challenges of the algorithmic construction of formal ontologies in textual and audio (musical notational) systems. Pseudo formal ontology can be extracted by methods of cluster analysis of proposals and phrases, as the accumulated discourse corps is a reflection of historical ontology. Of course, the selection of discourse for inclusion in the analysis should take into account lexical, historical, genre and other characteristics. The cluster analysis and methods of semantic classification of word usage in a large volume texts (like the actual texts and music scores) make it possible to construct a multi-layered ontology without the involvement of experts.

Will show abilities and constraints of the method of formal ontologies, which is offered. We denote ${}^{n}\Lambda = \left\{\lambda_i \left| i = \overline{1, n_\lambda}\right.\right\}$ base set of ontology - the set of basic facilities for its construction (word or musema). Then the natural ontology can be called any pair of this type ${}^{n}\mathrm{O} = \langle {}^{n}\mathrm{B}\left({}^{n}\Lambda\right), \Sigma\rangle$, where the base ontology ${}^{n}\mathrm{B}\left({}^{n}\Lambda\right) \subset \tilde{\beta}\left({}^{n}\Lambda\right)$ is an improper subset of the hyperboolean of the base set, and the semantic function $\Sigma : {}^{n}\mathrm{B}\left({}^{n}\Lambda\right) \to {}^{n}\Lambda$ function refers to the semantic category. Under the hyperboolean corresponding order shall mean an object defined by the following recurrence relation

$$
\begin{aligned}
{}^{1}\mathrm{B}\left(A\right) &= 2^{A} \\
{}^{n+1}\mathrm{B}\left(A\right) &= 2^{{}^{n}\mathrm{B}(A)}
\end{aligned}
$$

The first-order formal ontology ${}^{1}\mathrm{O} = \left\langle {}^{1}\mathrm{B}\left({}^{n}\Lambda\right), \Sigma\right\rangle$ does not operate the structures of semantic categories and does not support the naming. That is, semantic categories are simple subsets of the base set ${}^{1}\mathrm{B}\left({}^{n}\Lambda\right) \subset$

$\beta\left({}^{n}\Lambda\right)$ and some of the numbering $\mu:\ {}^{1}\mathrm{B}\left({}^{n}\Lambda\right)\to\mathrm{N}$ is given for them. Obviously, the construction of higher-order formal ontologies require sequential detailization the structure of ontologies.

The absence of naming categories and structures allows the formal construction of a system of first-level ontology, including ontology level goals, which is important for the Pragmatic Web [5]. Having ontologies focused on specific subject areas provides the basis for constructing models of semiotic texts, followed by separation of actants.

Further the problem of constructing ontologies related with the definition of the category. We generate the category (formal pseudoconcept) of the first order, which directly operate on the basic elements of the set.

$$\begin{array}{l}{}^{1}k=\left\langle {}^{1}\mathrm{B}\left(\Lambda\right),\Sigma\right\rangle\\ {}^{1}\mathrm{B}\left(\Lambda\right)\in 2^{\Lambda};\\ \Sigma:\begin{cases}2^{\Lambda}\to\Lambda\cup\Theta;\\ 2^{\Lambda}\to N,\end{cases}\end{array}$$

where ${}^{1}\mathrm{B}\left(\Lambda\right)$ specifies volume of formal concept, Σ is a name or identification function. That is, the formal pseudo-concept is a image of the real subject's concept, which reflected in the content of the intellectual system and named by the subject or algorithmically identified by the system. This term must be introduced in order for not to use semantic references outside the formal system that would make automated analysis impossible.

Now run the structural details of the naming of categories, highlighting the concept of a name ($name$), its definition (def), and a unique identifier (id)

$$\begin{array}{l}\Sigma=\Sigma\left\langle def,name,id\right\rangle\\ \begin{cases}def:2^{\Lambda}\to\Lambda\cup\Theta;\\ name:2^{\Lambda}\to\Lambda;\\ id:2^{\Lambda}\to N,\end{cases}\end{array}$$

Based on the above, we can formulate the problem of constructing a formal pseudo-categorization of 1st order

$${}^{1}\mathrm{K}=\left\{{}^{1}k=\left\langle {}^{1}\mathrm{B}\left(\Lambda\right),\Sigma=\left\langle def,name,id\right\rangle\right\rangle\right\}$$

The next step requires to involve contextual semiotic analysis of templates of basic set elements using. This method constructed on the basis ideas of FCA [6]. We distinguish syntactic role scheme, based on a priori data on the nature of the elements of a generalized discourse. Each syntax scheme is given predicate. Role-syntactic scheme a second-order ${}^1Schema = \left\{ {}^1P_i^{(n_i)}\left(x_1, x_2, \cdots, x_{n_i}\right)\right\}$ specify syntactic relations for the elements of a generalized discourse. Role-syntactic scheme of order 2 is obtained as of arguments the sets ${}^2Schema = \left\{ {}^2P_i^{(n_i)}\left(x_1, x_2, \cdots, x_{n_i}\right) \;\middle|\; x \in 2^{\Lambda}\right\}$ and define more general syntactic relations between the sets of objects.

If scheme are set by means of fuzzy logic, the corresponding predicates take the form:

$$\begin{aligned} &{}^mP_i^{(n_i)}\left(x_1, x_2, \cdots, x_{n_i}\right) : \Lambda^{n_i} \to [0;1]\,, \\ &{}^mP_i^{(n_i)} \in {}^mSchema, \\ &m \in \{1, 2\} \end{aligned}$$

The proposed approach allows to extend conception Pragmatic Web through the formalization of ontologies and formal modeling of goal.

References

[1] Мазурок І.Є. Формальний опис класу інформаційних систем, заснованих на знаннях //Збірник статей: Нові інформаційні технології навчання в учбових закладах України, 1998. – випуск VI. – С.217-226.

[2] Мазурок И.Е, Молчанюк Ю.В. Моделирование аудио информации в интеллектуальных системах // Нові інформаційні технології в навчальних закладах України, 2003. - №9, ч. 2-с. 23-29.

[3] Мазурок І. Є. Інтелектуальні інформаційні системи з розвиненою моделлю захисту і авторизації// Вісник Запорізького державного університету. Сер. фіз.-мат. наук. – 2002.– №1.– С.65-68.

[4] ISO/TS 15926-8:2011. Industrial automation systems and integration – Integration of life-cycle data for process plants including oil

and gas production facilities – Part 8: Implementation methods for the integration of distributed systems: Web Ontology Language (OWL) implementation

[5] Aldo de Moor, Mary Keeler and Gary Richmond. Towards a Pragmatic Web. Proceeding ICCS '02 Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces.- 2002.- Pages 235-249.

[6] Uta Priss. Formal Concept Analysis in Information Science// In: Cronin, Blaise (ed.), Annual Review of Information Science and Technology. Vol 40, 2006, pp. 521-543

On supersolvability of finite groups with $\mathbb{P}$-subnormal subgroups

V. S. Monakhov

Francisk Skorina Gomel State University, Sovetskaya str., 104, 246019, Gomel, Republic of Belarus (E-mail: `Victor.Monakhov@gmail.com`)

We consider finite groups only. The Huppert theorem [1, VI.9.5] asserts that a finite group G is supersolvable if and only if every maximal subgroup of G has prime index. It follows that if H is a proper subgroup of a supersolvable group G, then there exists the chain of subgroups

$$H = H_0 \subset H_1 \subset \ldots \subset H_n = G \tag{1}$$

such that $|H_{i+1} : H_i|$ is prime for all i.

A. F. Vasilyev, T. I. Vasilyeva and V. N. Tyutyanov in [2] introduced the following definition. Let $\mathbb{P}$ be the set of all prime numbers. A subgroup H of a group G is called $\mathbb{P}$-*subnormal* in G whenever either $H = G$ or there is a chain of subgroups (1) such that $|H_i : H_{i-1}|$ is prime for all i.

It is clear that a subgroup with $\mathbb{P}$-*subnormal* maximal subgroups is supersolvable. But a group whose 2-maximal subgroups are $\mathbb{P}$-*subnormal*

may be nonsupersolvable. These groups are studied in [3] where particularly the following proposition is proven: *every 2-maximal subgroup of a group G is $\mathbb{P}$-subnormal if and only if $\Phi(G^{\mathfrak{U}}) = 1$ and every proper subgroup of G is supersolvable.* Here $G^{\mathfrak{U}}$ is the smallest normal subgroup of G for which the corresponding factor group is supersolvable and $\Phi(G^{\mathfrak{U}})$ is the Frattini subgroup of $G^{\mathfrak{U}}$. For example, $[E_{7^2}]S_3$, where E_{7^2} is an elementary abelian group of order 7^2 and S_3 is the symmetric group of order 6 is a group with this property. All subgroups of this group except S_3 and each of its conjugates are $\mathbb{P}$-subnormal. A group with $\mathbb{P}$-subnormal p-subgroups may also be nonsupersolvable, see [2], [4].

Therefore quite naturally there appears a problem of finding a system of subgroups in a group, which $\mathbb{P}$-subnormality guarantees supersolvability of the whole group. In this direction the following results are obtained:

Theorem. 1. *A group is supersolvable if and only if the normalizers of all of its Sylow subgroups are $\mathbb{P}$-subnormal.*

2. *A group is supersolvable if and only if all of its Hall subgroups are $\mathbb{P}$-subnormal.*

3. *A group is supersolvable if and only if all of its p-subgroups and all of its biprimary noncyclic z-subgroups are $\mathbb{P}$-subnormal.*

Recall that z-group is a group with cyclic Sylow subgroups. A group G is called primary if $|\pi(G)| = 1$ and biprimary if $|\pi(G)| = 2$. Here, $|\pi(G)|$ is the total number of distinct prime divisors of $|G|$. The set of all prime divisors of $|G|$ is denoted $\pi(G)$. We write $[A]B$ for a semidirect product with a normal subgroup A.

Example 1. In general, a group with $\mathbb{P}$-subnormal cyclic p-subgroups and $\mathbb{P}$-subnormal noncyclic biprimary z-subgroups may be nonsupersolvable. The group

$$G = [E_{5^2}](\langle a\rangle\langle b\rangle),\ |a| = |b| = 4.$$

is a minimal nonsupersolvable group of order 400, and it has the desired properties. The number of this group in the library of SmallGroups [5] is

[400,129]. All subgroups of the group G except for the noncyclic maximal subgroup $\langle a\rangle\langle b\rangle$ of order 16 are $\mathbb{P}$-subnormal in G.

Example 2. In general, a group with $\mathbb{P}$-subnormal p-subgroups and $\mathbb{P}$-subnormal Schmidt subgroups may be nonsupersolvable. For example, the group

$$G = [E_{7^2}]([\langle a\rangle]\langle b\rangle),\ |a| = 3^2,\ |b| = 2.$$

has the desired properties. It is minimal nonsupersolvable group of order $882 = 2 \cdot 3^2 \cdot 7^2$. The number of this group in the library of SmallGroups [5] is [882,17]. All subgroups of the group G except for the maximal subgroup $[\langle a\rangle]\langle b\rangle$ are $\mathbb{P}$-subnormal. The subgroup $[\langle a\rangle]\langle b\rangle$ of order $3^2 \cdot 2$ is a biprimary noncyclic z-group, but it is not a Schmidt group.

References

[1] B. Huppert. Endliche Gruppen I. Berlin, Heidelberg, New York. 1967.

[2] A.F. Vasilyev, T.I. Vasilyeva, V.N. Tyutyanov, On the finite groups of supersoluble type // Siberian Mathem. J. 2010. Vol. 51, No. 6. P. 1004–1012.

[3] V.N. Kniahina, V.S. Monakhov, Finite groups with $\mathbb{P}$-subnormal 2-maximal subgroups // ArXiv. org e-Print archive, arXiv:1105.3663, 18 May 2011.

[4] V.N. Kniahina, V.S. Monakhov, Finite groups with $\mathbb{P}$-subnormal primary cyclic subgroups // ArXiv. org e-Print archive, arXiv:1110.4720V2, 18 Nov 2011.

[5] GAP (2009) Groups, Algorithms, and Programming, Version 4.4.12.*www.gap-system.org*.

Some identities of Ramanujan type

Yu. V. Nesterenko

Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Lenin Hills, 1, 119899, Moscow, Russia (E-mail: nester@mi.ras.ru)

The Eisenshten series $E_{2k}(\tau), k \geq 1$, can be defined as

$$E_{2k}(\tau) = 1 + \frac{4k}{B_{2k}} \sum_{n=1}^{\infty} \sigma_{2k-1}(n) q^n, \qquad q = e^{2\pi i \tau}, \quad \tau \in \mathbb{C}, \, \Im\tau > 0,$$

where B_{2k} are Bernulli numbers and $\sigma_m(n) = \sum_{d|n} d^m$. It is well known that the functions $e^{\pi i \tau}, E_2(\tau), E_4(\tau), E_6(\tau)$ are algebraically independent over $\mathbb{C}$ but for any imaginary quadratic ξ, $\Im\xi > 0$, the field generated over $\mathbb{Q}$ by three numbers $E_2(\xi), E_4(\xi), E_6(\xi)$ has transcendence degree 2. This degeneracy allowed to prove algebraic independence of the numbers π, e^{π} and $\Gamma(1/4)$ (with $\xi = i$).

We plan to discuss properties of functions

$$g_{u,v}(\tau) = \sum_{n=1}^{\infty} n^u \sigma_{-v}(n) q^n, \qquad 0 \leq u < v.$$

that are algebraically independent together with $e^{\pi i \tau}, E_2(\tau), E_4(\tau), E_6(\tau)$ over $\mathbb{C}$. Some identities of modular type are proved for these functions. As a consequence we prove that the transcendence degree of the field generated by the values of $g_{u,v}(\xi)$ is smaller that it is expected. The interest to these functions is motivated by equalities of the type

$$\zeta(3) = \frac{7\pi^3}{180} - 2 \sum_{n=1}^{\infty} \sigma_{-3}(n) e^{-2\pi n}.$$

Similar formulas hold for other odd Riemann zeta-values. For the first time they were announced in 1901 by M. Lerch. They can be deduced from some functional identity announced in 1916 by S. Ramanujan and have been proved by E. Grosswald in 1970.

Automated Objects Cooperation Behavior Model

F. Novikov

I.I. St. Petersburg State Polytechnical University, Department of applied mathematics Polytechnicheskaya, 29, 195251, St. Petersburg, Russia (E-mail: fedornovikov@rambler.ru)

Developers of software and hardware systems do have ultimate need to prescribe the behavior of components and agents involved. Exact and exhaustive description of actions and events sequences is essential in many application areas: in software development, in business modeling, in hardware protocols construction. Description of actions and events sequence is generally referred to as *behavior description* or *behavior model* or, for short, *behavior*.

There are a lot of behavior formal models, and the most popular of them are unified within Unified Modeling Language (UML) [1]. Conrad Bock, UML apologist, put forward the discussion on comparison of behavior models in his earlier paper [2]. These models were control flow models, data flow models and state machine models. This classification seems to be not quite complete, and not orthogonal to the current state of the art. We prefer to distinguish state transitions model (state machine diagrams), control and object flow model (activity diagrams) and messaging model (sequence and other interaction diagrams). In any case, having behavior described with some formal diagrams, one could find out discrepancies in software models, apply formal verification techniques and gain other benefits. The problem is these models suits well in their particular domain, namely object-oriented software systems design, and reveal lack of generalization at the ontological level [3]. Thus we introduce state machine diagram extension, easy to use and at the same time general enough to cover wide range of practical applications.

Next section contains brief description of behavior model proposed, then we argue the model to be general enough due its Turing completeness,

and at last we analyze well known example of behavior description.

Automated Objects Cooperation

First of all, we are going to describe the behavior of several communication objects or actors, rather than behavior of one stand alone object. Thus one of the most important features we want to achieve in the introduced behavior model is modularity. Activity diagrams have no means of interaction with each other, thus we gave them up. Sequence diagram, especially in the form of communication diagram, do have such means, but these diagrams are of extremely low level obviously, far below the level of ontology we need, so we gave them up too. State machine diagram is the only suitable choice, but it needs some extensions. Therefore we extend the notion of a finite automaton specified by the UML state machine with features serving for interaction with other objects and/or with an external environment. We assume communicating object to be the first class real world object capable to do whatever it have to do. That is why there are no restrictions on the acceptable actions in the behavior model described. It distracts us from limited model of finite automata and leads us to a powerful model of automated objects cooperation.

Figure 1 shows basics concepts of the behavior model. Part (a) shows four principal interfaces for automated object interaction. Part (b) shows the model of an automated object interacting with other objects. Let us consider the entire model in detail. An automated object mod-

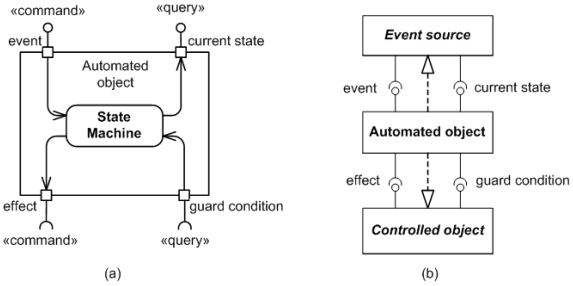


Figure1. Automated Objects Cooperation: (a) objects interfaces and (b) interaction of objects.

els a behavior. This behavior maps a sequence of events coming from

an Event *source* object to a sequence of effects which may yield event performed by a *Controlled object*. The mapping is modeled through two interactions, as shown in Figure 1: one between *Automated object* and *Controlled object* and another one between *Automated object* and *Event source* object.

When the automated object receives an event, it handles it just as an ordinary state machine: it checks the guard condition, changes the current state and executes effects addressed to the controlled object. We model these four concepts as interfaces *event*, *current state*, *effect* and *guard condition*, depicted in Figure 1(a).

An *event* is a way to command the automated object and to transmit input to it. Therefore we model it as the command provided by the automated object: the ball with the "*command*" stereotype in the top left corner of Figure 1(a).

An *effect* is the automated object's output, performed on the controlled object. It is modeled as the command required by the automated object: the socket with the "*command*" stereotype in the bottom left corner of Figure 1(a).

A *guard condition* provides possibility to take into account the state of the controlled object. So we model it as the query required by the automated object: the socket with the "*query*" stereotype in the bottom right corner of Figure 1(a).

A *current state* is an internal data of the automated object, which value might be useful to observe from outside. Therefore it is modeled as the query provided by the automated object - the ball with the "*query*" stereotype in the right top corner of Figure 1(a).

Here we use Bertrand Meyer's command-query separation principle [4], which distinguishes two types of the operations of an interaction interface: a query that returns data and does not change object's state, and a

command that changes object's state. To distinguish these two types of operations we introduce stereotypes "*query*" and "*command*", respectively. To distinguish between provided and required interfaces we use the standard notation of UML component diagrams, namely, balls and sockets attached to the borders of objects. Make note, that we use all four possible combinations: provided commands, required commands, provided queries and required queries. Thus the model is most complete.

There is certain duality of the provided and required commands: an effect on required side turns out to be an event on provided side; similarly, current state on provided side constructs guard condition on required side. It means that *Event source* and *Controlled object* are not singular entities, but rather play specific roles as depicted in Figure 1(b). In fact, every automated object could acts as a source of events or as a controlled object for any other objects or even for itself. This possibility is modeled through the realization relationships between the *Automated object* and the *Event source* and the *Controlled object*, as shown in Figure 1(b). Note that names Event source and Controlled object are given in italics to show these are roles, or abstract objects in terms of UML. This unification of all interacting components, when any automated object could play any role, provides the modularity declared above.

The interaction model described here can be used in two general ways. If *Event source* and *Controlled object* are implemented by the external components, then the interaction model captures the interaction of an *Automated object* with an external environment. On the other hand, if *Event source* and *Controlled object* are implemented by automated objects, then the interaction model describes the way to assemble behaviors in arbitrary desired system.

The last but not the least: the automated object may be at the same time source of events and/or controlled object. Moreover, automated object might be connected with arbitrary number of sources of events and controlled objects simultaneously. The model is extremely flexible, there are no restrictions at all: everything is permissible, provided con-

formity of interfaces (respect of signed contracts). That is why we call the model *Automated Objects Cooperation* (AOC).

Turing Completeness of the Behavior Model

Having in mind the problem of behavior description the question arises: whether formal model of behavior rich enough to capture all practical needs? In the case of AOC the answer is positive: the model is Turing complete and thus sufficient to express any definite behavior. The proof is given in Figure 2 in a form of emulator of Turing machine, which behavior is captured by three cooperative automated objects: control unit, program memory (table of state transitions) and data memory (tape). The Figure 2 gives an overview of notation used as well. Control unit

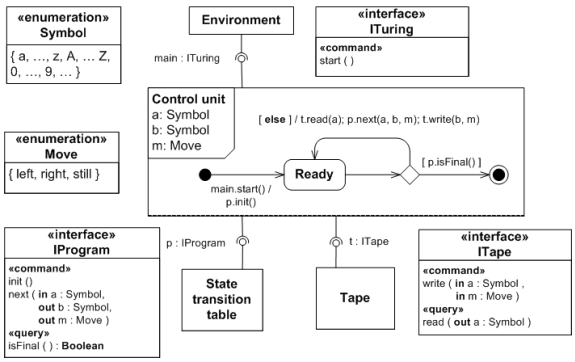


Figure2. Turing machine as Automated Objects Cooperation.

provides only one main command: it could be started from outside and that is all. Being started Turing machine puts program memory into initial state, gets ready and immediately (spontaneously, as it used to refer in UML) reads symbol from the tape, moves along the tape if needed, writes symbol to the tape and so on. Control unit uses local variables, denoted *a*, *b*, and *m* of enumerated types *Symbol* and *Move*. What are inside "black boxes" *Environment*, *State transition table* and *Tape* doesn't matter. There might be AOCs of any kind inside them, or there might be some electronic devices, or human beings, or... That will do while declared interfaces are provided and the signed contracts are respected.

We see that behavior of Turing machine as it used to be described in manuals has been captured pretty well. But is this diagram of any practical interest? Is it useful in analysis, comparison, and harmonization of behavior descriptions? We believe it is. Even superficial analysis of the diagram in Figure 2 shows, that common description of Turing machine behavior is essentially incomplete. For instance, we need the ability to populate the tape with arguments before starting the machine, and we need the ability to review results, after Turing machine reached a final state. Similarly, we need the ability to fill the state transition table cells before starting the machine, and we might badly need the ability to watch its current state if we are going to debug Turing machine. Thus we arrive at the necessity to construct more complicated AOC somewhat like depicted at Figure 3. That's not all. An ideal Turing machine

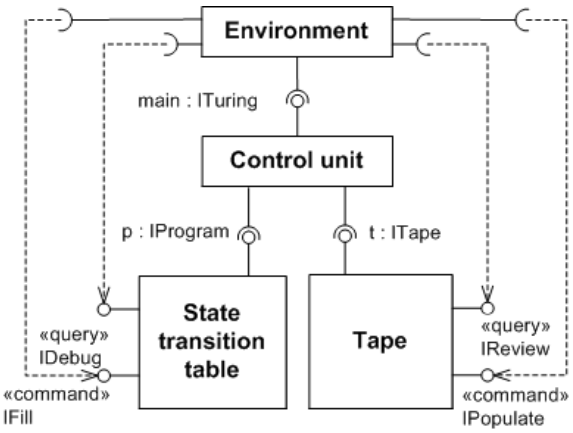


Figure3. Enhanced Automated Objects Cooperation for Turing machine.

never crashes, but real world programs do being broken sometimes, as we know. Their behavior under exceptional circumstances is crucial for the applications. The AOC technique allows to model concurrent behavior with exceptions uniformly, as we'll see in the next section.

Writer and Reader Example

Most processes in real world are concurrent, thus description of concurrent behavior is in the focus of standards harmonization. Let us re-

visit common example of communicating Writer and Reader processes. Writer sequentially writes some records from time to time and reader reads them in the same sequence if any. Both are independent and proceed on their own accord, but they are signed the contract. The Reader must read a record if it have placed an order. Of course, the Reader couldn't read a record if it hadn't been wrote yet, and the Writer shouldn't write new record if the previous one hadn't been read yet. How they should communicate safely? Since 1965, when Dijkstra firstly introduce the problem, there were put forward hundreds solutions, more or less fortunate. The AOC solution does not claim to scientific novelty, it is just simple (not to say trivial) and obvious, and that is the point (Figure 4). The only assumption is that handling any event is an atomic operation. Of course, the solution in Figure 4 is not complete. For in-

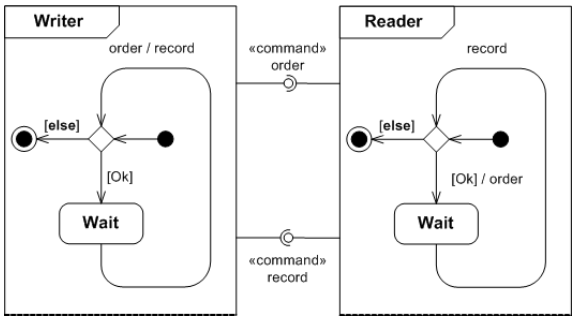


Figure4. The Writer and the Reader cooperation.

stance, it does not treat indefinitely waiting in the case of the other partner is rest in peace. The AOC technique allows refining the model in straightforward manner, reusing most of design (Figure 5). To sum it up, automated objects cooperation we introduced seems to be quite adequate for behavior descriptions of various kinds.

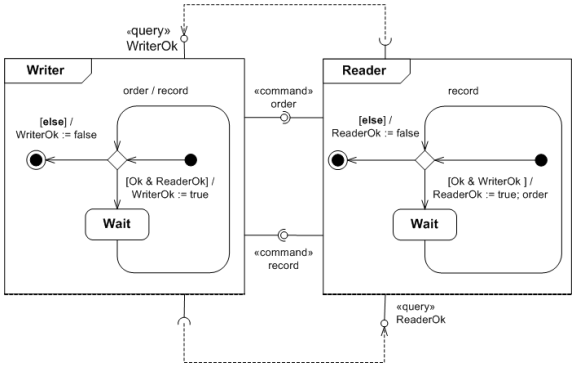


Figure5. The Writer and the Reader cooperation avoiding infinite waiting.

References

[1] Novikov F., Ivanov D. Modeling with UML (in Russian), Nauka i Tekhnika, 2010.

[2] Bock C. Three Kinds of Behavior Model. Journal Of Object-Oriented Programming, Vol 12, No 4, July/August 1999.

[3] Bock C. Unified Behavior Models. Journal Of Object-Oriented Programming, Vol 12, No 5, September 1999.

[4] Meyer B. Object-oriented Software Construction. Hemel Hempstead: Prentice Hall, 1988.

Application of a hybrid approach to forecasting time series

V. G. Pienko[1] and O. A. Penko[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: `valerpenko@mail.ru`)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: `helen.odes@gmail.com`)

The task of forecasting is a very common problem, which has a wide scope and great variety of formal methods for the solution. There is a great practical importance of forecasting techniques in relation to the

economic and managerial problems of various sizes. However, a collision with reality reveals the potential problems of classical methods. Main reason of such kind of problem is a quality of a source data - often it is either incomplete or distorted sufficiently. In such a situation reasonably natural is an application of complex methods, which use artificial intelligence.

In this paper we attempt to develop an approach to the problem of predicting the behaviour of time series, although the main results can be used for solving multivariate prediction. The proposed approach in this paper is to apply the following procedures:

1. Using a sliding window method to convert the time series in a training set for the neural network.
2. Construction of a feed forward neural network, using this training set.
3. Training a neural network using genetic algorithm with real-coded attributes.

Usage as a primary mechanism of artificial neural networks caused by it's high level adaptation properties, that allows us to hope to overcome the difficulties associated with defects in the source data.

Sliding window method is a natural procedure that allows to prepare elements of a training set based on time series. In this case one part of the time series will be used to build a proper training set, and another part will be used as a test in order to avoid the network over fitting and determine the correct moment of training completion.

Using a genetic algorithm for neural network training motivated by the desire to reduce the danger of a neural network to achieve a local minimum of its error. The structure of neural network created by the proposed method can have a large number of settings, first of all weights of synaptic connections. In the traditional binary coding features in the genetic algorithm and demands for high accuracy of data it can lead to huge cost of memory for coding attributes, and as a consequence, a significant increase in computational complexity of the genetic algorithm. These considerations led to the use of variants of the genetic algorithm with real-coded attributes. In this case you will need the use of spe-

cialized genetic operators described in the literature. Approbation of such an approach has theoretical and practical component, because it is known controversial statements about its effectiveness.

The above approach has been implemented as a software system, that has been tested on a relatively small sample of the source data. The results showed a high rate of convergence of the training process and received high level of generalized prediction on the basis of the criterion of Mean Squared Error (MSE).

During the implementation of this approach should be to decide on a fairly large number of parameters - the width of the sliding window, proportion split the data into training and test part, the parameters determining the structure of the neural network, parameters used in the genetic operations. At this stage, the values of these parameters are chosen heuristically based on the experience of the researcher. So it is promising a following step in developing a procedure for automatic selection of the values of these parameters. As the cycle of training the neural network was not the odd time-consuming in terms of computation, it is possible to automate the selection of these parameters by applying the same genetic algorithm. In this case, species of the population will be one of the possible configurations of the experimental parameters and fitness function - the value of MSE.

References

[1] John E. Hanke and Arthur G. Reitsch Business Forecasting, 7th edition.: Trans. from English. - M.: *Publishing House, "Williams"*, 2003. - 656 p. Ill.

[2] E.E. Tikhonov. Forecasting methods in market conditions: a textbook. - Nevinnomyssk, 2006. - 221 p.

The signs of periodic and aperiodic tiles

O. A. Petrov[1] and T. I. Petrushina[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: oleg.petrov.od@yandex.ru)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: tatyana.petrushina@gmail.com)

In this paper we consider the problem of covering the plane of irregular patterns and explore some signs of periodicity and aperiodicity of the mosaics.

The problem of tiling the plane of the irregular pattern is as follows: it is required for a given finite set of tiles of the same shape and the rules of their local neighborhood to construct a covering of the plane of these plates, so that does not violate the rules of the neighborhood, nor lead to voids and overlaps, as well as that for any infinite subset tiles tiling any transformation of the form

$$\{x \to x + R * x + n * \overleftarrow{e}\}$$

is not translated this subset of itself.

Mosaic is called aperiodic if the tiles were included in it, you can build only aperiodic tilings. [1]

This problem has occupied many mathematicians like Penrose [2], Goodman-Strauss, Taylor, Sokolar [1]. They built a series of mosaics and proven irregularity constructed examples. Taylor and Sokolar solved the einstein problem construction of an irregular mosaic, consisting of only one plate.

A very important task, which is still not solved, is the problem of classifying the mosaics on the regularity or irregularity. It can not be solved directly because it is impossible to build an infinite tiling. Therefore it is necessary to provide indirect evidence on which to classify the mosaic. We will use such features as the properties of tilings of certain limited areas.

Let we count the number of ways to tile a limited area. Suppose we have a long, narrow plot size, and we have consistently tile it left

to right. Then call the profile [3] all the information that is necessary for subsequent tiling. Because the rules of the neighborhood are local in nature, the profiles will have a width of 2 columns. (See example Figure 1)

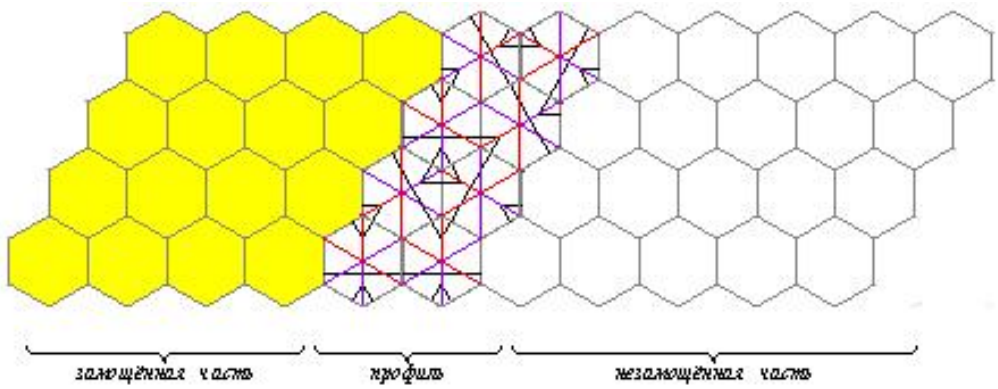


Figure1. Profile of tiling

Now with the help of profile dynamic programming we can calculate the number of tilings of our site. [3] Computational complexity is (1).

$$T(w,h) = O(w\Delta P^2) \tag{1}$$

where P is number of allowed profiles.

We denote the number of ways to tile the area h w as N(h,w). Then, for irregular mosaics function (2) will have a period of 4, starting with some w.

$$R(w,h) = \frac{N(w,h+1)}{N(w,h)} \tag{2}$$

At the same time, for regular mosaics from some w, the sequence is periodic with period 1, or is not periodic. Thus, investigating the local properties of the mosaic, it can be concluded about its global properties.

We formulate two definitions and sufficient condition for the regularity of the mosaic.

Definition 1. Prefix of a column is called a continuous subsequence of plates of the column, starting with his first plate.

Definition 2. Suffix is called a continuous column of subsequence tiles that column, ending his last plate.

Theorem 1. *Suppose that for some there is a mosaic of allowable height profile, such that it can be to go to him also for his first column prefix length 2 coincides with the suffix of length 2. Then the mosaic there is a regular tiling.*

References

[1] Joshua E. S. Socolar, Joan M. Taylor, An aperiodic hexagonal tile: 21 p. url: http://arxiv.org/abs/1003.4279.
[2] R. Penrose, The Rôle of Aesthetics in Pure and Applied Mathematical Research: Bull. Inst. Math. Appl., 10:266-271, 1974.
[3] Dvorkin M. E., Dynamic programming over profiles: Materials Winter School Programming, Kharkov, KNURE, 2011 pp. 56-61.

On making the list of (0,1) exponent matrices

M. V. Plakhotnyk

Kyiv National Taras Shevchenko University, str. Volodymyrs'ka 54, 01601 Kyiv, Ukraine (E-mail: `makar_plakhonyk@ukr.net`)

In our talk we will introduce and compare two different algorithms of making a list and whence of calculating the number of (0,1) exponent matrices of given order.

Exponent matrices are widely used in algebra (in ring theory) as powerful tool of description of tiled orders. Also there in a connection between reduced exponent matrices and finite partially ordered sets.

An integer matrix $A = (\alpha_{pq}) \in M_n(\mathbb{Z})$ of order n with zeros on diagonal is called exponent matrix if for all possible indices i, j, k inequalities

$\alpha_{ij} + \alpha_{jk} \geqslant \alpha_{ik}$ take place.

An exponent matrix is called reduced exponent matrix if it has no symmetrical zeros.

There is one to one correspondence between (0,1) reduced exponent matrices (i.e. exponent matrices, all which elements are equal to either 0 or 1) of order n and partially ordered sets with n elements. It is the following. Let $A = (\alpha_{pq})$ be a (0; 1)-reduced exponent matrix of order n. If $\alpha_{ij} = 0$ then let it be the relation $t_i < t_j$ in the partially ordered set $\{t_1, \ldots, t_n\}$.

In this thesis we compare two ways of making a list of (0,1) exponent matrices.

In [1] it was introduced the notion of super minimal (0,1) exponent matrix and these matrices were described. (0,1) exponent matrix is such (0,1) exponent matrix which is not greater (in non strict sense) then any other (0,1) exponent matrix. It is also proved in [1] that the number of super minimal (0,1) exponent matrices or order n is equal to $2^n - 2$.

All super minimal (0,1) exponent matrices at first are ones of the form

$$A_{n,k} = \left(\begin{array}{c|c} \mathcal{O}_k & U \\ \hline \mathcal{O} & \mathcal{O}_{n-k} \end{array}\right)$$

such that $\mathcal{O}_k$ together with $\mathcal{O}_{n-k}$ are square matrices which and consist with zeros and have orders k and $n - k$ correspondingly, matrix $\mathcal{O}$ consist with zeros and matrix U consist with ones, where $1 \leqslant k < n$ and secondly super minimal (0,1) exponent matrices are those which can be obtained from $A_{n,k}$ with simultaneous permutations of lines and columns which have the same numbers.

In [2, Theor. 5] it is shown that the maximal length of ordered chain of (0,1) exponent matrices of order n is equal to $\frac{n(n+1)}{2}$ and also in [2] it is introduced the notion of union of two exponent matrices (it is a matrix, whose element is equal to maximum of correspond elements of former matrices) which stays the property of matrix to be an exponent one.

Using these facts, making a list of reduced (0,1) exponent matrices of order n can be realized as union of not more then $\frac{n(n+1)}{2}$ matrices of

the list of $2^n - 2$. Such operation needs consideration of $\sum_{k=1}^{n(n+1)/2} \binom{2^n-2}{k}$ matrices.

Another way of making a list of (0,1) exponent matrices is to use direct step by step consideration in the following way. We can use the direct linear ordering on the set of square matrices of order n as on the set of vectors of order n^2 where the lowest $(0,1)$ exponent matrix is zero matrix and the greatest is that one whose all non diagonal elements are equal to 1.

If for some matrix $A = (\alpha_{pq})$ the inequality $\alpha_{ij} + \alpha_{jk} \geqslant \alpha_{ik}$ is violated then lets consider not directly next matrix but that one for which this inequality works. We do not have the evident formula for number of necessary consideration of matrices but have made some numerical calculations for small orders which let to compare two methods and results are introduced in the table below.

Order of matrices	Number of steps with using minimal matrices	Number of steps with using direct calculation	Number of (0,1) exponent matrices
3	50	63	28
4	1026	15.913	354
5	31.395	614.429.671	6.941
6	1.431.593	34.896.814.868.837.200	209.526

Table1. Results of the calculation

Nevertheless there were only matrices of small orders under consideration, the experimentally found numbers of matrices which are necessarily have to be considered, obviously shows that direct calculation in the problem of making the list of all (0,1) exponent matrices is much more quicker way then with using of super minimal matrices and their unions.

References

[1] Kirichenko V.V., Plakhotnyk M.V., "Superminimal exponent matrices", Bull. of Kyiv Taras Shevchenko National University, Ser. Phys.& Mech., N 2, 2011, p. 20-22.

[2] Kirichenko V.V., Plakhotnyk M.V., "The description of (0,1) - exponent matrices", Bull. of Kyiv Taras Shevchenko National University, Ser. Phys.& Mech., N 3, 2011, p. 34-36.

Об унитарных делителях целых гауссовых чисел в секторах

П. В. Попович

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: polina_555@rambler.ru)

Пусть $\mathbb{Z}[i]$ - кольцо целых гауссовых чисел, $\alpha \in \mathbb{Z}[i]$. Число $\delta \in \mathbb{Z}[i]$ называется унитарным делителем α ,если $\delta|\alpha$ и $(\delta, \alpha) = 1$.

Пусть $k \in \mathbb{N}$, $k \geq 2$. Представление $\alpha = \delta_1 \cdots \delta_k$, $\delta_j \in \mathbb{Z}[i]$ назовем k-унитарным, если $(\delta_i, \delta_j) = 1$ для всех $i, j = 1, \ldots, k$, при чем $i \neq j$.

Обозначим через $\tau_k^*(\alpha)$ - количество различных k-унитарных представлений α, $\alpha \in \mathbb{Z}[i]$. В силу мультипликативности функции $\tau_k^*(\alpha)$ имеем $\tau_k^*(p^m) = k$.

Для $k \geq 2$ и $\Re(s) > 1$ справедливо равенство

$$\sum_{0 \neq \alpha \in \mathbb{Z}[i]} \frac{\tau_k^*(\alpha)}{N(\alpha)^s} = \prod_p \left(1 + \frac{k}{N(p)^s} + \frac{k}{N(p)^{2s}} + \ldots\right) = \frac{Z^k(s)}{(Z(2s))^{\frac{k(k-1)}{2}}} G_k(s),$$

где $Z(s)$ - Z-функция Гекке поля гауссовых чисел, а $G_k(s)$ есть регулярная функция в области $\Re(s) > \frac{1}{3}$, причем при $k = 2$ функция $G_k(s)$ тождественно равна 1.

Мы строим асимптотическую формулу для сумматорных функций $T_k(x) = \sum\limits_{N(\alpha) \leq x} \tau_k^*(\alpha)$, $k = 2, 3, \ldots$.

Для построения асимптотической формулы воспользуемся фор-

мулой Перрона

$$T_k(x) = \frac{1}{2\pi i}\int\limits_{c-iT}^{c+iT} \frac{Z^k(s)}{(Z(2s))^{\frac{k(k-1)}{2}}} G_k(s)\frac{x^s}{s}ds + O\left(\frac{x^c}{T(c-1)^k}\right),$$

причем $Z(2s) \neq 0$ в области $\Re(s) \geq 1 - e^{\frac{-c\cdot(\log|t|+10)^{-3/5}}{(\log\log|t|+10)^{1/5}}}$.

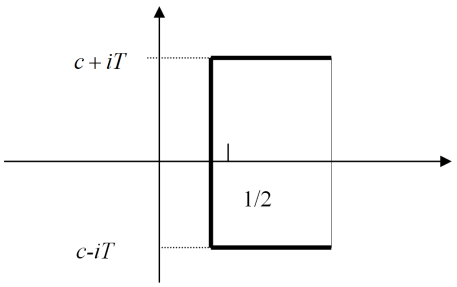
Передвигая контур интегрирования на прямую

$$\Re(s) = 1 - e^{\frac{-c\cdot(\log|t|+10)^{-3/5}}{(\log\log|t|+10)^{1/5}}}$$

мы проходим через полюс функции $Z(s)$ в точке $s = 1$.

Учтем что $\operatorname{res}\limits_{s=1}\left(\frac{Z^k(s)}{(Z(2s))^{\frac{k(k-1)}{2}}}\cdot\frac{x^s}{s}\right) = x\cdot P_{k-1}(\log x)$, где $P_k(u)$ - многочлен степени k от $\log x$.

А затем вычисляем интегралы на отрезках контура:



На горизонтальных участках используем оценку $Z(s)$ в критической полосе

$$Z(s) \ll \begin{cases} |t|^{2\cdot\frac{1-\sigma}{3}}\log|t|, \text{если } \frac{1}{2} \leq \sigma \leq 1 \\ |t|^{\left(\frac{1}{6}+\frac{1}{2}-\sigma\right)\cdot 2}, \quad \text{если } 0 < \sigma \leq \frac{1}{2} \end{cases} \text{ и}$$

$$\frac{1}{Z(2s)} \ll (\log(|t|+10)) \quad \Re(s) \geq \frac{1}{2} - e^{-\frac{c}{2}\cdot\frac{(\log|t|+10)^{-3/5}}{(\log\log|t|+10)^{1/5}}}$$

Кроме того, мы изучаем функцию $\tau_k^*(\alpha)$ в узких секторах

$$\alpha \in s(x;\varphi_1,\varphi_2) = \left\{\omega \in \mathbb{Z}[i] \middle| N(\omega) \leq x,\ 0 \leq \varphi_1 \leq \arg\omega \leq \varphi_2 \leq \frac{\pi}{2}\right\}.$$

В частности доказаны теоремы:

Theorem 1. *При $x \to \infty$ справедливы асимптотические формулы*

$$\sum_{N(\alpha)\le x} \tau_k^*(\alpha) = \begin{cases} A_1 x\log x + A_0 x + O\left(x^{\frac{1}{2}} e^{\frac{-c(\log x)^{3/5}}{(\log\log x)^{-1/5}}}\right), & k=2 \\ xP_k(\log x) + O\left(x^{\frac{5}{8}+\varepsilon}\right), & k=3,4 \\ xP_k(\log x) + O\left(x^{\alpha_k+\varepsilon}\right), & k\ge 5,\ \alpha_k = \frac{56k-165}{56k}, \end{cases}$$

где $P_k(n)$ - многочлен степени $(k-1)$ с коэффициентами, зависящими только от k.

Theorem 2. *Пусть $0 \le \varphi_1 < \varphi_2 \le \frac{\pi}{2}$, тогда при $x \to \infty$ имеем*

$$\sum_{N(\alpha)\le x} \tau_2^*(\alpha) = \frac{\pi}{2}(\varphi_2-\varphi_1)x(\log x + B_0) + O\left(x^{\frac{1}{2}}\log^{\frac{4}{3}} x\right).$$

Рассмотрим функцию $F(s,\chi) = \sum\limits_{\chi(q)} \sum\limits_{\alpha} \frac{\tau_k^*(\alpha)}{N(\alpha)^s}\chi(N(\alpha))$.

Откуда следует,что $\sum\limits_{\chi(q)} \chi_q(a)F(s,\chi) = \sum\limits_{N(\alpha)\equiv a(q)} \frac{\tau_k^*(\alpha)}{N(\alpha)^s}$, где χ_q - характер в $\mathbb{Z}$ по модулю q.

Таким образом, доказана асимптотическая формула

$$\sum_{\substack{N(\alpha)\equiv a(q)\\ N(\alpha)\le x}} \tau_2^*(\alpha) = A_1(q)\frac{x}{q}\log x + A_0(q)\frac{x}{q} + O\left(\frac{x^{\frac{1}{2}}+\varepsilon}{q^{\frac{7}{8}}}\right).$$

References

[1] De Koninck J.M. and Ivic A., Topics on Arithmetical functions // North Holland, Amsterdam, 1980.

[2] Ivic A., The Riemann zeta function, John Wiley and sons // New York, 1985.

[3] Titchmarsh E.C., The theory of Riemann zeta function // Foreigin Litricute Pres., Moscow, 1953.

The function $\tau_3(w)$ in arithmetic progressions

A. S. Radova

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: radova_as@mail.ru)

Let $\tau_3(w)$ denote the number of representations $w = \alpha_1\alpha_2\alpha_3$, where $\alpha_i \in Z[i]$.
We define

$$T(x;\alpha_0,\gamma): = \sum_{\substack{w\in Z[i]\\ w\equiv w_0(\gamma)\\ N(w)\le x}} \tau_3(w) \tag{1}$$

where $N(\gamma) \le X$. We study the function $\tau_3(w)$ in arithmetic progression $w \equiv \alpha_0(mod\gamma)$, provided that $N(\gamma)$ increases with x.
D. R. Heath-Brown obtained [1] the asymptotic formula

$$\sum_{\substack{n\equiv a(q)\\ n\le x}} \tau_3(n) = \frac{x}{\varphi(q)} res_{s=1}(L(s,\chi_0)^3\frac{x^{s-1}}{s}) + O(\frac{x^{\frac{86}{107}+\varepsilon}}{q^{\frac{66}{107}}}) \tag{2}$$

where $x < X^{\frac{1}{2}+\frac{1}{82}}$, $q < x^{\frac{1}{2}}$.
We use the Hecke-function in the critical strip and evaluation of the second moment on the critical line.
Then we have the following

Theorem 1. *Let $\alpha_0, \gamma \in Z[i]$, $(\alpha_0,\gamma) = 1$, then*

$$T(x;\alpha_0,\gamma): = \sum_{\substack{w\in Z[i]\\ w\equiv w_0(\gamma)\\ N(w)\le x}} \tau_3(w) = \frac{x}{N(\gamma)} P_2(logx) + O(x^{5/8+\varepsilon}N(\gamma)^{-3/8}) \tag{3}$$

where $P_2(u) = a_2u^2 + a_1u + a_0$, a_i are computable constants.

References

[1] D. R. Heath-Brown, The divisor function $d_3(n)$ in arithmetic progressions - Acta Arithmetica, XLVII (1986), №1, p. 29-55.

Using the criterion approach for analysis of the expediency of implementation of the cloud services of AMR system

Olga I. Roznovets[1] and Ludmila A. Voloschuk[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: olga_roznovets@mail.ru)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: lavstumbre@gmail.com)

In the context of the global economic crisis many enterprises and institutions faced an acute problem of saving money and resources, including electricity. One of the ways to solve this problem is an accurate, reliable and just-in-time control and accounting of electricity carried out by automated meter reading (AMR) systems. Of particular interest is the implementation of AMR system using the concept of the cloud computing which allows to reduce the complexity and cost of IT-infrastructure, that is especially important for small and medium businesses.

Currently, cloud computing [1] is considered as the most promising concept of organization of a common space of distributed information and analytical services for companies and private users. This approach provides significant benefits for consumers of cloud services (such as reducing the cost of maintenance of hardware and software infrastructure, payment for services rendered by a well-defined rates) and providers (economies of scale).

Obviously, the expediency of implementation of the separate business processes as independent software modules in order to place them into

the cloud must be justified taking into account many factors. Known information systems design methodologies (UML, SADT, IDEF, DATARUN, ARIS) do not provide a means of assessing of expediency of making a functional business processes as the separate services. So obvious the topicality of creation the criterion approach and decision-making mechanisms that enable at the design phase of the information system to identify the separate subsystems which can be reused in other applications as web-services or can be available to users as a cloud services, according to the Software as a Service (SaaS) model.

Expediency of implementation of the separate subsystems of enterprise applications as a cloud services requires an assessment of a number of criteria. Thus, the article "Assess enterprise applications for cloud migration" [2] is produced, using the analytic hierarchy process [3], quantitative and qualitative assessment of the general criteria in terms of value to the business, technical capabilities and the degree of risk. At the same time, taking into account the specifics of a particular domain is very important and actual.

The report examines the results of using the analytic hierarchy process in determining the expediency of creation a public cloud service that implements the tasks of the AMR system. Assesses the additional criteria, such as the parameters of the stored data, the using intensity of service, scalability, requirements for computing resources, as well as the requirements of specific clients, their technological and financial capabilities.

References

[1] Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology / NIST, 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] Brijesh Deb. Assess enterprise applications for cloud migration / IBM Corporation, 2010. http://public.dhe.ibm.com/software/dw/cloud/library/cl-assessport-pdf.pdf

[3] Thomas L. Saaty. The analytic hierarchy process: planning, priority setting, resource allocation / McGraw-Hill International Book Co., pp. 287, 1980

Inversive congruential generator of pseudorandom numbers

V. Rudetskyi

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: v.rudetsky@yahoo.com)

In the majority of problems of modern cryptography use of sequences of random numbers is required. However in practice instead of random sequences of numbers the sequences of pseudorandom numbers that pass statistical tests on a chance are used.

Let $p \geq 3$ be a prime, $n > 2$ be an integer, $a, b, c, y_0 \in \mathbb{Z}_{p^n}$, $(c, p) = 1$, $a \equiv b \equiv 0 \pmod{p}$. The sequence $\{y_k\}$, $k = 0, 1, 2, \ldots$, generated by recursion

$$y_{k+1} = \frac{a}{y_k} + bk + c \pmod{p^n}, \tag{1}$$

creates the sequence $\{x_k\}$, $k = 0, 1, 2, \ldots$, $x_k = \frac{y_k}{p^n}$ called the sequence of pseudorandom numbers generated by inversive congruential generator with a variable shift $bk + c$.

Earlier such sequences studied by Eichenauer and Lehn[1], Niederreiter[2], where shift was fixed and equal to b. Then, in the subsequent years, in the work[3] it called in a question of unpredictability of the sequence $\{x_k\}$ generated by congruential generator with a variable shift. We will investigate the generator (1) with variable shift.

Lemma 1. *For each k, $k = 0, 1, 2, \ldots$ the following representation of the element of sequence $\{y_k\}$:*

$$y_k = A_0 + A_1 y_0^{-1} + A_2 y_0^{-2} + \ldots,$$

where A_i, $i = 0, 1, 2, \ldots$ are the polynomials on k, with A_0 divided by p, A_1 divided by p^2, A_j, $j \geq 2$ divided at least by p^3, holds.

Lemma 2. *For each k, $k = 0, 1, 2, \ldots$ the following representation of the element of sequence $\{y_k\}$:*

$$y_k = B_0 + B_1 k + B_1 k^2 + \ldots,$$

where B_i, $i = 0, 1, 2, \ldots$ are the polynomials on y_0^{-1}, with B_0 divided by p, B_1 divided by p^2, B_j, $j \geq 2$ divided at least by p^3, holds.

Denote $S_N(h, y) := \sum_{k=0}^{N-1} e^{2\pi i \frac{hy_k}{p^n}}$, $N \leq \tau$, where h is an integer, $(h, p^n) = p^\delta$, $\delta < n$, $0 \leq p^\delta < n$, $k = 0, 1, 2, \ldots$.

Using estimates of the exponential sums $S_N(h, y)$ with the representations form Lemma 1 and Lemma 2, we obtain the estimates of discrepancy for $s = 1, 2, 3, 4$.

Theorem 1. *Let $p \geq 3$ is a prime, $n > 2$ is an integer, $a, b, c, y_0 \in \mathbb{Z}_{p^n}$, $(c, p) = 1$, $a \equiv b \equiv 0 \pmod{p}$. Then*

$$D_N^{(s)}(y_0) \ll p^{\frac{n}{2}} N^{-1+\varepsilon}$$

are uniformly distributed over $N \in \left[p^{\frac{n}{2}+2\varepsilon}, p^{n-1}\right]$.

References

[1] Eichenauer J. and Lehn J., A non-linear congruential pseudorandom number generator, Statist. Hefte, 27 (1986), 315-326

[2] Niederreiter H., Finite fields and their applications,Contributions to GeneralAlgebra 7, Vienna, 1990, Teubner, Stuttgard, 1991

[3] S.R. Blackburn, D. Gomez-Peres, I. Gutierrez and I. Shparlinski. Predicting nonlinear pseudorandom number generators. Math. Comp., 74: 1471–1494, 2004.

The non-symmetric divisor function in narrow sectors

O. V. Savastru

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: sav_olga@bk.ru)

Let $\mathbb{Z}[i]$ be the ring of Gaussian integers. For $x \in \mathbb{R},\ x > 1,\ \alpha_0, \gamma \in \mathbb{Z}[i]$,
$0 \leqslant \varphi_1 < \varphi_2 \leqslant \frac{\pi}{2}$ consider the summatory function given by

$$T(x, \gamma, \alpha_0, \varphi_1, \varphi_2) := \sum_{\substack{\alpha \equiv \alpha_0 \pmod{\gamma} \\ N(\gamma) \leqslant x \\ \varphi_1 < \arg \alpha \leqslant \varphi_2}} \tau_{1,1,2}(\alpha)$$

where $\tau_{1,1,2}(\alpha)$ denote the number of representations of $\alpha \in \mathbb{Z}[i]$ as $\alpha = \alpha_1 \alpha_2 \alpha_3^2$, where $\alpha_1, \alpha_2, \alpha_3$ are Gaussian integers.

Applying the bound for the Kloosterman sums over $\mathbb{Z}[i]$, the method of Vinogradov we get the asymptotic formula in case, when the norm of a difference of progression grows.

Theorem 1. *Let* $\alpha_0, \gamma \in \mathbb{Z}[i],\ N(\gamma) > 1,\ \alpha_0 \not\equiv 0 \pmod{\gamma}$, $(\alpha_0, \gamma) = \beta$. *Then for every* $\epsilon > 0,\ x \geqslant N^{2+\epsilon}(\gamma)$ *and* $\varphi_2 - \varphi_1 \geqslant \frac{N^{\frac{1}{2}}(\gamma)}{x^{\frac{1}{4}-\epsilon}}$

$$T(x, \gamma, \alpha_0, \varphi_1, \varphi_2) = \frac{\varphi_2 - \varphi_1}{2\pi} \times$$
$$\times \left[c_0(\gamma, \alpha_0) \frac{x}{N(\gamma)} \log \frac{x}{N(\beta)} + c_1(\gamma, \alpha_0) \frac{x}{N(\gamma)} \right] + O\left(\frac{x^{\frac{3}{4}+\epsilon}}{N^{\frac{1}{2}}(\gamma)} \right),$$

where $c_0(\gamma, \alpha_0),\ c_1(\gamma, \alpha_0)$ *are computable functions.*

Analogical problem studied for function $\tau_{1,1,2}(n)$ under the ring of rational integers by Krätzel, Liu.

References

[1] Liu H.Q. Divisor problems of 4 and 3 dimensions [text] / Liu H.Q. // Acta Arith. – 73 (1995). – p. 247-269.

Analogue of Vinogradov's theorem over the ring of Gaussian integers

S. Sergeev

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: sorathss@gmail.com)

The following assertion is proved.

Theorem 1. *Let θ be irrational,*

$$\left|\theta - \frac{a}{q}\right| \leq \frac{1}{q^2}, (a,q) = 1, q \leq x\left(\log x\right)^{-10}$$

Then

$$\tilde{\pi}\left(x,\theta\right) << \left(x^{\frac{2}{3}}q^{\frac{1}{3}} + x^{\frac{5}{6}}q^{-\frac{1}{3}} + x^{\frac{11}{12}}q^{-\frac{1}{6}}\right)\left(\log x\right)^2$$

This result generalizes the result of Dupain, Hall and Tenenbaum on an expansion of the Vinogradov's theorem on the trigonometric sum over the prime numbers.

Corollary 1. *Let θ satisfies Theorem 1. Then*

$$\tilde{\pi}\left(x,\theta\right) = o\left(\tilde{\pi}\left(x\right)\right) := \sum_{N(p)\leq x} 1$$

Theorem 2. *Let $f\left(w\right)$ be a multiplicative function over $\mathbb{Z}\left[i\right]$, $\left|f\left(w\right)\right| \leq 1$. We proved, that for almost all irrationals $\theta \in [0;1]$ we have for $\delta < \frac{k}{\log\log x} < 2-\delta,\ 0<\delta<1$, and $x \to \infty$*

$$\sum_{\substack{w \in \mathbb{Z}\left[i\right] \\ \Omega\left(w\right) = k \\ N\left(w\right) \leq x}} f\left(w\right) e^{2\pi i Re\theta w} = o\left(\tilde{\pi}_k\left(x\right)\right)$$

In the set of multiplicative functions over $\mathbb{Z}$ satisfying $\left[f\left(w\right)\right] \leq 1$ such result have been proved by Indlekofer and Katai.

References

[1] И. М. Виноградов, Метод тригонометрических сумм в теории чисел, Труды МИАН им. В. А. Стеклова, 23 (1947), 1–109.
[2] Y. Dupain, R. R. Hall, G. Tenenbaum, Sur le quirepartition modulo 1 de certaines fonetions de diviseurs

Proving Consistency of Petri Net Grid Models

T. R. Shmeleva

A.S. Popov Odessa National Academy of Telecomunications, str. Kovalska 1, 65023, Odessa, Ukraine (E-mail: tishtri@rambler.ru)

Introduction

For computing grids analysis infinite Petri nets with regular structure were developed and successfully applied [1-3]. But introduced in [1-3] parametric description of infinite Petri nets does not allow the composing systems for calculating t-invariants. The present work is aimed to the developing techniques for calculating t-invariants and proving the consistency of Petri net models.

Dual Parametric Description of Grid

The (direct) parametric description of infinite Petri nets with regular structure developed and successfully applied in [1-3] consists of lines with the following form

$$t_i : pin_{j_k} * apin_{j_k}, ... \rightarrow pout_{j_l} * apout_{j_l}, ...; indices - range, \qquad (1)$$

where t_i is the described transition, pin_{j_k} its input places, $pout_{j_l}$ its output places and $apin_{j_k}, apout_{j_l}$ denotes the multiplicity of corresponding arcs; multiplicity equal to unit is omitted. Thus, for the ordinary Petri net, the following notation is used

$$t_i : pin_{j_k}, ... \rightarrow pout_{j_l}, ...; indices - range. \qquad (2)$$

As it was shown in [1-3], the direct parametric description is very useful for composition of infinite systems of linear algebraic equation for calculating p-invariants of infinite Petri nets with regular structure. The equation of the system constructed on (1) has the form

$$apin_{j_k} * xpin_{j_k} - ... + apout_{j_l} * xpout_{j_l} + ... = 0; indices - range, \quad (3)$$

where $xpin_{j_k}, xpout_{j_l}$ are unknowns corresponding to Petri net places. But the direct parametric description does not help much at calculating t-invariants of infinite Petri nets with regular structure because in the system for calculating t-invariants equations correspond to places and unknowns correspond to transitions. And constructing such a system on the direct parametric description is not a trivial task. That is why other methods were applied in [1] for calculating t-invariants that are grounded on explicit constructing cyclic transitions firing sequences. Let us introduce the dual parametric description of infinite Petri nets with regular structure that consists of lines with the following form

$$p_j : tin_{i_k} * atin_{i_k}, ... \rightarrow tout_{i_l} * atout_{i_l}, ...; indices - range, \quad (4)$$

where p_j is the described place, tin_{i_k} its input transitions, $tout_{i_l}$ its output transitions and $atin_{i_k}, atout_{i_l}$ denotes the multiplicity of corresponding arcs; multiplicity equal to unit is omitted. Thus, for the ordinary Petri net, the following notation is used

$$p_j : tin_{i_k}, ... \rightarrow tout_{i_l}, ..; indices - range. \quad (5)$$

In Section 3, infinite systems of linear algebraic equations are constructed for calculating t-invariants of infinite Petri nets with regular structure. The dual parametric description of an open square grid node has the form:

$$\left(\begin{pmatrix} (\mathrm{po}_u : \mathrm{to}_u \rightarrow) \\ (\mathrm{pol}_u :\rightarrow \mathrm{to}_u) \\ \left(\mathrm{pi}_u :\rightarrow \left(\mathrm{ti}_{u,v}, v = \overline{1,4}, v \neq u\right)\right) \\ \left(\mathrm{pil}_u : \left(\mathrm{ti}_{u,v}, v = \overline{1,4}, v \neq u\right) \rightarrow\right) \\ \left(\mathrm{pb}_u : \left(\mathrm{ti}_{u,v}, v = \overline{1,4}, v \neq u\right) \rightarrow \mathrm{to}_u\right) \end{pmatrix}, u = \overline{1,4} \right). \quad (6)$$

The following indices are used: index u denotes the current port and index v denotes the target port in packets forwarding process. The constant parameter with value 4 denotes the number of square sides; in the

composition of triangular of hexagonal grids [2] values 3 and 6 are used as well.

Infinite Systems of Equations for Calculating T-invariants

Parametric systems for calculating t-invariants of Petri nets are composed easily on their dual parametric description. As an equation corresponds to a place and states the balance of arcs connecting it with its input and output transitions, the equation constructed on (4) has the form

$$atin_{i_k} * ytin_{i_k} - ... + atout_{i_k} * ytout_{i_k} + ... = 0; indices - range, \quad (7)$$

The unknowns are traditionally named with the suffix corresponding to the name of a transition. The system composed on the dual parametric description (6) of a closed grid has the following form:

$$\left(\left(\begin{array}{l} (-\mathrm{yto}_u + \mathrm{yte}_u = 0) \\ (-\mathrm{yte}_u + \mathrm{yto}_u = 0) \\ \left(-\mathrm{yte}_u + \sum\limits_{v=\overline{1,4},v\neq u} \mathrm{yti}_{u,v} = 0 \right) \\ \left(-\sum\limits_{v=\overline{1,4},v\neq u} \mathrm{yti}_{u,v} + \mathrm{yte}_u = 0 \right) \\ \left(-\sum\limits_{v=\overline{1,4},v\neq u} \mathrm{yti}_{u,v} + \mathrm{yto}_u = 0 \right) \end{array} \right) . \right) , u = \overline{1,4} \Bigg) . \quad (8)$$

The system (8) contains finite number of equations. But when composing system on the parametric description of a grid, a system is obtained with the number of equations depending on the value of the parameters. In [1-3] the technique of solving such parametric equations was described for solving systems to find p-invariants which can be applied to solving systems composed on the dual parametric description to find t-invariants of infinite Petri nets with regular structure.

Conclusions

The dual parametric description of infinite Petri nets with regular struc-

ture was introduced and applied to compose infinite systems for calculating t-invariants on an example of a closed square grid model.

References

[1] T.R. Shmeleva, D.A. Zaitsev, I.D. Zaitsev Analysis of Square Communication Grids via Infinite Petri Nets // Transactions of Odessa National Academy of Telecommunication, no. 1, 2009, p. 27-35.

[2] D.A. Zaitsev Verification of Grid Structures with Specific Edge Conditions // 3rd Workshop Program Semantics, Specification and Verification: Theory and Applications, Nizhni Novgorod, Russia, July 1-2, 2012, P. 111-120.

[3] D.A. Zaitsev, T.R. Shmeleva Verification of hypercube communication structures via parametric Petri nets // Cybernetics and Systems Analysis, 2010, Vol. 46, No. 1, P. 105-114.

The comparative characteristic of the intrusion detection systems on basis of neural networks

I. M. Shpinareva

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: ira-shpinareva@rambler.ru)

The efficiency of modern intrusion detection systems depends largely on the applied methods of analysis of network traffic. Generally the following methods are described: signature-based method, expert systems, statistical method and neural networks. Each of these methods has its own advantages and disadvantages. But the most promising direction is IDS on basis of neural networks. The neural network performs analysis of information and provides an opportunity to assess the harmonization of data with the characteristics, which it was trained to recognize. The credibility of assessment depends entirely on the effectiveness of the

learning stage. The advantage of this method is that in addition to the initial period of training, neural networks gained experience with the passage of time, to the extent that, as she conducts analysis of the data obtained from the network traffic. But IDS based on neural network have one drawback - it is a false alarm.

In the study were described IDS built on the basis of multi-layer neural network and fuzzy TSK network.

Neural network of architecture MLP consists of three fully-communication layers with the twelve inputs and one output. To each of hidden nodes and the output node for different significance compounds was applied transformation on the basis of sigmoidal function [1]. Training of the neural network is made with the help of Backpropagation. For the qualitative training of the network were chosen the most optimal training parameters: α - incidence parameter of sigmoidal function, β - shift parameter of the function, η - the coefficient of speed of training, μ - the coefficient of inertia. Values of weight coefficients and threshold values to the beginning of the training (i.e., already their close adjustment) filled with random values from intervals $[-0.5, 0.5]$ and $[-1, 1]$ respectively, so that the signal on the network could spread from the input to the output, and the neural network was able to calculate the value of it.

The second IDS based on a system of a fuzzy vent of Takagi-Sugeno-Kanga[2]. Fuzzy network TSK is a multilayered network, consisting of five layers:

1. The first layer performs fuzzyfication of each variable x_j, $(j = 1, 2, \ldots, N)$, using the generalized function of Gauss $\mu_A^{(i)}(x_j)$. This is a parametric layer with parameters $\left(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)}\right)$, subjected to adaptation in the learning process.
2. The second layer performs the aggregation of the individual variables x_j, determining the resulting value of the coefficient of accessories $w_i = \mu_A^{(i)}(x)$ for vector x. This layer is nonparametric.
3. The third layer is a generator of TSK function, calculating values

$$y_i(x) = p_{i0} + \sum_{j=1}^{N} p_{ij} x_j. \tag{1}$$

In this layer also takes place multiplication of signals $y_i(x)$ on values w_i, formed in the previous layer. This is a parametric layer in which linear weight is subject to adaptation p_{ij}, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, defining the function of the implications of the TSK model.

4. The fourth layer consists of two neuron-adders, one of which calculates a weighted sum of the signals $y_i(x)$, and the second defines the sum of weights $\sum_{i=1}^{M} w_i$. This is a nonparametric layer.
5. The fifth layer consists of one output neuron - this is a normalizing layer, in which the weights are subject to normalization in accordance with the formula

$$y(x) = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i y_i(x). \tag{2}$$

In the process of learning is specified parameters only the first (nonlinear) and the third (linear) layers. Hybrid algorithm of learning was selected for the network training.

Corresponding software was developed in the environment of Microsoft Visual Studio 2010 on C# language for building, simulation and training INN.

For normal functioning IDS should receive the input data directly from the flow of network data. Forming the initial data for the neural network method must allow to detect intrusions on non-standard services, additionally the analysis will be performed only for certain fields to detect intrusions upon the protocol TCP/IP. As a component of the NN vector there were chosen those values, which statistical indicators can testify to the implementation of a certain kind of violations of the traffic network. The input vector consists of 12 coordinates.

In the described models the output values that are less than 0.5 indicate the lack of the intrusion, and all that is more than 0.5-availability of the intrusion. The network activity data of the testing host meant for training and testing IDS models were collected with the help of the program "sniffer"" wireshark-win32-1.2.6 and were placed in a special database. This database contains about 200 training of vectors, in which the

system has been trained. The testing host was attacked by well-known intrusions of UDP Flood and TCP Flood. IDS examined the network traffic through every 5 seconds.

In the result of the testing IDS MLP neural network made less false positives than the fuzzy TSK network.

The results of the simulation of IDS based on neural network testify to the prospects of the issue, but for the normal functioning it is necessary to solve a number of issues.

References

[1] Simon Haykin. Neural networks: a complete course, 2 edition. :-M. Publishing House "Williams", pp. 1104 , 2006.

[2] Ossowski S. Neural networks for information processing / Trans. with the floor. I.D. Rudinsky. - placeCityMoscow: Finances and Statistics, pp. 344-346, 2002.

О распределении элементов полугрупп натуральных чисел

Ю. Н. Штейников

Московский государственный университет имени М.В.Ломоносова, Ленинские горы, 119991 Москва, ГСП-1, Россия (E-mail: yuriisht@gmail.com)

Пусть $\subset \mathbb{N}$ - полугруппа, то есть если $_1, _2 \in$, то $_{12} \in A$. В частности, можно взять множество $A = \{n \in \mathbb{N} : \ n \in G \pmod m\}$, где $m \in \mathbb{N}$, а G - мультипликативная подгруппа группы $\mathbb{Z}_m^*$. Например, если положить $m = p^2$, где - простое число и $G = \left\{g \in \mathbb{Z}_{p^2}^* : \ g^{p-1} = 1\right\}$, то мы получаем

$$A = A_p = \left\{n \in \mathbb{N} : \ n^{p-1} \equiv 1 \pmod{p^2}\right\}.$$

Нас будет интересовать случай, когда для некоторых действительных q; $\nu < 1$ выполнено неравенство:

$$|\{n \in A;\, n \le q\}| < q^{\nu}. \tag{1}$$

Например, пусть $= {}_p$. Так как группа $\mathbb{Z}^*_{p^2}$ - циклическая, то отсюда следует, что $G = \ -1$. Значит для этого примера $|\{n \in A : n \le p^2\}| = \ -1$. Здесь, как несложно видеть, можно положить $q = p^2$ и $\nu = \frac{1}{2}$.
Пусть для > 0 определим:

$$f(x) = |A \cap [1, x]|.$$

Мы хотим оценить сверху $f(x)$, как функцию от q и от u.
В работе [3] получены оценки на количество чисел не превосходящих n, которые принадлежат подгруппе порядка t группы $\mathbb{Z}^*_p$. Эти оценки содержательны, когда t мало по сравнению с p. Из нашей работы вытекают оценки в случае, когда t растет как степень p, а n мало.
Покажем, что верно следующее утверждение.

Theorem 1. *Пусть полугруппа A удовлетворяет условию (1) и $x = (\log q)^u$.*

1) если $\log\log x = o(\log\log q)$, *то*

$$\frac{f(x)}{x} \le \exp\left\{-(C + o(1))u(1-\nu)^2 \log\left(u(1-\nu)^2\right)\right\},$$

где C - некоторая абсолютная константа.

2) если $\gamma = \frac{\log\log x}{\log\log q}$ *и* $\log x = o(\log q)$, *то*

$$f(x) \le x^{1 - C_\gamma + o(1)},\ q \to \infty,$$

где $C_\gamma = \frac{(1-\nu)^2\gamma}{4(1-\gamma)}$, *если* $\gamma \le \frac{2}{3-\nu}$ *и* $C_\gamma = 2 - \nu - \frac{1}{\gamma}$, *если* $\gamma > \frac{2}{3-\nu}$.

The calculating of the admissible sequences

V. V. Shvyrov

Luhansk Taras Shevchenko National University, Oboronna Str., 2., 91011 Luhansk, Ukraine (E-mail: slsh@i.ua)

Let R is a serial ring (see [2]). Denoting the composition lengths of the indecomposable projective right R-modules by $c_i = Len(e_iR)$, then the sequence $c_1, c_2, \ldots, c_n$ is called an *admissible sequence* for R. In this case we have follow inequalities (see [1]):

$$2 \leq c_i \leq c_{i-1} + 1 \text{ for } i = 2, \ldots, n; \quad c_1 \leq c_n + 1.$$

For the calculating all admissible sequences length n we can use follow VBA code for Excel application, where cell (2,5) contain the needed length parameter for the sequences.

```
  Dim n, buk1 As Integer
  n = Cells(2, 5).Value-1
  Cells(4, 1).Value = "1":  Cells(4, 2).Value = "12"
  For i = 2 To n
   tt = 4:  z = 3
   While Cells(tt, i).Value <> ""
  temp = Cells(tt, i).Value :  temp1 = Mid(temp, Len(temp),
1)
  For j = CInt(temp1) + 1 To 2 Step -1
  z = z + 1 :  Cells(z, i + 1).Value = CStr(temp) & CStr(j)
  Next j:  tt = tt + 1 :  Wend :  Next i
```

Theorem 1. *The number of admissible sequences of length n is equal to the n-th Catalan number.*

n=1	n=2	n=3	n=4	n=5
1	12	123	1234	12345
		122	1233	12344
			1232	12343
			1223	12342
			1222	12334
				12333
				12332
				12323
				12322
				12234
				12233
				12232
				12223
				12222
A=1	A=1	A=2	A=5	A=14

Table1. List of the admissible sequences for $n = 1, \ldots, 5$.

References

[1] F.W. Anderson and K.R. Fuller, *Rings and Categories of Modules*, Springer- Verlag, New York and Berlin, 2nd ed., 1992.

[2] N.M. Gubareni, V.V. Kirichenko, *Rings and Modules.//* Czestochowa. - 2001. - 306 p.

On the number of zeros of some analytic functions related to the Hurwitz zeta-function

Darius Šiaučiūnas

Šiauliai University, P. Višinskio str. 19, LT-77156 Šiauliai, Lithuania (E-mail: siauciunas@fm.su.lt)

The Hurwitz zeta-function $\zeta(s, \alpha)$, $s = \sigma + it$, with parameter α, $0 < \alpha \leq 1$, is defined, for $\sigma > 1$, by Dirichlet series

$$\zeta(s, \alpha) = \sum_{m=0}^{\infty} \frac{1}{(m + \alpha)^s},$$

and by analytic continuation elsewhere. It is known that $\zeta(s,\alpha)$ has zeros in the half-plane $\sigma > 1$, and, for rational or transcendental parameter α, in the critical strip $\{s \in \mathbb{C} : \ \frac{1}{2} < \sigma < 1\}$.

In the report, we construct classes of functions F such that the composite functions $F(\zeta(s,\alpha))$ have infinitely many zeros in the critical strip. We give one example of such functions. Let $D = \{s \in \mathbb{C} : \ \frac{1}{2} < \sigma < 1\}$. Denote by $H(D)$ the space of analytic functions on D equipped with the topology of uniform convergence on compacta. Moreover, for $a_1, \dots, a_r \in \mathbb{C}$, let

$$H_{a_1,\dots,a_r}(D) = \left\{g \in H(D) : \ (g(s) - a_j)^{-1} \in H(D), \ j = 1, \dots, r\right\}.$$

Denote by $U_{a_1,\dots,a_r}$ the class of continuous functions $F : \ H(D) \to H(D)$ such that $F(H(D)) \supset H_{a_1,\dots,a_r}(D)$.

Theorem 1. *Suppose that $F \in U_{a_1,\dots,a_r}$ with $\mathrm{Re} a_j \notin \left(-\frac{1}{2}, \frac{1}{2}\right)$, $j = 1, \dots, r$, and α is a transcendental number. Then, for every σ_1, σ_2, $\frac{1}{2} < \sigma_1 < \sigma_2 < 1$, there exists a constant $c = c(\sigma_1, \sigma_2, \alpha, F) > 0$ such that, for sufficiently large T, the function $F(\zeta(s,\alpha))$ has more than cT zeros in the rectangle $\{s \in \mathbb{C} : \ \sigma_1 < \sigma < \sigma_2, \ 0 < t < T\}$.*

A proof of the theorem is based on the universality in the Voronin sense of the function $F(\zeta(s,\alpha))$.

On simulation of automata over finite ring

V. V. Skobelev

Institute of Applied Mathematics and Mechanics of National Academy of Sciences of Ukraine,
str. Rose Luxemburg 74, 83114 Donetsk, Ukraine (E-mail: vv_skobelev@iamm.ac.donetsk.ua)

Introduction

Applications of algebraic models in cryptography naturally lead to analysis of properties of automata determined via systems of equations over

some finite associative-commutative ring $\mathcal{K} = (K, +, \cdot)$ with unit. If reversible automaton is applied as mathematical model for stream cipher then problems of parametric identification and identification of initial state are the basic ones (since parameters play the role of long-term secret key, while initial state plays the role of short-term secret key). It is evident that solving of these problems is reduced to solving (non-linear, as a rule) systems of equations over the ring $\mathcal{K}$, formed via some experiment with analyzed automaton. It is well known high complexity for solving non-linear systems of equations over finite rings (at least, it is NP-complete one). Thus, there naturally arises problem of design of some model intended to simulate investigated automaton with some exactness. This problem is analyzed in the given paper.

Analyzed model

For fixed $l, n_1, n_2, n_3 \in \mathbb{N}$, $\mathbf{A} \subseteq K^l$ ($|\mathbf{A}| \geq 1$), $\mathbf{f}_1 : K^{n_1+n_2+l} \to K^{n_1}$ and $\mathbf{f}_2 : K^{n_1+n_2+l} \to K^{n_3}$ system of equatins

$$\begin{cases} \mathbf{q}_{t+1} = \mathbf{f}_1(\mathbf{q}_t, \mathbf{x}_{t+1}, \mathbf{a}) \\ \mathbf{y}_{t+1} = \mathbf{f}_2(\mathbf{q}_t, \mathbf{x}_{t+1}, \mathbf{a}) \end{cases} \quad (t \in \mathbb{Z}_+)$$

determines over the ring $\mathcal{K}$ some family of finite automata $\mathcal{M} = \{M_{\mathbf{a}}\}_{\mathbf{a}\in\mathbf{A}}$, such that for any fixed parameters $\mathbf{a} \in \mathbf{A}$ elements $\mathbf{q}_t \in K^{n_1}$, $\mathbf{x}_t \in K^{n_2}$ and $\mathbf{y}_t \in K^{n_3}$ are, correspondingly, state, input and output of the automaton $M_{\mathbf{a}}$ at instant t.

Let $F_{\mathbf{a},\mathbf{q}_0}$ ($\mathbf{a} \in \mathbf{A}, \mathbf{q}_0 \in \mathbf{K}^{n_1}$) be the mapping of input of semigroup $(K^{n_2})^+$ into output of semigroup $(K^{n_3})^+$, determined by initial automaton $(M_{\mathbf{a}}, \mathbf{q}_0)$. Then to any automaton $M_{\mathbf{a}}$ ($\mathbf{a} \in \mathbf{A}$) it corresponds the family of mappings $\mathcal{F}_{\mathbf{a}} = \{F_{\mathbf{a},\mathbf{q}_0}\}_{\mathbf{q}_0\in\mathbf{K}^{n_1}}$.

Simulation model

For fixed $r, l_1 \in \mathbb{N}$, $\mathbf{B} \subseteq K^{l_1}$ ($|\mathbf{B}| \leq |\mathbf{A}|$) and families of mappings $\{\varphi^{(1)}_{\mathbf{b}} : K^{n_1} \times K^{n_2} \to K^{n_3}\}_{\mathbf{b}\in\mathbf{B}}$, $\{\varphi^{(2)}_{\mathbf{b}} : K^{n_1} \times (\bigcup_{j=1}^{r-1} K^{n_3})^j \times K^{n_2} \to K^{n_3}\}_{\mathbf{b}\in\mathbf{B}}$ and $\{\varphi^{(3)}_{\mathbf{b}} : K^{n_1} \times (K^{n_3})^r \times K^{n_2} \to K^{n_3}\}_{\mathbf{b}\in\mathbf{B}}$ we determine the family of mappings $\mathcal{G}_{\mathbf{B}} = \{G_{\mathbf{b}} : K^{n_1} \times (K^{n_2})^+ \to (K^{n_3})^+\}_{\mathbf{b}\in\mathbf{B}}$, such

that $G_{\mathbf{b}}(\mathbf{q}_0, \mathbf{x}_1 \dots \mathbf{x}_m) = \mathbf{y}_1 \dots \mathbf{y}_m$ $(\mathbf{b} \in \mathbf{B}, m \in \mathbb{N})$, where

$$\mathbf{y}_i = \begin{cases} \varphi_{\mathbf{b}}^{(1)}(\mathbf{q}_0, \mathbf{x}_1), \text{ if } i = 1 \\ \varphi_{\mathbf{b}}^{(2)}(\mathbf{q}_0, \mathbf{y}_1 \dots \mathbf{y}_{i-1}, \mathbf{x}_i), \text{ if } i = 2, \dots, r \\ \varphi_{\mathbf{b}}^{(3)}(\mathbf{q}_0, \mathbf{y}_{i-r} \dots \mathbf{y}_{i-1}, \mathbf{x}_i), \text{ if } r < i \le m \end{cases}$$

for any $\mathbf{q}_0 \in K^{n_1}$ и $\mathbf{x}_1 \dots \mathbf{x}_m \in (K^{n_2})^+$. Let $H_{\mathbf{b},\mathbf{q}_0} : (K^{n_2})^+ \to (K^{n_3})^+$ $(\mathbf{b} \in \mathbf{B}, \mathbf{q}_0 \in K^{n_1})$ be mappings, such that $H_{\mathbf{b},\mathbf{q}_0}(\mathbf{x}_1 \dots \mathbf{x}_m) = G_{\mathbf{b}}(\mathbf{q}_0, \mathbf{x}_1 \dots \mathbf{x}_m)$ for any $\mathbf{x}_1 \dots \mathbf{x}_m \in (K^{n_2})^+$ $(m \in \mathbb{N})$.

It is evident that every family $\mathcal{H}_{\mathbf{b}} = \{H_{\mathbf{b},\mathbf{q}_0}\}_{\mathbf{q}_0 \in K^{n_1}}$ $(\mathbf{b} \in \mathbf{B})$ determines some automaton over the ring $\mathcal{K}$. Thus, any surjection $h : \mathbf{A} \to \mathbf{B}$ corresponds to any automaton $M_{\mathbf{a}} \in \mathcal{M}$ some automaton determined via family of mappings $\mathcal{H}_{h(\mathbf{a})}$. We determine ordered pair $(\mathcal{G}_{\mathbf{B}}, h)$ to be simulation model for the family $\mathcal{M}$ of automata.

Exactness of simulation model

It is supposed that for any $\mathbf{a} \in \mathbf{A}$ and $\mathbf{q}_0 \in K^{n_1}$ there exist

$$\underline{\gamma}_{\mathbf{a},\mathbf{q}_0} = \underline{\lim}\gamma_{\mathbf{a},\mathbf{q}_0,m} = \lim_{m\to\infty} \inf\{\gamma_{\mathbf{a},\mathbf{q}_0,i} | i \in \mathbb{N}_m\}$$

and

$$\overline{\gamma}_{\mathbf{a},\mathbf{q}_0} = \overline{\lim}\gamma_{\mathbf{a},\mathbf{q}_0,m} = \lim_{m\to\infty} \sup\{\gamma_{\mathbf{a},\mathbf{q}_0,i} | i \in \mathbb{N}_m\},$$

where

$$\gamma_{\mathbf{a},\mathbf{q}_0,m} = \frac{|K^{n_2}| - 1}{m(|K^{n_2}|^{m+1} - |K^{n_2}|)} \sum_{i=1}^{m} |K^{n_2}|^i \alpha_{\mathbf{a},\mathbf{q}_0,m}$$

and

$$\alpha_{\mathbf{a},\mathbf{q}_0,m} = \sum_{\mathbf{x}_1 \dots \mathbf{x}_m \in (K^{n_2})^m} (m - \varrho(\mathbf{y}_1 \dots \mathbf{y}_m, \widetilde{\mathbf{y}}_1 \dots \widetilde{\mathbf{y}}_m)),$$

where $\mathbf{y}_1 \dots \mathbf{y}_m = F_{\mathbf{a},\mathbf{q}_0}(\mathbf{x} \dots \mathbf{x}_m)$, $\widetilde{\mathbf{y}}_1 \dots \widetilde{\mathbf{y}}_m = H_{h(\mathbf{a}),\mathbf{q}_0}(\mathbf{x} \dots \mathbf{x}_m)$ and $\varrho(\mathbf{y}_1 \dots \mathbf{y}_m, \widetilde{\mathbf{y}}_1 \dots \widetilde{\mathbf{y}}_m)$ is Hamming distance between words $\mathbf{y}_1 \dots \mathbf{y}_m$ and $\widetilde{\mathbf{y}}_1 \dots \widetilde{\mathbf{y}}_m$.

Let $\underline{\eta} = \min\limits_{\mathbf{a}\in\mathbf{A}} \min\limits_{\mathbf{q}_0 \in K^{n_1}} \underline{\gamma}_{\mathbf{a},\mathbf{q}_0}$ and $\overline{\eta} = \max\limits_{\mathbf{a}\in\mathbf{A}} \max\limits_{\mathbf{q}_0 \in K^{n_1}} \overline{\gamma}_{\mathbf{a},\mathbf{q}_0}$. It is natural to determine ordered pair $(\mathcal{G}_{\mathbf{B}}, h)$ to be to $[\underline{\eta}, \overline{\eta}]$-exact simulation model for $\mathcal{M}$.

Now we consider special case, when $\underline{\gamma}_{\mathbf{a},\mathbf{q}_0} = \overline{\gamma}_{\mathbf{a},\mathbf{q}_0} = \gamma_{\mathbf{a},\mathbf{q}_0}$ for all $\mathbf{q}_0 \in K^{n_1}$. We can set $\nu_1 = \min\limits_{\mathbf{a}\in\mathbf{A}} \eta_{\mathbf{a}}$, $\nu_2 = |\mathbf{A}|^{-1} \sum\limits_{\mathbf{a}\in\mathbf{A}} \eta_{\mathbf{a}}$, $\nu_3 = \min\limits_{\mathbf{a}\in\mathbf{A}} \zeta_{\mathbf{a}}$ and $\nu_4 = |\mathbf{A}|^{-1} \sum\limits_{\mathbf{a}\in\mathbf{A}} \zeta_{\mathbf{a}}$, where $\eta_{\mathbf{a}} = \min\limits_{\mathbf{q}_0\in K^{n_1}} \gamma_{\mathbf{a},\mathbf{q}_0}$ and $\zeta_{\mathbf{a}} = |K^{n_1}|^{-1} \sum\limits_{\mathbf{q}_0\in K^{n_1}} \gamma_{\mathbf{a},\mathbf{q}_0}$. Numbers $\nu_1, \ldots, \nu_4$ cover all combinations 'in the worst case' and 'in average'. Thus, we can determine ordered pair $(\mathcal{G}_{\mathbf{B}}, h)$ to be ν-exact ($\nu \in \{\nu_1, \ldots, \nu_4\}$) simulation model for $\mathcal{M}$. It is natural todetermine ordered pair $(\mathcal{G}_{\mathbf{B}}, h)$ to be asymptotically exact simulation model $(\mathcal{G}_{\mathbf{B}}, h)$ as one, if $\nu_1 = \cdots = \nu_4 = 1$.

Design of asymptotically exact simulation model is illustrated for the family of automata

$$\begin{cases} q_{t+2} = a_1 + a_2 q_{t+1}^2 + a_3 q_t + a_4 x_{t+1} \\ y_{t+1} = a_5 q_{t+2} \end{cases} \quad (t \in \mathbb{Z}_+),$$

with the set of parameters $\mathbf{A} = \{(a_1, a_2, a_3, a_4, a_5) | a_1, a_2, a_3 \in K\backslash\{0\}; a_4, a_5 \in K^{inv}\}$, where K^{inv} is the set of invertible elements of the ring $\mathcal{K}$.

Conclusions

It is elaborated some approach for design models intended to simulate with some exactness a family of automata determined via system of equations with parameters over finite ring $\mathcal{K}$. Characterization of non-trivial classes of families of automata over the ring $\mathcal{K}$ for which design of sufficiently exact simulation model is either hard problem or easy problem forms some trend of research. Another trend forms characterization of non-trivial classes of families of automata over the ring $\mathcal{K}$ for which exist $[\underline{\eta}, \overline{\eta}]$-exact model such that $|\overline{\eta} - \underline{\eta}| < \varepsilon$, where ε is given positive number.

Multiplicative functions weighted by the Kloosterman sums

Tran The Vinh

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: ttvinhcntt@yahoo.com.vn)

Let $f(n)$ be the multiplicative functions and let $f = g * h$ is a representation of f as Dirichlet convolution for multiplicative functions g and h. And, moreover, we assume in the half-plane $\Re(s) > 1$ the generative series

$$G(s) = \sum_{n=1}^{\infty} \frac{g(n)}{n^s}, \quad H(s) = \sum_{n=1}^{\infty} \frac{h(n)}{n^s}$$

converges.

We construct the asymptotic formula for the summatory functions of type $\sum\limits_{n \le x} f(n)K(1, n; q)$, where $K(a, b; q)$ in the classical Kloosterman sum.

Such summatory functions usually have application in analytic number theory and also in cryptography and coding theory.

We use the following lemma

Lemma 1. *Let $q = q_1 q_2$, $(q_1, q_2) = 1$, q_1 is a square-free, q_2 is a square-full positive numbers. Let $(n, q) = d_1 d_2$, $d_1 | q_1$, $d_2 | q_2$. Then for $d_1 d_2 > 1$ we have*

$$K(1, n; q) = \begin{cases} \mu(d_1) K\left(1, n d_1^2; \frac{q}{d_1}\right) & if\, d_2 = 1, \\ 0 & if\, d_2 > 1. \end{cases}$$

We prove the following relation

$$\sum_{n=1}^{\infty} \frac{f(n)K(1, n; q)}{n^s} = \sum_{d_1 | q_1} \frac{\mu(d)}{d^s} \sum_{a_1 a_2 \equiv 1 \pmod{\frac{q}{d}}} \sum_{b_1, b_2 | \frac{q}{d}} \frac{\mu(b_1)\mu(b_2)}{(b_1 b_2)^s} \times$$
$$\times \sum_{d_1 d_2 = d} G\left(s; \frac{a_1 b_1 d_1 \overline{d}}{\frac{q}{d}}\right) H\left(s; \frac{a_2 b_2 d_2 \overline{d}}{\frac{q}{d}}\right),$$

where $\overline{d}$ is a multiplicative inversive $\bmod \frac{q}{d}$, $\Re(s) > 1$.

For $f(n) = \sigma(n)$, where $\sigma(n)$ is a sum of divisors of n, we have

Corollary 1. *For $q > 1$ we have*

$$\sum_{n\le x}\sigma(n)K(1,n;q)=\frac{C_0(q)}{q}+\frac{C_1(q)x}{q}+\frac{C_2(q)x^2}{q}+0\left(x^{8/5}q^{7/12}\tau(q)\log x\right),$$

where the constants $C_0(q)$, $C_1(q)$ are computable, $1 \le C_2(q) \le 2^{\omega(n)}$.

Exponential sums on PRN's

P. D. Varbanets

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: varb@sana.od.ua)

Let p be a prime number, $m > 1$ be a positive integer. Consider the following recursion

$$y_{n+1} \equiv a\overline{y}_n + b(mod\ p^m), (a, b \in \mathbb{Z}), \tag{1}$$

where $\overline{y}_n$ is a multiplicative inversive *mod* p^m for y_n if $(y_n, p) = 1$. The parameters a, b, y_0 we call the multiplier, shift and initial value, respectively.

In the works of Eichenauer, Lehn, Topuzoğlu, Niederreiter, Flahive, Shparlinski, Grothe, Emmerih etc. were proved that the inversive congruential generator (1) produces the sequence $\{x_n\}$, $x_n = \frac{y_n}{p^m}$, $n = 0, 1, 2, \ldots$, which passes s-dimensional serial tests on equidistribution and statistical independence for $s = 1, 2, 3, 4$ if the defined conditions on relative parameters a, b, y_0 are accomplishable.

It was proved that this generator is extremely useful for Quasi-Monte Carlo type application (see, [3],[4]). The sequences of PRN's can be used for the cryptographic applications. Now the initial value y_0 and the constants a and b are assumed to be secret key, and then we use the output of the generator (1) as a stream cipher. By the works [1],[2] it follows that we must be careful in the time of using the generator (1).

In the current paper we give the generalization for the generator (1).

We consider the following recursive relation

$$y_{n+1} \equiv a\overline{y}_n + b + cF(n+1)y_0(mod\ p^m) \tag{2}$$

under conditions

$(a,p) = (y_0,p) = 1,\ \ b \equiv c \equiv 0(mod\ p),\ \ F(u)$ is a polynomial over $\mathbb{Z}[u]$.

The generator (2) we call the generator with a variable shift $b+cF(n+1)y_0$. The computational complexity of generator (2) is the same as for the generator (1), but the reconstruction of parameters a, b, c, y_0, n and polynomial $F(n)$ is a tricky problem even if the several consecutive values $y_n, y_{n+1}, \ldots, y_{n+N}$ will be revealed. Thus the generator (2) can be used in the cryptographical applications. Notice that the conditions $(a,p) = (y_0,p) = 1$, $b \equiv c \equiv 0(mod\ p)$ guarantee that the recursion (2) produces the infinite sequence $\{y_n\}$.

The main purpose of our talk is showing of passing the test on equidistribution and statistical independence for the sequence $\{x_n\}$, $x_n = \frac{y_n}{p^m}$, and hence, the main point to be shown is the possibility for such sequences to be used in the problem of real processes modeling and in the cryptography. With this aim in view we construct the new estimates of the exponential sums on sequence of pseudorandom numbers generated by the different types of congruential generators.

References

[1] S.R. Blackburn, D. Gomez-Peres, I. Gutierrez and I. Shparlinski. Predicting nonlinear pseudorandom number generators. *Math. Comp.*, 74(251):1471–1494, 2004.

[2] S.R. Blackburn, D. Gomez-Peres, I. Gutierrez and I. Shparlinski. Reconstructing noisy polynomial evaluation in residue rings. *J. of Algorithm*, 61(2):47–59, 2006.

[3] H. Niederreiter. Random number generation and Quasi-Monte Carlo methods. *SIAM, Philadelphia*, 1992.

[4] H. Niederreiter and I. Shparlinski. Recent advances in the theory of nonlinear pseudorandom number generators. *Proc. Conf. on Monte*

Carlo and Quasi-Monte Carlo Methods, 2000, Springer-Verlag, Berlin, 86–102, 2002.

Norm Kloosterman Sums over $\mathbb{Z}[i]$.

S. P. Varbanets

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: varb@sana.od.ua)

The classical Kloosterman sums and their generalizations find the various applications in additive number theory.

In the work[1] there are considered the applications of the Kloosterman sums over the ring of Gaussian integers for the lower bounds of the moments of Hecke zeta-function to be obtained. In our talk we study the n-dimensional norm Kloosterman sums over the ring of Gaussian integers that have no analogue in the ring $\mathbb{Z}$. Such estimates it is possible to apply for constructing the asymptotical formulas of summatory functions associated with the divisor function over $\mathbb{Z}[i]$ in arithmetic progression.

For the Gaussian integers $\alpha_0, \alpha_1, \dots, \alpha_n$ and positive integer h we define n-dimensional Kloosterman sum

$$\widetilde{K}(\alpha_0, \alpha_1, \dots, \alpha_n; q, h) := \sum_{S(C)} e^{2\pi i \Re(\alpha_0 x_0 + \dots \alpha_n x_n)/q},$$

where the notation $S(C)$ means that summation passes under condition

$$C : \{x_j \in \mathbb{Z}[i] \diagup q\mathbb{Z}[i],\ j = 0, 1, \dots, n;\ N(x_0, x_1, \dots, x_n) \equiv h \pmod q\}.$$

(here $N(x)$ denotes the norm of $x \in \mathbb{Z}[i]$, i.e. $N(x) = (\Re(x))^2 + (\Im(x))^2$).

We will obtain the non-trivial estimates for $\widetilde{K}(\alpha_0, \alpha_1, \dots, \alpha_n)$. In particular, we have

Theorem 1. *Let h is a norm residue modulo p and $(h,p)=1$ and let $\alpha_0 \in \mathbb{Z}[i]$, $\alpha_0 \not\equiv 0 \pmod{p}$. Then*

$$|\widetilde{K}(\alpha_0,\alpha_1,\dots,\alpha_n;p^m,h)| \le 2(4n-1)p^{2n(m-m_0)}I(\alpha_1,\dots,\alpha_n;p^m),$$

where $I(\alpha_1,\dots,\alpha_n;p^m)$ is the number of solutions of the system of congruences over unknowns $u_j, v_j \in \mathbb{Z}_{p^{m-m_0}}$

$$\begin{cases} a_j v_j + b_j u_j \equiv 0 \pmod{p^{m-m_0}}, \\ N(\alpha_0)u_j + 2ka_j \prod\limits_{j=1}^{n} \left(u_j^2+v_j^2\right)^2 \equiv 0 \pmod{p^{m-m_0}}, \\ j=1,\dots,n. \end{cases}$$

(here $m_0=[m+1/2]$).

Let χ be a Dirichlet character modulo q_1, $q_1|q$. We study the twisted norm Kloosterman sum

$$\widetilde{K}_\chi(\alpha,\beta;q,h) = \sum_{\substack{x,y\in(\mathbb{Z}[i]/q\mathbb{Z}[i]) \\ N(xy)\equiv h \pmod{q}}} \overline{\chi}(x)e^{2\pi i\Re(\alpha x+\beta y)/q}.$$

For sum $\widetilde{K}_\chi(\alpha,\beta;q,h)$ we also obtain "the root" estimate. These results generalize the results from [2],[3].

References

[1] R.W.Bruggeman, Y.Motohashi, Sum formula for Kloosterman sums and fourth moment of the Dedekind zeta-function over the Gaussian number field, Functiones et Approximatio, XXXI, (2003), 23-92.

[2] Varbanets S.P., The norm Kloosterman sums over $\mathbb{Z}[i]$, Anal. Probab. Methods Number Theory, A. Laurinčikas and E. Manstavičius(Eds.), (2007), 225-239.

[3] Savastru O. Varbanets S., Norm Kloosterman sums over $\mathbb{Z}[i]$, Algebra and Discrete Mathematics, 11(2), (2011), 82-91.

Systems of representation of real numbers generated by Fibonacci sequences and their modifications

N. M. Vasylenko

National Pedagogical Dragomanov University, Pyrogova St. 9, 01601, Kyiv, Ukraine (E-mail: samkina_nata@mail.ru)

System of representation of numbers (numeration system) is a set of tools for notation of numbers. Traditionally positional s-adic numeration system ($2 \leq s \in \mathbb{N}$) is widely used for representation of real numbers. Moreover, the interest in representation of real numbers by the special series is increased during the last centuries. Engel, Cantor, Lüroth, Oppenheim, Ostrogradsky-Pierce series et al. [1, 3] are among them.

Such interest in different forms of representation of real number is quite natural. The development of different theories of representation of real number allows us to extend our knowledge about real number and solve some problems.

Let us recall that number sequence (a_n) is called a *Fibonacci sequence* if

$$a_{n+2} = a_{n+1} + a_n, \quad a_1, a_2 \in \mathbb{R}.$$

Simplest example of Fibonacci sequence is a classic sequence: $u_0 = u_1 = 1$, $u_2 = 2,\ u_3 = 3,\ \ldots$.

One can use Fibonacci sequences to construct systems of representation of real numbers and develop metric, ergodic, fractal and probabilistic theories of real numbers on their bases.

Representation 1. It is known [4] that the sequence of numbers reciprocal to terms of classic Fibonacci sequence can generate the system of representation of numbers belonging to $\left[0, \sum\limits_{n=1}^{\infty} \frac{1}{u_n} = S\right]$ with two-symbol alphabet.

Theorem 1. *For any real number* $x \in [0, S]$ *the following expansion holds:*

$$x = \sum_{n=1}^{\infty} \frac{f_n}{u_n} = \Delta_{f_1 f_2 \ldots f_n \ldots}, \quad f_n \in \{0, 1\}. \tag{1}$$

Expansion of real number $x \in [0, S]$ in the form of series (1) is called a *Φ-representation*, and f_n is called nth digit of Φ-representation of x (nth Φ-symbol).

It is also known [5] that there exists a continuum set of ways to represent any number from $(0, S)$ in the form (1). It can be a positive importance as well as it says about weak point of this representation (for example, for identification of number or comparison of numbers).

Representation (1) whose Φ-symbols determined by the formulae:

$$f_1(x) = \begin{cases} 1, \text{if } \frac{1}{u_1} \le x, \\ 0, \text{if } \frac{1}{u_1} > x, \end{cases} \quad f_i(x) = \begin{cases} 1, \text{if } \sum\limits_{n=1}^{i-1} \frac{f_n(x)}{u_n} + \frac{1}{u_i} \le x, \\ 0, \text{if } \sum\limits_{n=1}^{i-1} \frac{f_n(x)}{u_n} + \frac{1}{u_i} > x, \end{cases}$$

$2 \le i \in \mathbb{N}$, is **unique** and it is called a **canonical Φ-representation**.

Theorem 2. *If* $\Delta^*_{f_1(x) f_2(x) \ldots f_k(x) \ldots}$ *is a canonical* Φ*-representation of some real number* $x \in [0, S]$*, then for any positive integer* k*,*

$$f_{2k}(x) f_{2k+1}(x) f_{2k+2}(x) \ne 011, \quad f_k(x) f_{k+1}(x) f_{k+2}(x) f_{k+3}(x) \ne 0111.$$

Corollary 1. *If* Φ*-representation of real number contains combination of consecutive digits* 0111*, then it is not canonical.*

Corollary 2. *If* Φ*-representation of real number contains combination of consecutive digits* 011*, and* 0 *is on even place, then it is not canonical.*

Transition to canonical Φ-representation allows to solve a problem on comparison of two arbitrary numbers belonging to $[0, S]$.

Theorem 3. *Two arbitrary real numbers from* $[0, S]$ *represented by their canonical* Φ*-expansion are equal if and only if their respective* Φ*-symbols coincide.*

Representation 2. There are infinitesimal sequences among Fibonacci sequences. They form one-dimensional linear space, and its simplest representative is a sequence $\left(\widehat{\varphi}^{n-1}\right)$, where $\widehat{\varphi} = \frac{1-\sqrt{5}}{2}$, $n \in \mathbb{N}$.

It is known [2] that any real number $x \in [-1, \varphi]$ can be represented in the form

$$x = \sum_{n=1}^{\infty} \varepsilon_n \widehat{\varphi}^{n-1} = \varepsilon_1 \widehat{\varphi}^0 + \varepsilon_2 \widehat{\varphi}^1 + \cdots + \varepsilon_n \widehat{\varphi}^{n-1} + \cdots = \Delta_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_n \ldots}, \quad (2)$$

where $\varphi = -\widehat{\varphi}^{-1}$, $\varepsilon_n \in \{0, 1\}$.

The Equality (2) is called the *$\widehat{\varphi}$-representation* of this number.

As in first case almost all (with respect to Lebesgue measure) numbers from $(-1, \varphi)$ can be represented in the form (2) by continuum number of ways. We define canonical $\widehat{\varphi}$-representation as a most economical and convenient for development of the theory.

Definition 1. $\widehat{\varphi}$-representation of number x from $[-1, \varphi]$ is called the canonical representation if $\varepsilon_k \varepsilon_{k+1} \varepsilon_{k+2} \neq 001$ for any $k \in \mathbb{N}$.

Theorem 4. *Any real number $x \in [-1, \varphi]$ has a canonical representation moreover it is unique.*

Theorem 5. *The following equality holds:*

$$[-1, \varphi] = \bigcup_{c \in \{0,1\}} \left(\Delta_{110} \cup \Delta_{c110} \cup \Delta_{c(1-c)110} \cup \Delta_{c(1-c)c110} \cup \ldots \right).$$

It is easy to show that for cylinders in the form

$$\Delta_{\underbrace{c(1-c)\ldots c(1-c)}_{2k}110} \quad \text{and} \quad \Delta_{\underbrace{c(1-c)\ldots(1-c)c}_{2k+1}110},$$

there exists partition analogous to partition in Theorem 5. Using the above mentioned partitions one can prove that for any real number $x \in [-1, \varphi]$ there exists sequence $(\alpha_n) \in \mathbb{Z}$ such that $x = \Delta^{\infty}_{\alpha_1 \alpha_2 \ldots \alpha_n \ldots}$.

In the talk, we provide a comparative analysis of representation of real numbers using two different Fibonacci sequences (classic and infinitesimal) and give some modifications for representation of real numbers using the latter.

References

[1] Albeverio S., Baranovskyi O., Pratsiovytyi M., Torbin G. The Ostrogradsky series and related Cantor-like sets, *Acta Arith.*, Vol. 130, no. 3, pp. 215-230, 2007.

[2] Vasylenko N. M. On fractal properties of some Cantor-like sets generated by $\widehat{\varphi}$-representation of real numbers, *Bulletin of University of Kyiv,* Series: Physics & Mathematics, **3,** pp. 121-124, 2010.

[3] Viader P., Paradis J., J. Miralles de Imperial, Bibiloni L. Els sistemes de representacio dels nombres reals (I) *Butlleti de la Societat Catalana de Matematiques*, **16**, num. 2, pp. 91-120, 2001.

[4] *Василенко Н. М.* Фібоначчієві подання дійсних чисел // Науковий часопис НПУ імені М. П. Драгоманова. Серія 1. Фіз.-мат. науки. — Київ: НПУ імені М. П. Драгоманова, 2005. — **6.** — С. 261-271.

[5] *Василенко Н. М.* Деякі метричні співвідношення, породжені Φ-зображенням дійсних чисел // Науковий часопис НПУ імені М. П. Драгоманова. Серія 1. Фіз.-мат. науки. — Київ: НПУ імені М. П. Драгоманова, 2006. — **7.** — С. 190-203.

Some results about Piltz's divisor problem over the matrix ring $M_2(\mathbb{Z})$

I. N. Velichko

I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: velichko_student@mail.ru)

Let $M_k(\mathbb{Z})$ denotes the ring of integer matrices of order k, $GL_k(\mathbb{Z})$ is the unite group of $M_k(\mathbb{Z})$. We denote the number of different (to association) representations of matrix $C \in M_k(\mathbb{Z})$ in the form $C =$

$A_1A_2A_3,\ A_1, A_2, A_3 \in M_k(\mathbb{Z})$ as $\tau_3^{(k)}(C)$. It is interesting to investigate an asymptotic behavior of the sum

$$T_{3,(k)}(x) = \sum_{n \le x} {\sum_{G \in M_k(\mathbb{Z}),\ |det\, G| = n}}' \tau_3^{(k)}(G)$$

for different k (the second sum is taken throughout all matrices G accurate to integer unimodular factor).

G.Bhowmik and H.Menzer [1], H.-Q.Liu [2], A.Ivič [3], N.Fugelo and I.Velichko [4] studied an asymptotic behavior of the similar sums, but only for two factors.

Our aim is to construct the asymptotic formula for the function $T_{3,(2)}(x)$ and to investigate the second moment of error term.

Using Perron's summation formula and the estimates of the sixth, eighth and tenth moments of $\zeta(s)$, the following results have been obtained :

Theorem 1. *For $x \to \infty$ the estimate*

$$T_{3,(2)}(x) = xP_5(\log x) + O(x^{23/30})$$

holds, where $P_5(u)$ is a polynomial of degree 5.

Theorem 2. *Let $\Delta_{3,(2)}(x) = \sum\limits_{n \le x} t_3^{(2)}(n) - xP_5(\log x)$, where $P_5(u)$ is the polynomial from previous theorem. Then for $x \to \infty$ we have the estimate*

$$\int_1^x (\Delta_{3,(2)}(x))^2 dx \ll x^{29/12}. \qquad (1)$$

References

[1] Bhowmik G., On the number of subgroups of finite Abelian groups / G. Bhowmik, H. Menzer // Abh. Math. Sem. Univ. Hamburg. – 1997. – N 67 – P. 117–121.

[2] Liu H.-Q., Divisor problems of 4 and 3 dimensions / H.-Q. Liu // Acta Arith. – 1995. – N 73 – P. 249–269.

[3] Ivič A., On the number of subgroups of finite abelian groups / A. Ivič // Theorie Nombres de Bordeaux. – 1997. –N 9 – P. 371–381.
[4] Fugelo N., Average orders of divisor function of integer matrices in $M_3(\mathbb{Z})$ / N. Fugelo, I. Velichko // Annales Univ. Sci. Budapest. – 2008. – N 28 – P. 261–270.

Inversive Congruential Generator of Complex Numbers

P. D. Varbanets[1] and S. A. Zadorozhny[2]

[1] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: varb@sana.od.ua)

[2] I.I. Mechnikov Odessa National University, str. Dvoryanskaya 2, 65026 Odessa, Ukraine (E-mail: s_zador@mail.ru)

There is a well-known Erdos-Turan-Koksma inequality. It estimates the discrepancy of a sequence of real numbers. In our article we proof a similar estimate but for the sequence of complex numbers.

Denote the ring of Gaussian integers as $\mathbb{Z}[i] := \{a+ib \mid a,b \in \mathbb{Z},\ i^2 = -1\}$. For $M > 1$ natural number let $\mathbb{Z}_M[i]$ ($\mathbb{Z}_M^*[i]$) denotes complete system of residues (reduced system of residues). For any complex number $c = a + ib$, $N(c) = a^2 + b^2$, $Sp(c) = 2a$.

Theorem 1. *Let $M > 1$ is an integer. Then for any sequence of points $\{y_n\}$, $y_n \in \mathbb{Z}_M[i]$ discrepancy of $\{y_n/M\}$ satisfies to the inequality*

$$D_N \le \frac{2\pi}{M} + \frac{2\log^2 M}{M^{5/3}} \sum_{\substack{\alpha \in \mathbb{Z}_M[i] \\ \alpha \neq 0}} \left(\frac{1}{M^2} + \frac{1}{N(\alpha)}\right) \frac{1}{N} \left| \sum_{n=0}^{N-1} \exp\left(\pi i Sp\left(\frac{\alpha y_n}{M}\right)\right)\right|. \tag{1}$$

Define the inverse generator of complex numbers. Let $\mathfrak{p} \in \mathbb{Z}[i]$ is a Gaussian prime, m is a natural number, $\alpha, \beta \in \mathbb{Z}[i]$, $(\alpha, \mathfrak{p}) = 1$, $\beta \equiv 0 \pmod{\mathfrak{p}}$, w_0 be the Gaussian integer, $(\omega, \mathfrak{p}) = 1$, then

$$w_{n+1} = \alpha w_n^{-1} + \beta \pmod{\mathfrak{p}^m}, \qquad w_n^{-1} w_n \equiv 1 \pmod{\mathfrak{p}^m} \tag{2}$$

We also apply the above theorem to the generator (2) and got the following result

Theorem 2. *Let* $\mathfrak{p}$ *is a Gaussian prime,* $\mathfrak{p} \neq 1+i$*,* α*,* β*,* w_0 *are Gaussian integers,* $(\alpha, \mathfrak{p}) = 1$*,* $\nu_{\mathfrak{p}}(\beta) = b \geq 1$*,* $(w_0, \mathfrak{p}) = 1$*,* $m \geq 2b$ *is an integer. Then for the sequence* $\{x_n\}$*,* $x_n = w_n/\mathfrak{p}^m$*, эде* w_n *which is defined by* (2)*, where* $\alpha \not\equiv w_0^2 \pmod{\mathfrak{p}}$*, we have*

$$D_N(x_0, x_1, \ldots, x_{N-1}) \leq \frac{2\pi}{p^m} + \frac{3p^{5m/6}}{N} \log^3 p^m.$$

(here, $p = N(\mathfrak{p})$ *if* $p \notin \mathbb{Z}$*, and* $p = \mathfrak{p}$ *if* $p \in \mathbb{Z}$*)*

Thus the conclusion can be made that the sequence $\{w_n\}$ is uniformly distributed in a one-circle.

References

[1] Hellekalen P., General discrepancy estimates: the Walsh function system, Acta Arith, 67(1994), pp 209–218

[2] Niederreiter H., Random number generation and quasi-Monte Carlo methods, SIAM, Philadelphia, 1992.

[3] Varbanets P., Varbanets S., Exponential sums on the sequences of inversive congruential pseudorandom numbers with prime-power modulus, Voronoi Impact on Modern Science, Book 4, v.1, Proc. Inform. Conf. Analytic Number Theory and Spatial Tessellations, Kyiv, Ukraine, 2008, pp 112–130.

Petri Net Paradigm of Computation

D. A. Zaitsev

International Humanitarian University, str. Fontaskaya Doroga 33, 65009 Odessa, Ukraine (E-mail: zsoftua@yahoo.com, Web: http://daze.ho.ua)

Petri nets and Modeling of Systems

Petri net [1, 2] is a bipartite directed graphs which a dynamic process is defined on. One part of vertices named places and drawn as circles models conditions; the other part named transitions and drawn as bars models events. Dynamic elements named tokens are situated inside places and moved among them as result of transitions firing. Places, transitions, arcs and tokens are considered either elementary or loaded with additional characteristics/functions creating a series of Petri net subclasses.

Petri nets are traditionally applied for modeling systems in a wide range of application areas [1, 2]. Elementary Petri net standing between finite automaton and Turing machine allows the analytical methods of investigation [2]. Loaded Petri nets [1] are often considered as graphical language of simulation systems and either model-checking approach or statistical analysis is employed as the basic technique.

Writing programs in Petri net language

Since the Turing-completeness of inhibitor Petri nets was proven [3], one can write programs in Petri net language. Many works considered Petri net as a graphical substrate of asynchronous parallel programming language [4, 5]. But Petri net language was often treated as an auxiliary intermediate language which is finally translated into instructions of a definite classical processor.

In fig. 1 an inhibitor Petri net implementing addition of integers is shown: a) source data are put in places x and y, putting token into place s starts the computation; b) the obtained result is observed in place z when place f (finish) is marked. Note that the written sequence

of transitions firing is not obligatory; transitionsв t_2 and t_3 firing order allows arbitrary permutations.

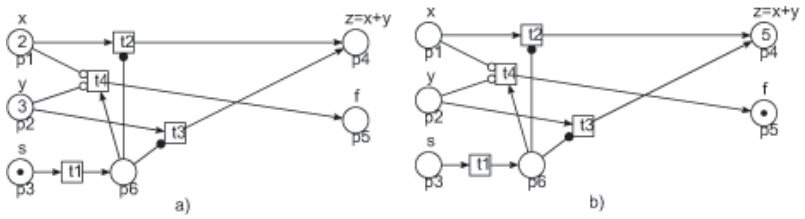


Figure1. Inhibitor Petri net implementing addition of integers: a) initial state; b) final state; a transitions firing sequence is $t_1 t_2^2 t_3^3 t_4$

Model-driven development of nowadays is taking advantage of Petri net models which are transformed into specifications of systems during the process of design. But the final stage supposes a translation of Petri net into a sequence of instructions that throws away many advantages.

Asynchronous approach often opposes the classical parallel approach where algorithms are sequential a priory and there required techniques of their parallelization for execution on a multi-processor computers or clusters. In the asynchronous approach algorithms are written preserving original parallelism of an application area and their further parallelization is not required.

In fig. 2 basic operators of a parallel programming language are represented by inhibitor Petri net corresponding to: a) operator1; operator2; b) if (c) operator1 else operator2; c) while (c) operator; d) parbegin-parend. But writing programs directly in Petri net language allows arbitrary combinations of control and data flows; even control and data flows can be so closely interweaved that hardly distinguishable separately.

Macro and micro levels of Petri net paradigm of computation

Petri net paradigm can be implemented on either macro-level or micro-level of computation. In case of macro-level implementation (fig. 3), Petri net coordinates sequential code fragments written in usual programming language which load transitions, arcs and tokens. Thus Petri net processor is added to a cluster of usual processors.

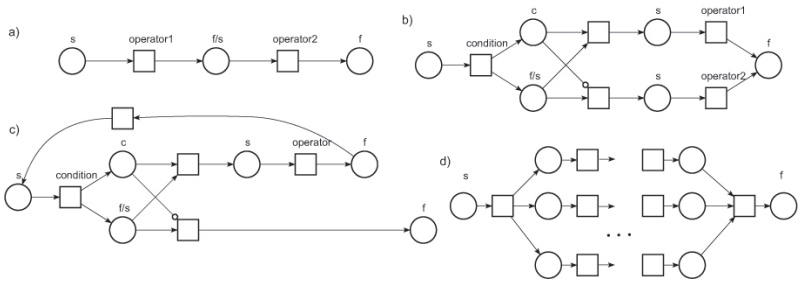


Figure2. Representing programming language operators by Petri nets: a) sequence, b) branching), c) loop, d) parallelization

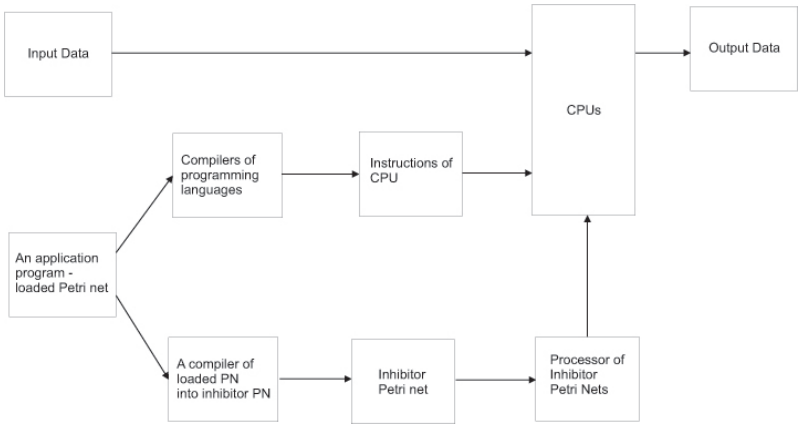


Figure3. Macro-level implementation of Petri net paradigm of computations

Implementation on micro-level (fig. 4) stipulates a whole description of algorithms in Petri nets when there is nothing except Petri net to express data and control flows and operations over them. There are certain inconveniences in description of arithmetic operations by Petri nets and traditionally ineffective way of their implementation that prevents the practical use of micro-level approach.

A certain gap between asynchronous programming theory and the practice of traditional programming for computers with classical architecture was staying permanent for years. The only area which employed Petri net programming language was programmable logical controllers (PLC) [1, 9]. Experimental implementations of Petri net based languages [4, 5] either supposed a compilation into classical languages or interpretation of programs with software built processors of Petri nets. Thus hardware architecture used did not allow taking advantages of asynchronous approach because asynchronous processor was emulated on a synchronous hardware.

Recently, Universal Petri net (UPN) was built [6] as well as Petri nets which execute any given Turing Machine (PNTM) [7] and Normal Algorithm of Markov (PNNAM) [8] providing the compatibility of concepts. UPN is considered a prototype of a Petri net processor (PNP) built in Petri nets while PNTM and PNNAM are prototypes of co-processors executing algorithms written either as sequence of instructions or productions. Thus a program written in Petri net language is executed by UPN and other kinds of code fragments are executed by PNTM, PNNAM etc.

A fragment of UPN, namely subnet calculating firing conditions on transitions is shown in fig. 5. It is constructed on a C program and uses a single control flow represented by the advance of a single token from place s to place f. Variables are situated in the named places in the upper part of the figure. The processing order corresponds to the fragment of C code in fig. 6.

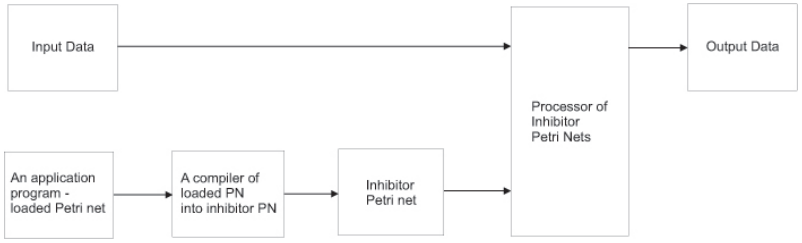


Figure4. Micro-level implementation of Petri net paradigm of computations

It looks like all the necessary job has been done to start the production of hardware Petri net processors and design computers and clusters of them. Although we meet oftener real-life programs written in Petri net [9, 12], something hampers their rapid development. These obstacles are revealed and discussed further.

Constructing efficient Petri net hardware and software

At first, it is not convenient to write programs in elementary classes of Petri nets, for instance, in inhibitor Petri nets. Loaded Petri nets

are convenient and powerful but their hardware implementation is too cumbersome.

At second, constructions used in theory are not effective when applying them in practice. UPN built [6] employs sequential algorithm of arithmetic operations implementation based on increment/decrement primitives and considers a single control flow only which is utmost ineffective though clear and convenient in theory for providing proofs by induction. Thus, new modifications of UPN should be build which maximally employ asynchronous style of processing.

Therefore, two principal gaps are yawning: a gap of Petri net programming technology and a gap of effective Petri net hardware taking advantages of unified asynchronous style.

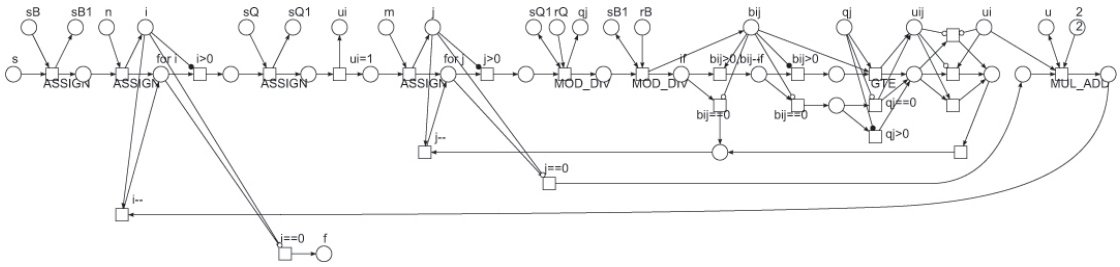


Figure5. A fragment of UPN for finding friable transitions

In micro-level implementations, the class of inhibitor Petri net is employed as an analog of a processor instruction set and loaded classes of Petri nets as programming languages destined for developing applications. The both should be designed: a technology of programming in loaded Petri nets and a technology of translating loaded nets into inhibitor Petri nets.

Software processors of Petri nets employ sequential approach of checking all the transitions on each step that requires in the worst case to check all places for each transition. So when there is a trace of l transitions its implementation requires nml operations, where n is the number of transitions and m is the number of places. Moreover, firable sets of transitions should be fired independently breaking ties of sequential way of perceiving the events.

Synchronous and asynchronous concepts only prevail in various formal systems but do not appear in a pure sort. Even classical Petri net

```
/* traditional implementation of Petri net single step */
for(i=n;i>o;i--)
{
  for(j=m;j>0;j--)
  {
    <processing arcs connecting transition i and place j>
  }
}
```

Figure6. Traditional implementation of a step of the Petri net dynamics

behavior implies synchronization on a step when all the transitions are checked and only one of them is chosen and fired. The class of synchronous Petri nets which the maximal set of friable transitions fires on a step in is Turing-complete as well. But a new class of Petri net is required without definite steps where all the transitions act independently but the result of their firing at some specified instants of observation should be the same.

A new type of hardware directly and effectively implementing Petri net could be named a calculating memory (CM). CM stores Petri net structure and its current state and provides independent calculation firing conditions on all transitions independently (fig. 7). The only local synchronization is required in case of a conflict of a few transitions for a token. But blocking incidental places when a transition fires and signaling other transitions to recalculate their friable conditions could mend this flaw. The closest analog to PNP is Programmable Logical Matrices (PLM) but it works in synchronous way.

A gap in programming technology is mended via effective implementation of basic arithmetic, logic and other kinds of operations. For instance, very basic questions of integer and real numbers representation for providing effective algorithms as well as algorithms themselves should be developed. Places are represented as counters of limited capacity that modifies essentially the scope of prototyped UTM.

Foundation stones and benefits

Technology of programming in hi-level Petri nets, techniques of translating hi-level Petri nets into IPN, and effective hardware PNP executing

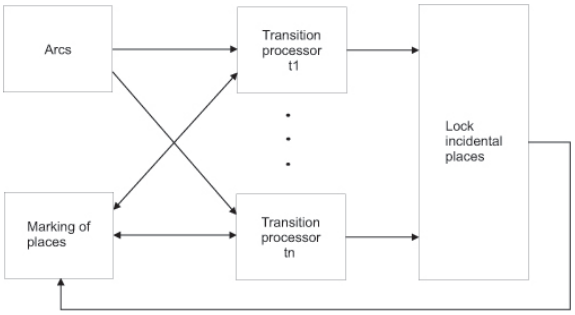


Figure7. Scheme of Petri net processor

inhibitor Petri net, are three foundation stones for promoting Petri net paradigm of computation.

An advantage of considerable speed-up of computation and software development process justifies the efforts required.

References

[1] Sleptsov A. I. and Yurasov A. A. Automation of Designing Control Systems of Flexible Computer-Aided Productions [in Russian], Tekhnika, Kiev (1986).

[2] Murata T. Petri nets: Properties, analysis, and applications, Proc. IEEE, 77, 541-580 (1989).

[3] Agerwala T. A complete model for representing the coordination of asynchronous processes, Hopkins Computer Science Program, Res. Rep., No. 32, John Hopkins University, Baltimore (1974).

[4] Usher M., Jackson D. A Petri net based visual programming language. Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, Volume: 1, 1998 , p. 107 - 112.

[5] Iordache M.V., Antsaklis P.J. Petri nets and programming: A survey. American Control Conference, 2009. ACC '09, 2009, p. 4994 - 4999.

[6] Zaitsev D.A. Universal Petri net. Cybernetics and Systems Analysis, No. 4, 2012, DOI: 10.1007/s10559-012-9429-4.

[7] Zaitsev D.A. Constructing Petri Net which executes Turing Machine. Proc. of IV International Conference Computer Mathematics in Science, Engineering and Education, CMSEE-2010, Poltava (Ukraine), October 1-31, 2010, Kiev: NAS Ukraine Press, P.12-14.

[8] Zaitsev D.A. Inhibitor Petri Net that Executes an Arbitrary Given Markov Normal Algorithm. Modeling and Analysis of Information Systems, 18, 4 (2011) 80-93.

[9] ShihSen Peng, MengChu Zhou Petri net based PLC stage programming for discrete-event control design. Systems, Man, and Cybernetics, 2001 IEEE International Conference on, Volume: 4, 2001 , p. 2706 - 2710.

[10] Rossmann, J.; Eilers, K. Translating robot programming language flow control into Petri nets. Emerging Technologies and Factory Automation (ETFA), 2011 IEEE 16th Conference on Digital Object Identifier, p. 1 - 7.

[11] Palomeras, N., Ridao, P. ; Carreras, M. ; Silvestre, C. Using petri nets to specify and execute missions for autonomous underwater vehicles. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on Date of Conference: 10-15 Oct. 2009, p. 4439 - 4444.

[12] Dodd R.B. Coloured Petri Net Modelling of a Generic Avionics Mission Computer. Air Operations Division: Defence Science and Technology Organisation, Australia, DSTO-TN-0692, 2006, 94 p.

Index