Russian Academy of Sciences
Euler International Mathematical Institute

First Russian-Finnish Symposium on Discrete Mathematics

St.Petersburg, Russia

September 21–24, 2011

Abstracts
(updated version)

St.Petersburg, 2011

The symposium is supported by

- the Russian Academy of Sciences
- the Finnish Academy of Sciences,
- the Dynasty Foundation
- the Russian Foundation for Basic Research (RFBR grant 11-01-06084-r)

This is an updated version of the Abstracts containing talks that were actually presented at the Symposium. The original printed version has ISBN 978-5-9651-0571-7.

Content

Invited lectures

Complexity of solving tropical linear systems and conjecture on a tropical effective Nullstellensatz

Dima Grigoriev
Lille, France

An algorithm for solving tropical linear systems is designed and its complexity is studied. The classical Nullstellensatz can be treated as a reduction of solvability of a polynomial system to solvability of a linear system with the Cayley matrix. The effective Nullstellensatz reduces solvability to a submatrix of a bounded size. A conjecture on a tropical effective Nullstellensatz is discussed. It holds for univariate tropical polynomials.

Current Progress in Majorana Theory

Alexander Ivanov
Imperial College London

In my lecture I will discuss the current state of the Majorana theory.

The Monster book. In April 2009 the book [A.A. Ivanov, *The Monster Group and Majorana Involutions*, Cambridge Univ. Press, Cambridge 2009] has been published. This was a result of over 20 years of research. The main original aim was to provide the first complete construction and uniqueness proofs for the largest and most famous among the 26 sporadic simple groups, known as the Monster group. The proof was have culminated the proof by Ivanov and Norton the so-called Y-conjecture during the Durham symposium on 'Groups and Geometries' in July 1990. At that time the Y-conjecture became a theorem which John Conway called NICE (where 'N' is for Norton, 'I' is for Ivanov, 'C' is for Conway and 'E' is for everyone else involved). The proof was the most spectacular example of the so-called 'Geometric Presentations of Groups'. The classical version of this is precisely the Steinberg presentations for the groups of Lie type, under this title an invited lecture at the Kyoto 1990 International Congress of mathematicians was given by A.A.Ivanov and this is the title of a conference to be held in Birmingham this summer.

Halfway through the actual writing up of the book (October 2005–May 2008). The exceptional importance of the so-called Monster algebra (also known as the Conway–Griess–Norton algebra) has been further appreciated. This came through a result of a Japanese mathematician S. Sakuma, who has classified all the subalgebras in the Monster algebra generated by pairs of axial vectors corresponding to the $2A$ (also known as the Baby Monster) involutions. There are nine isomorphism types of these subalgebras identified and studied by J. Conway and S. Norton. The outstanding importance of Sakuma's result was that the subalgebras were classified under very mild assumptions involving the fusion rules of the eigenspaces of the axial vectors. The far reaching importance of these properties brought about the need of special name for them that is how the term 'Majorana theory' emerged and for the first time was announce at the Oberwolfach conference on 'Groups and Geometries' in April of 2008.

During the 2008 Oberwoffact conference the Abel prises of J. Thompson and J. Tits were also celebrated. At the special session our prominent colleagues shares their memories of the most exciting period of 1970's when the finite simple group were classified and when most spectacular sporadic simple groups. Among them there was Berns Fischer the 'father' of the Monster group. Right after the lecture martin Liebeck suggested to make record of Fischer's story and this is how the last chapter of the Monster group emerged, which is probably the most attractive one (especially for the general mathematical readers).

Majorana representations of groups. After the Majorana setting was axioma-tised it became clear that one should start with classifying the Majorana representations of small groups. At such circumstances one usually starts looking at the A_5's. There are two classes of A_5-subgroups in the Monster whose involutions are of type $2A$. Some preliminary estimates led to a conjecture that the corresponding axial vectors in the Monster algebra generate 26- and 21-dimensional subalgebras, respectively. A conjecture was posed in Chapter 8 of the Monster book that A_5 possess only two Majorana representations corresponding (in a sense which has been made explicit and rigorous) to the subalgebras of the Monster algebras. At that time this was more like a dream and no-one could have believed that in less than two years these conjecture will be corrected (the second representations turned out to be just 20-dimensional) and fully proved. The current status of the classification project of the Majoorana representations of the small groups is the following:

(i) the representations of the dihedral group have been classified in the original paper by S. Sakuma.

(ii) the representations of S_4 are completely classified and the result os published: [A.A. Ivanov, D.V. Pasechnik, Á. Seress, and S. Shpectorov, Majorana representations of the symmetric group of degree 4, *J. Algebra* **324** (2010), 2432-2463.]

(iii) the representations of A_5 are completely classified and the manuscript [A.A. Ivanov and Á. Seress, Majorana Representations of A_5] *Math. Z.* (submitted)

(iv) the representation of $L_3(2)$ are completely classified and the manuscript [A.A. Ivanov and S. Shpectorov, Majorana Representations of $L_3(2)$] *Adv. Geom.* (to appear)

(v) an important class of the representations of A_6 and A_7 has been characterized in [A.A. Ivanov, On Majorana representations of A_6 and A_7] *Comm. Math. Physics.*

(vi) the Majorana representations of $L_2(11)$ is the research project of Sophie Docelle [second year Ph D student at Mathematics Department, Imperial College].

(vii) the Majorana representations of $L_3(3)$ is the research project of Alonso Castillo [first year Ph D student at Mathematics Department, Imperial College].

Certainly larger groups are under consideration including A_8, which so far appeared to be a much harder case.

Refinement of the knowledge of the Monster algebra. Simon Norton from Cambridge possesses an incredible amount of information about the Monster group and its algebra. A very small part of this information is published (usually without any proofs and with justification looked obscure for an outsider). The Majorana theory provides a tool to put this information is a systematic and checkable form. The true success of this project can be seen in correction of some information revealed by Norton. These can be illustrated by the A_5-algebras: one of

them is 20 (rather than 21)-dimensional and the crucial relations in the other one needs certain signs to be alternated. So to say, the Majorana theory brings us beyond Norton's expertise of the Monster.

Classifying subconfigurations. Besides classifying the Majorana representations of specific groups, another promising direction in developing the Majorana theory is to study some specific configurations of the Majorana axes and identification of the subalgebras they generate. The first such problem would be the classification of the subalgebras generated by triples of Majorana axes containing pairs generating $2A$-algebras (there are 36 such configurations in the Monster as given by S. Norton). Michael Aschbacher is particular keen on this direction of developing and hopefully at some stage we could have a close cooperation with him on this project.

Identifying Codes in Graphs: a Special Class of Dominating Sets

Tero Laihonen
Department of Mathematics, University of Turku, Finland

Identifying codes were introduced by Karpovsky, Chakrabarty and Levitin in 1998, and they can be applied, for example, to locating objects in sensor networks. Let a network be modelled by a simple, connected and undirected graph $G = (V, E)$ with vertex set V and edge set E. We can place a sensor in any vertex u. A sensor is able to check its closed neighbourhood $N[u]$ (i.e., the adjacent vertices and itself) and report to a central controller if it detects something wrong there (for example, like a smoke detector). The idea is to place as few sensors as possible in such a way that we can uniquely determine where (that is, in which vertex) the problem occurs (if any) knowing only the set of sensors which gave us the alarm.

Let us denote the subset of vertices, where we placed the sensors, by C. In order to find the sought object (like fire in a building) in our network, we need to choose C in the following way. Denote the set of sensors monitoring a vertex $u \in V$ by $I(u) = N[u] \cap C$. Suppose that C satisfies the following two conditions: (i) $I(u) \neq \emptyset$ for every $u \in V$ and (ii) $I(u) \neq I(v)$ for all $u, v \in V$, $u \neq v$. Hence, $I(x)$ is the set of sensors giving the alarm if there is a problem in x, and since $I(v)$ is unique and nonempty for each $v \in V$, we can determine the vertex with a problem (if there is any). Such a subset $C \subseteq V$ satisfying the two requirements is called an *identifying code*. Obviously, the set C is a dominating set of a graph if the first condition $I(u) \neq \emptyset$ is satisfied for all vertices u. It should be noticed that not all graphs admit an identifying code. Moreover, Slater has introduced a closely related concept of locating-dominating sets, where the second condition is replaced by $I(u) \neq I(v)$ where $u, v \in V \setminus C$. In the seminal paper, the identifying codes were generalized in two ways: 1) an *r-identifying code*: a sensor can check a closed neighourhood within distance r, 2) an *$(r, \leq \ell)$-identifying code*, which can uniquely locate several (up to ℓ) objects in a network.

The original motivation for identifying codes came from finding malfunctioning processors in a multiprocessor system. The most studied underlying graphs include, for instance, square and triangular grids, hexagonal mesh, paths, cycles and binary hypercubes. More general graph theoretic questions have also been investigated over the years.

In this talk, we will consider recent developments in the field. In particular, conjectures concerning paths and cycles will be discussed, as well as optimal density of a 2-identifying code in the infinite hexagonal mesh.

Recognizing and isomorphism testing schurian tournaments in polynomial time

Ilya Ponomarenko
Steklov Institute of Mathematics at St.Petersburg, Russia

A *tournament* is a directed graph in which any two distinct vertices are joined by a unique arc. At present the best algorithm tests the isomorphism of n-vertex tournaments in time $n^{O(\log n)}$, L. Babai-E. Luks (1983). It is based on the two following observations: the automorphism group of a tournament has odd order, and the size of odd order primitive permutation group of degree n is at most n^3.

A standard technique in the graph isomorphism problem is to consider colored graphs. A canonical coloring of vertices and edges can be constructed by the Weisfeiler-Leman algorithm. In fact, its output is just the *coherent configuration* associated with the input graph (this configuration can be regarded as a special partition of a complete directed graph into regular subgraphs). A typical example of a coherent configuration is obtained from a group G acting on a set Ω: in this case the partition of Ω^2 is formed by the orbits of the componentwise action of G on Ω^2. A colored graph is called *schurian*, if the associated coherent configuration is obtained from a permutation group in the above way. For example, a colored tournament T is schurian, if any color class of arcs is the orbit of the group $\mathrm{Aut}(T)$ acting on the arc set of T.

In general, not every coherent configuration can be obtained from a permutation group in the above way (if it was so, then the Weisfeiler-Leman algorithm can be used to test isomorphism of two graphs in a polynomial time). However, a part of permutation group theory can be transfered to coherent configurations. In this way one can find a combinatorial analog for an odd order permutation group. It is much more difficult to find such an analog for the above mentioned upper bound on the size of odd order primitive permutation group. Doing this we come to our main result which can be formulated as follows.

Theorem. Let $\mathcal{T}_n$ be the class of all schurian tournaments on n vertices. Then the following problems can be solved in time $n^{O(1)}$:

(1) given a tournament T on n vertices, test whether $T \in \mathcal{T}_n$,

(2) given a tournament $T \in \mathcal{T}_n$ find the group $\mathrm{Aut}(T)$,

(3) given tournaments $T_1, T_2 \in \mathcal{T}_n$ find the set $\mathrm{Iso}(T_1, T_2)$.

Exponents of labeled digraphs and synchronizing automata

Mikhail Volkov*
Ural Federal University, Ekaterinburg, Russia

A DFA $\mathscr{A} = \langle Q, \Sigma \rangle$ is called *synchronizing* if the action of some word $w \in \Sigma^*$ resets $\mathscr{A}$, that is, leaves the automaton in one particular state no matter at which state in Q it is applied: $q \cdot w = q' \cdot w$ for all $q, q' \in Q$. Any such word w is said to be a *reset word* for the DFA. The minimum length of reset words for $\mathscr{A}$ is called the *reset threshold* of $\mathscr{A}$.

*This talk is based on a joint work of Dmitry Ananichev, Vladimir Gusev and the speaker supported by the Russian Foundation for Basic Research, grant 10-01-00524, and by the Federal Education Agency of Russia, grant 2.1.1/13995.

In 1964 Černý [2] constructed for each $n > 1$ a synchronizing automaton $\mathscr{C}_n$ with n states whose reset length is $(n-1)^2$. Soon after that he conjectured that these automata represent the worst possible case, that is, every synchronizing automaton with n states can be reset by a word of length $(n-1)^2$. This simply looking conjecture resists researchers' efforts for more than 40 years. Even though the conjecture has been confirmed for various restricted classes of synchronizing automata, no upper bound of magnitude $O(n^2)$ for the reset threshold of n-state synchronizing automata is known in general. The best upper bound achieved so far is $\frac{n^3-n}{6}$, see [6].

One of the difficulties that one encounters when approaching the Černý conjecture is that there are only very few *extreme* automata, that is, n-state synchronizing automata with reset threshold $(n-1)^2$. In fact, the Černý series $\mathscr{C}_n$ is the only known infinite series of extreme automata. Besides that, only a few isolated examples of such automata have been found. Moreover, even *slowly* synchronizing automata, that is, automata with reset length close to the Černý bound are very rare. This empirical observation is supported also by probabilistic arguments. For instance, the probability that a composition of $2n$ random self-maps of a set of size n is a constant map tends to 1 as n goes to infinity [5]. In terms of automata, this result means that the reset threshold of a random automaton with n states and at least $2n$ input letters does not exceed $2n$. For further results of the same flavor see [7, 8]. Thus, there is no hope to find new examples of slowly synchronizing automata by a lucky chance or via a random sampling experiment.

We therefore have designed and performed a set of exhaustive search experiments. A brief description of our experiments and some theoretical analysis of their outcome are presented in [1]. One of the main observations reported in [1] was a remarkable similarity between the distribution of reset thresholds of synchronizing automata and the distribution of exponents of primitive digraphs. In particular, we were able to deduce in a uniform way several series of slowly synchronizing automata, both new and already known ones, from some classical series of primitive digraphs with large exponents from [3, 9].

Despite this initial success, it turns out that the notion of exponent is too weak to be useful for isolating synchronizing automata with maximal reset threshold in some important classes, e.g. in the class of Eulerian automata. The reason for this is that we discard too much information when passing from synchronizability to primitivity—we forget anything but length about paths labeled by reset words. Thus, we have tried another approach in which more information is preserved, namely, the Parikh vectors of the paths are taken into account. Let $\mathscr{A} = \langle Q, \Sigma \rangle$ be a DFA with $|\Sigma| = k$ and fix some ordering of the letters in Σ. We define a subset $\mathrm{E}_1(\mathscr{A})$ of $\mathbb{N}_0^k$ as follows: a vector $\mathsf{v} \in \mathbb{N}_0^k$ belongs to $\mathrm{E}_1(\mathscr{A})$ if and only if there is state $r \in Q$ such that for every $p \in Q$, there exists a path from p to r such that v is the Parikh vector of the path's label. If the set $\mathrm{E}_1(\mathscr{A})$ is non-empty, then the automaton $\mathscr{A}$ is called 1-*primitive*. The minimum value of the sum $i_1 + i_2 + \cdots + i_k$ over all k-tuples $(i_1, i_2, \ldots, i_k)$ from $\mathrm{E}_1(\mathscr{A})$ is called the 1-*exponent* of $\mathscr{A}$ and denoted by $\exp_1(\mathscr{A})$. Clearly, every synchronizing automaton $\mathscr{A}$ is 1-primitive and $\exp_1(\mathscr{A})$ serves as a lower bound for the reset threshold of $\mathscr{A}$. One can find some applications of this lower bound in [4].

Here we suggest a further generalization. Let $\mathscr{A} = \langle Q, \Sigma \rangle$ be a DFA with $Q = \{1, 2, \ldots, n\}$ and let k be a non-negative integer. We say that the automaton $\mathscr{A}$ is k-*primitive* if there exist words $u_1, u_2, \ldots, u_n$ such that $1 \cdot u_1 = 2 \cdot u_2 = \cdots = n \cdot u_n$ and every word of length at most k occurs as a factor in each of $u_1, u_2, \ldots, u_n$ the same number of times. Note that the last condition implies that the words $u_1, u_2, \ldots, u_n$ have the same length. The minimal length of words that witness k-primitivity of $\mathscr{A}$ is called the k-*exponent* of $\mathscr{A}$ and is denoted

by $\exp_k(\mathscr{A})$.

Consider now an arbitrary synchronizing automaton $\mathscr{A}$. It is clear that $\mathscr{A}$ is k-primitive for every k and $\exp_k(\mathscr{A})$ serves as a lower bound for the reset threshold of $\mathscr{A}$. Thus, we have the following non-decreasing sequence:

$$\exp_1(\mathscr{A}) \leq \cdots \leq \exp_k(\mathscr{A}) \leq \exp_{k+1}(\mathscr{A}) \leq \ldots . \tag{1}$$

At every next step we require that words $u_1, u_2, \ldots, u_n$ get more similar to each other than they were in previous step. Thus, sooner or later these words "converge" to a reset word and the sequence stabilizes at the reset threshold of $\mathscr{A}$. Our hope is that studying the sequence (1) may shed new light on the Černý conjecture.

References

[1] Ananichev, D.S., Gusev, V.V., Volkov, M.V.: Slowly synchronizing automata and digraphs. In: Hliněný, P., Kučera, A. (eds.), Mathematical Foundations of Computer Science, Lect. Notes Comp. Sci., vol. 6281, pp. 55–65. Springer, Heidelberg (2010)

[2] Černý, J.: Poznámka k homogénnym eksperimentom s konečnými automatami. Matematicko-fyzikalny Časopis Slovensk. Akad. Vied 14(3) 208–216 (1964)

[3] Dulmage, A.L., Mendelsohn, N.S.: Gaps in the exponent set of primitive matrices. Ill. J. Math. 8, 642–656 (1964)

[4] Gusev, V.V.: Lower bounds for the length of reset words in Eulerian automata. In: 5th Workshop on Reachability Problems (accepted)

[5] Higgins, P.M.: The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum 37, 31–36 (1988)

[6] Pin, J.-E.: On two combinatorial problems arising from automata theory. Ann. Discrete Math. 17, 535–548 (1983)

[7] Skvortsov, E. Tipikin, E.: Experimental study of the shortest reset word of random automata. In: Bouchou-Markhoff, B. et al. (eds.), Implementation and Application of Automata. Lect. Notes Comp. Sci., vol. 6807, pp. 290–298. Springer, Heidelberg (2011)

[8] Skvortsov, E., Zaks, Yu.: Synchronizing random automata. Discr. Math. Theor. Comput. Sci. 12(4), 95–108 (2010)

[9] Wielandt, H.: Unzerlegbare, nicht negative Matrizen. Math. Z. 52, 642–648 (1950)

IP-sets defined by words of low complexity

Luca Q. Zamboni
Institut Camille Jordan, Université Lyon 1
Department of Mathematics, FUNDIM, University of Turku

Let $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ denote the set of natural numbers, and $\mathrm{Fin}(\mathbb{N})$ the set of all non-empty finite subsets of $\mathbb{N}$. A subset A of $\mathbb{N}$ is called an IP-set if A contains $\{\sum_{n\in F} x_n | F \in \mathrm{Fin}(\mathbb{N})\}$ for some infinite sequence $x_0 < x_1 < x_2 \cdots$ of natural numbers. An IP-set $A \subseteq \mathbb{N}$ is called an IP*-set if its complement A^c is not an IP-set. In this talk we show how certain families of aperiodic words of low subword complexity can be used to generate a wide assortment of IP-sets or IP*-sets having additional nice properties inherited from the rich combinatorial structure of the underlying word. For example, we will show that in the Fibonacci word W (fixed by the morphism $0 \mapsto 01$, $1 \mapsto 0$) the position of 0s in W is an IP*-set. In contrast, in the infinite word $0W$, both the position of the 0s and the positions of the 1s are IP-sets, and hence neither set is IP*. While we focus primarily on Sturmian and episturmian words, we also consider a broad class of words generated by substitution rules. For instance, by considering partitions of $\mathbb{N}$ defined by words generated by a generalization of the Thue-Morse substitution to an alphabet of size $r \geq 2$, we show that

Theorem 1. *For each pair of positive integers r and N there exists a partition of*

$$\mathbb{N} = A_1 \cup A_2 \cup \cdots \cup A_r$$

such that

- *$A_i - n$ is an IP-set for each $1 \leq i \leq r$ and $1 \leq n \leq N$.*
- *For each $n > N$, exactly one of the sets $\{A_1 - n, A_2 - n, \ldots, A_r - n\}$ is an IP-set.*

By considering partitions defined by words generating minimal subshifts which are topologically weak mixing (for example the subshift generated by the substitution $0 \mapsto 001$ and $1 \mapsto 11001$) we prove that

Theorem 2. *For each positive integer r there exists a partition of $\mathbb{N} = A_1 \cup A_2 \cup \cdots \cup A_r$ such that for each $1 \leq i \leq r$ and $n \geq 0$, the set $A_i - n$ is an IP-set.*

Finally, by considering infinite words on infinite alphabets generated by iterated palindromic closure, we construct infinite partitions of $\mathbb{N}$ such that each element of the partition is an IP-set:

Theorem 3. *There exists an infinite partition*

$$\mathbb{N} = \bigcup_{i=1}^{\infty} A_i$$

such that each of the sets A_i is an IP-set.

Our methods simultaneously exploit the general theory of combinatorics on words, the arithmetic properties of numeration systems defined by substitutions, various notions arising in topological dynamics including proximality and equicontinuity, the spectral theory of symbolic

dynamical systems, and the beautiful and elegant theory, developed primarily by N. Hindman and D. Strauss linking IP-sets to the algebraic/topological properties of the Stone-Čech compactification of $\mathbb{N}$. Using the key notion of p-$\lim_n$, regarded as a mapping from words to words, we are able to apply ideas from combinatorics on words in the framework of ultrafilters. Time permitting, we will discuss a connection between IP-sets and the strong coincidence condition for primitive irreducible substitutions of Pisot type. This talk is based on joint work with M. Bucci and S. Puzynina both from the University of Turku.

Contributed talks

Chromatic uniqueness of elements of height ≤3 in lattices of complete multipartite graphs

Vitaly A. Baransky, Tatiana A. Senchonok
Ural State University, Ekaterinburg

Let G be arbitrary graph. Given a positive integer t, a *t-coloring* of G is a mapping ϕ from the set of vertices V into the set $\{1, 2, \ldots, t\}$ such that $\phi(u) \neq \phi(v)$ if u and v are adjacent in G. The *chromatic number* χ of a graph G is the smallest value of t possible to obtain a t-coloring.

For a natural number x, denote by $P(G, x)$ the number of all possible colorings of the graph G into x given colors. It is well known (see [1]) that the function $P(G, x)$ is a polynomial of degree n in the variable x. This polynomial is called the *chromatic polynomial* of the graph G. Two graphs are called *chromatically equivalent*, or *χ-equivalent*, if they have equal chromatic polynomials.

A graph G is called *chromatically unique*, or *χ-unique*, if it is isomorphic to any graph chromatically equivalent to it. This notion was introduced in [2]. Numerous investigations have been carried out by different authors, in which chromatic equivalence and chromatic uniqueness of graphs have been studied. Much attention was paid to studying the chromatic uniqueness of complete multipartite graphs.

A t-partite graph is a graph whose graph vertices can be partitioned into t disjoint sets so that no two vertices within the same set adjacent. A complete t-partite graph is a t-partite graph such that every pair of graph vertices in the t-sets are adjacent. We denote by $K(n_1, \ldots, n_t)$ the complete n-vertices t-partite graph with partite sets of size $n_1, \ldots, n_t$. Koh and Teo [3] proved that a complete bipartite graph $K(n_1, n_2)$ is chromatically unique for $n_1 \geq n_2 \geq 2$. Li N.Z. and Liu R.Y. [4] proved that a graph $K(1, n_2, \ldots, n_t)$ is χ-unique if and only if $max\{n_2, \ldots, n_t\} \leq 2$.

The main problem here is the following: is any complete multipartite graph $K(n_1, n_2, \ldots, n_t)$ chromatically unique for $t \geq 3$ and $n_1 \geq n_2 \geq \ldots \geq n_t \geq 2$?

A *partition* of a positive integer n is a nonincreasing sequence on nonnegative integers $u = (u_1, u_2, \ldots)$ such that $n = \Sigma_{i=1}^{\infty} u_i$. It is clear that u contains only a finite number $l = l(u)$ of nonempty components. The number l is called the length of a partitionn u. Denote by $NPL(n, t)$ the set of all partitions of a positive integer n of length t. In [5] introduce the relation $\geq$ on the set $NPL(n, t)$ by setting $u = (u_1, u_2, \ldots, u_t) \geq (v_1, v_2, \ldots, v_t) = v$ for any $u, v \in NPL(n, t)$ if $u_1 + u_2 + \ldots + u_i \geq v_1 + v_2 + \ldots + v_i$ for any $i = 1, 2, \ldots, t$. It is proved that $NPL(n, t)$ is a lattice with respect to $\geq$. It is clear that there exists, up to isomorphism, a bijection between complete n-vertex t-partite graphs and elements of the lattice $NPL(n, t)$. Therefore, the order $\leq$ on $NPL(n, t)$ induces the corresponding order on the set of such graphs. We can identify a complete multipartite n-graph with the corresponding partition of n.

Chao and Novacky Jr. [6] showed that complete t-partite n-graphs of the form $K(q + 1, \ldots, q+1, q, \ldots, q)$ are chromatically unique. In the over words, complete multipartite graphs that a minimal elements in lattices $NPL(n, t)$ are chromatically unique. Baransky and Koroleva [7] proved that atoms in lattices $NPL(n, t)$ are chromatically unique too.

The main our result is the following theorem.

Theorem. *Let integers n, t, and h be such that $0 < t < n$ and $h \leq 3$. Then, any complete t-partite graph with nontrivial parts that has height h in the lattice $NPL(n,t)$ is chromatically unique.*

Assume that each graph is assigned a number according to some rule. This number is called a chromatic invariant if it is the same for any two chromatically equivalent graphs.

For proving Theorem we have used invariant $I_2(G)$ — the number of edges of the graph G, invariant $I_3(G)$ — the number of triangles in a graph G and invariant $pt(G, \chi + 1)$ — the number of partitions of the set of vertices of the graph G into $\chi+1$ nonempty subsets consisting of pairwise nonadjacent vertices.

References

[1] Asanov M.O., Baransky V.A., Rasin V.V. Discrete Mathematics: Graphs, Matroids, Algorithms. Izhevsk: Regulyarnaya Khaotich. Dinamika, 2001. P. 362 [in Russian].

[2] Chao C.Y., E.G. Whitehead Jr. On chromatic equivalence of graphs // Theory and Appl. of Graphs. 1978. Vol. 642. P. 121–131.

[3] Koh K.M., Teo K.L. The search for chromatically unique graphs // Graphs Combin. 1990. Vol. 6. P. 259–285.

[4] Li N.Z., Liu R.Y. The chromaticity of the complete t–partite graph $K(1; p_2; ..; p_t)$ // Xinjiang Univ. Natur. Sci. 1990. Vol. 7, 3. P. 95–96.

[5] Baransky V.A., Koroleva T.A. Lattice of partition of natural numbers // Dokl. RAN. 2008. Vol. 418, 4. P. 439–442.

[6] Chao C.Y., G.A. Novacky Jr. On maximally saturated graphs // Discrete Math. 1982. Vol. 41. P. 139–143.

[7] Baransky V.A., Koroleva T.A. Chromatic uniques of atoms in lattices of complete multipartite graphs // Proc. of the Steklov Inst. of Math. 2008. Suppl. 1. P. 15–23.

Some Methods Related to the Černý Conjecture

Mikhail Berlinkov

Ural Federal University, Ekaterinburg, Russia

We consider basic methods related to the Černý conjecture and related open questions.

Suppose $\mathscr{A}$ is a complete deterministic finite automaton whose input alphabet is Σ and whose state set is Q. The automaton $\mathscr{A}$ is called *synchronizing* if there exists a word $w \in \Sigma^*$ whose action *resets* $\mathscr{A}$, that is, w leaves the automaton in one particular state no matter at which state in Q it is applied: $q.w = q'.w$ for all $q, q' \in Q$. Any such word w is called *reset* for the automaton. Below by n we denote the number of states in Q.

In 1964 Černý introduced the notion of synchronizing automata and proved a simple synchronization criterion which can be checked in a quadratic time. Thus the main point here is the minimum length of reset words which we refer to as *reset length*. Černý also constructed for each $n > 1$ an n-state synchronizing automaton whose reset length equals $(n-1)^2$. Soon after that he conjectured that those automata represent the worst possible case, thus formulating the following hypothesis:

Conjecture 1 (Černý)**.** *Each n-state synchronizing automaton has a reset word of length at most $(n-1)^2$.*

By now this simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. The best upper bound known so far is due to Pin [10]: the reset length for each n-state synchronizing automaton is at most $\frac{n^3-n}{6}$. Thus the main open problem in this theory is to prove quadratic (in n) upper bounds on reset length and to find some more precise bounds for important special classes. There are several partial results in this direction. A majority of these results has been proved by using either merge (top-down) or extension (bottom-up) methods. Each of these methods constructs a finite sequence of words $V = (v_1, v_2, \ldots, v_m)$ such that the concatenation $v_1 v_2 \cdots v_m$ of the words in the sequence is a reset word for $\mathscr{A}$.

In methods of the merge type, the words in the sequence V subsequently merge the set Q to some state p, i.e.

$$|Q| > |Q.v_1| > |Q.v_1v_2| > \cdots > |Q.v_1v_2\cdots v_m| = |\{p\}|.$$

In methods of the extension type, the words in V subsequently extend some state p to the set Q, i.e.

$$|\{p\}| < |p.v_m^{-1}| < |p.v_m^{-1}v_{m-1}^{-1}| < \cdots < |p.v_m^{-1}v_{m-1}^{-1}\cdots v_1^{-1}| = |Q|.$$

Merge Methods. A merge method was used to reduce general case to the case of strongly connected automata [8] and to prove the aforementioned upper bound $\frac{n^3-n}{6}$. Some merge methods were also used to prove the Černý conjecture for a few special classes of automata. Let us consider one of them given in [12]. It gives a quadratic upper bound $\frac{n(n-1)}{2}$ for *aperiodic* automata, that is automata without non-trivial cycles induced by some word v. The underlying idea of the proof is to consider the transitive closure $\rho(\mathscr{A})$ of a relation on the states induced by minimal strongly connected component of the square automaton. One can show that the relation $\rho(\mathscr{A})$ is always stable under the action of the words and it becomes a non-trivial partial order for strongly connected aperiodic automata. It allows one to achieve an upper bound of order $n^2/6 + o(n^2)$ for the strongly connected case and thus yields an upper bound $n(n-1)/2$ for the general case of aperiodic automata. However, no matching lower bound is known, and the best lower bound for reset length of aperiodic automata known so far is of order $\Theta(1.5n)$ (see [1]). Our contribution here is that we could reduce the general case of aperiodic automata to the case of aperiodic automata with zero and also we could slightly improve the upper bound in this case to $\frac{e}{4(e-1)}n^2 + o(n^2)$ (unpublished). Thus if one will prove a (linear) upper bound for this case it would imply the same upper bound for all aperiodic automata.

Extension Methods. Extension methods have become popular over the last 15 years and brought a number of impressive achievements. For instance, by using approaches of this sort the Černý conjecture has been proved for circular automata [7] and Eulerian automata [9]. Some quadratic upper bounds were also proved for regular automata [11] and for strongly transitive automata [4, 5]. This line of research has led to a number of natural conjectures

whose validity in the general case would imply the validity of the Černý conjecture or at least some quadratic upper bounds for reset length. In [3] we refuted several conjectures of this kind and suggested some modifications for them. One of these modifications was used to prove a quadratic upper bound for one-cluster automata [2].

References

[1] Ananichev D. *The annulation threshold for partially monotonic automata.* // Russian Mathematics (Iz VUZ). 2010. Vol. 54. No. 1. PP. 1–9.

[2] Béal M., Berlinkov M., Perrin D. *A quadratic upper bound on the size of a synchronizing word in one-cluster automata* // International Journal of Foundations of Computer Science. 2011. V. 22. No. 2. P. 277–288.

[3] Berlinkov M. *On a conjecture by Carpi and D'Alessandro* // In: 14th Internaional Conference "Developments in Language Theory". Lecture Notes in Computer Science. 2010. V. 6224. P. 66–75.

[4] Carpi A., D'Alessandro F. *The synchronization problem for strongly transitive automata* // Lect. Notes Comp. Sci. 2008. V. 5257. P. 240–251.

[5] Carpi A., D'Alessandro F. *Strongly transitive automata and the Černý conjecture* // Acta Informatica. 2009. V. 46. P. 591–607.

[6] Černý J. *Poznámka k homogénnym eksperimentom s konečnými automatami* // Mat.-Fyz. Čas. Slovensk. Akad. Vied. 1964. V. 14. P. 208–216.

[7] Dubuc L. *Sur les automates circulaires eta la conjecture de Černý* // RAIRO Theor. Inform. and Appl. 1998. V. 32. P. 21–34.

[8] Eppstein D. *Reset sequences for monotonic automata* // SIAM J. Comput. 1990. V. 19. P. 500–510.

[9] Kari J. *Synchronizing finite automata on eulerian digraphs* // Theor. Comp. Sci. 2003. V. 295. P. 223–232.

[10] Pin J.-E. *On two combinatorial problems arising from automata theory* // Ann. Discrete Math. 1983. V. 17. P. 535–548.

[11] Rystsov I.K. *Quasioptimal bound for the length of reset words for regular automata* // Acta Cybernetica. 1995. V. 12. P. 145–152.

[12] Trahtman A. *The Černý conjecture for aperiodic automata* // Discrete Math. Theor. Comput. Sci. 2007. V. 9. No. 2. P. 3–10.

Reversible Christoffel factorizations

Michelangelo Bucci
Department of Mathematics, University of Turku
Alessandro De Luca
Department of Mathematics, University of Turku,
Dipartimento di Scienze Fisiche, Università degli Studi di Napoli Federico II Luca Q. Zamboni
Department of Mathematics, University of Turku

Abstract

We define natural decompositions of Sturmian words in Christoffel words, called *reversible Christoffel (RC) factorizations*. They arise from the sequence of Abelian equivalent prefixes of two Sturmian words with the same language. Our main result shows that each RC factorization has 2 or 3 distinct Christoffel words as its terms.

1 Introduction

In combinatorics on words and symbolic dynamics, it is often meaningful to locate the segments where two infinite words w, w' coincide, i.e., find maximal occurrences of factors u such that $w = pus$, $w' = p'us'$ for some p, p', s, s' with $|p| = |p'|$.

If w and w' are two fixed points of an irreducible Pisot substitution, the *strong coincidence conjecture* (proved in [2] in the binary case) implies that w and w' agree on arbitrarily long segments (they are *proximal*); moreover, w and w' have arbitrarily long prefixes which are Abelian equivalent, i.e., an "anagram" of each other. In general, we say that two infinite words w, w' satisfying this last condition are *Abelian comparable*. This induces two factorizations (*comparison*)

$$\begin{aligned} w &= x_1 x_2 \cdots x_n \cdots , \\ w' &= x'_1 x'_2 \cdots x'_n \cdots \end{aligned} \tag{2}$$

defined so that each pair of Abelian equivalent prefixes of w and w' can be written as $(x_1 \cdots x_k, x'_1 \cdots x'_k)$ for some $k \geq 0$.

In this work we look at Sturmian words over $A = \{0, 1\}$ from a similar point of view. Recall that an infinite word over A is *Sturmian* if it has exactly $n + 1$ distinct factors of each length $n \geq 0$. The *language* (of factors) of a Sturmian word is determined by its *slope* (frequency of the letter 1). As is well known, a Sturmian word of slope α encodes rotations by angle $2\pi\alpha$ on a circle. For each irrational slope $\alpha \in]0, 1[$, there is a single *characteristic* Sturmian word c, such that $p \in \mathrm{Pref}(c)$ if and only if $0p, 1p \in \mathrm{Fact}(c)$. Among aperiodic binary words, Sturmian words are characterized by the *balance* property: the number of occurrences of the letter 1 in two factors of the same length may differ at most by 1.

If $0u1$ and $1u0$ are both factors of a Sturmian word, then u is necessarily a palindrome, and $0u1$ (resp. $1u0$) is called a lower (resp. upper) *Christoffel word*; 0 and 1 are also considered to be Christoffel words, both lower and upper. For more general information on Sturmian and Christoffel words, we refer the reader to [3, 4].

Trivially, if two Sturmian words w, w' are Abelian comparable then they have the same slope, and hence the same language. By the balance property, the converse also holds, the only exceptions arising when $\{w, w'\} = \{0c, 1c\}$, where c is characteristic.

Our main result (Theorem 1) shows that in all other cases, each of the factorizations in (2) has at most 3 distinct terms; furthermore, all such terms are Christoffel words. This implies a result previously proved in [5]: if w and w' do not eventually coincide (i.e., if there are no words p, p' such that $|p| = |p'|$, $w = pc$, and $w' = p'c$ for some characteristic c), then they cannot be proximal.

2 Reversible Christoffel factorizations

Let w, w' be two Sturmian words having the same language, and suppose $\{w, w'\} \neq \{0c, 1c\}$ so that w and w' are then Abelian comparable; let (2) be their comparison.

For all $i \geq 1$, x_i and x'_i are Abelian equivalent. By the balance property, it follows either $x_i = x'_i \in A$, or $\{x_i, x'_i\} = \{0u1, 1u0\}$ for some factor u of w. Hence in all cases, x_i and x'_i are Christoffel words, with $x'_i = \widetilde{x_i}$. Thus we can write:

$$\begin{aligned} w &= x_1 x_2 \cdots x_n \cdots , \\ w' &= \widetilde{x_1}\widetilde{x_2} \cdots \widetilde{x_n} \cdots . \end{aligned} \tag{3}$$

Conversely, if $(x_n)_{n>0}$ is a sequence of Christoffel words such that both infinite words in (3) are Sturmian, then comparing w and w' yields exactly the same factorizations.

This motivates the following definition: we call *reversible Christoffel (RC) factorization* of a Sturmian word w any sequence $(x_k)_{k>0}$ of Christoffel words such that

1. $w = x_1 x_2 \cdots x_n \cdots$, and
2. $w' := \widetilde{x_1}\widetilde{x_2} \cdots \widetilde{x_n} \cdots$ is a Sturmian word.

A trivial RC factorization is obtained by choosing all x_k's to be single letters, so that $w' = w$. The definition implies that every choice of w' determines a distinct factorization of w, so that each Sturmian word admits uncountably many distinct RC factorizations. Using the balance property, it is also easy to prove that the terms of an RC factorization are either *all* lower Christoffel words, or all upper.

Let us now state our main theorem.

Theorem 1. *Let w be a Sturmian word, and $w = x_1 x_2 \cdots x_n \cdots$ be an RC factorization of w. The cardinality of the set $X = \{x_n \mid n > 0\}$ is either 2 or 3, and in the latter case, the longest element of X is obtained concatenating the other two.*

The proof relies on the following well known result [6], deeply related to the *three distance theorem* proved in [7] (see also [1]):

Theorem 2 (Three gap theorem)**.** *Let α be an irrational number in $]0, 1[$ and let $\beta \in]0, 1/2[$. The gaps between the successive integers j such that $\alpha j - \lfloor \alpha j \rfloor < \beta$ take either two or three values, one being the sum of the other two.*

Theorem 1 allows to consider RC factorizations as infinite words on the finite alphabet X. We conclude this note with two preliminary results on the structure of such words.

Proposition 3. *Let $w = x_1 x_2 \cdots$ and X be defined as in Theorem 1. If $X = \{u, v, z\}$ with $z = uv$, then substituting each occurrence of z in the chosen RC factorization with $u \cdot v$ produces a new RC factorization of w, which is also a Sturmian word on the alphabet $\{u, v\}$.*

Recall that if u, v are Christoffel words such that uv is Christoffel too and a factor of a Sturmian word w, then exactly one among u^2v and uv^2 is a factor of w. This gives rise to the following "converse" of Proposition 3:

Proposition 4. *Under the hypotheses of Proposition 3, if* $u^2v \in \mathrm{Fact}(w)$ *and* $x_1 \neq v$ *(resp.* $uv^2 \in \mathrm{Fact}(w)$ *and* $x_1 \neq u$*), then each occurrence of* v *(resp.* u*) in the factorization is immediately preceded and followed by* u *(resp.* v*); replacing each occurrence of* $u \cdot v$ *with one of* z *produces a new RC factorization, which is also a Sturmian word over* $\{u, z\}$ *(resp.* $\{v, z\}$*).*

References

[1] P. Alessandri and V. Berthé. Three distance theorems and combinatorics on words. *Enseign. Math. (2)*, 44:103–132, 1998.

[2] M. Barge and B. Diamond. Coincidence for substitutions of Pisot type. *Bull. Soc. Math. France*, 130:619–626, 2002.

[3] J. Berstel and P. Séébold. Sturmian words. In M. Lothaire, editor, *Algebraic Combinatorics on Words*. Cambridge University Press, Cambridge UK, 2002. Chapter 2.

[4] J. Berstel, A. Lauve, C. Reutenauer, and F. Saliola. *Combinatorics on Words: Christoffel Words and Repetition in Words*, volume 27 of *CRM monograph series*. American Mathematical Society, 2008.

[5] M. Bucci, S. A. Puzynina, and L. Q. Zamboni. Ip sets defined by words of low complexity. Preprint, 2011.

[6] N. B. Slater. The distribution of the integers n for which $\{\theta n\} < \phi$. *Proc. Cambridge Philos. Soc.*, 46:525–534, 1950.

[7] V. T. Sós. On the distribution mod 1 of the sequence $n\alpha$. *Ann. Univ. Sci. Budapest, Eötvös Sect. Math*, 1:127–134, 1958.

Using Relevance Queries for Identification of Read-Once Functions

Dmitry V. Chistikov

Faculty of Computational Mathematics and Cybernetics, Moscow State University, Russia

A Boolean function is called *read-once* if it can be expressed by a formula over $\{\wedge, \vee, \neg\}$ where no variable appears more than once. The problem of identifying an unknown read-once function f depending on a known set of variables $x_1, \ldots, x_n$ by making queries is considered. Algorithms are allowed to perform standard *membership queries* (MQ), which reveal the value of f on a given input vector, and queries of two special types.

These latter queries answer the following questions about projections of the target function f: given a projection f_p induced by a partial assignment of variables p,

(RQ) is a variable x_i relevant to f_p?

(ARQ) how many relevant variables does f_p have?

Queries of the first type (denoted RQ for *relevance queries*) take arguments p and x_i, and queries of the second type (denoted ARQ for *aggregated relevance queries*), take only one argument p. As usual, a variable x_i is called *relevant* to a function f if there exist two input vectors α and β disagreeing only on x_i (in ith position) such that $f(\alpha) \neq f(\beta)$.

Our main results are as follows. We develop two algorithms (one for each type of relevance queries) running in polynomial time and identifying an unknown n-variable read-once function. These algorithms can be expressed by deterministic decision trees. The first algorithm makes $O(n^2)$ membership and relevance queries (RQ), and the second algorithm makes $O(n \log^2 n)$ membership and aggregated relevance queries (ARQ).

Since the logarithm of the number of read-once functions of variables $x_1, \ldots, x_n$ is $\Theta(n \log n)$ (see, e. g., [1]), this expression gives a lower bound on the number of bits returned as answers to algorithms' queries in the worst case. Therefore, information-theoretic lower bounds for exact identification complexity (number of queries performed in the worst case) are $\Omega(n \log n)$ (for MQ and RQ) and $\Omega(n)$ (for MQ and ARQ), so the second algorithm is suboptimal only by a factor of $O(\log^2 n)$. If we count the number of bits received from oracles instead of the total number of queries, the second algorithm achieves $O(n \log^3 n)$ and the lower bound is $\Omega(n \log n)$.

The notion of relevance plays a major role in the study of read-once functions. In 1963, Subbotovskaya proved a criterion for determining whether a function is read-once, which involved the concepts of relevant and irrelevant variables [2]. In the context of learning, if the set of all relevant variables is known in advance, the problem of distinguishing an individual read-once function from all other read-once functions can be solved by checking its value on a polynomial number of input vectors [3]. If, however, this information is not available a priori, then the complexity of the problem is exponential: to distinguish $f(x_1, \ldots, x_n) \equiv 0$ from all read-once conjunctions of literals, all 2^n input vectors must be tested.

For exact identification problems, similar results are known. Suppose that, in addition to standard membership queries, algorithms are allowed to ask whether a given projection has at least one irrelevant variable. The problem of exact identification with these queries can be solved polynomially if and only if the set of relevant variables is known a priori [4, 5].

Acknowledgements. This research was supported by Russian Foundation for Basic Research, project number 09-01-00817, and by Russian Presidential grant MD–757.2011.9.

References

[1] P. Savicky, A. R. Woods. The number of Boolean functions computed by formulas of a given size. In: *Random Structures and Algorithms: Proceedings of the Eighth International Conference.* Vol. 13 (1998), 3–4. P. 349–382.

[2] B. A. Subbotovskaya. On comparing bases for implementing Boolean functions with formulae. *Dokl. AN SSSR* (*in Russian*). Vol. 149 (1963), 4. P. 784–787.

[3] A. A. Voronenko. On the length of checking test for repetition-free functions in the basis $\{0, 1, \&, \vee, \neg\}$. *Discrete Mathematics and Applications.* Vol. 15 (2005), 3. P. 313–318.

[4] A. A. Voronenko, D. V. Chistikov. Learning read-once functions using subcube parity queries. *Computational Mathematics and Modeling.* Vol. 22 (2011), 1. P. 81–91.

[5] A. A. Voronenko. On global testing (deciphering) problems for read-once Boolean functions. In: *Proc. of 3rd Russian workshop "Syntax and semantics of logical systems"* (*in Russian*). Irkutsk, izd. VSGAO, 2010. P. 17–22.

On complexity of quadratic permutations

Anna E. Frid
Sobolev Institute, Novosibirsk

We say that two sequences $a = \{a_i\}_{i\in\mathbb{N}}$ and $b = \{b_i\}_{i\in\mathbb{N}}$ of pairwise distinct reals are equivalent if $a_i < a_j$ if and only if $b_i < b_j$ for all i, j. An equivalence class of such sequences is called an *infinite permutation* and denoted by $\alpha = \overline{a} = \overline{b}$ [6]. An infinite permutation can be considered as a combinatorial object somehow analogous to an infinite word: for example, a factor of length n of a permutation is a (finite) permutation $\overline{\{a_i\}_{i=i_0}^{i_0+n-1}}$. We can consider complexity of permutations (defined as the number of distinct factors of a given length) and its variations [6, 2, 8, 9, 11, 12, 13], as well as other combinatorial properties [10, 7].

In this study we consider permutations defined by the sequences of fractional parts $\{\{n^2\alpha\}_{n=0}^{\infty}\}$ for some irrational α. Unlike the complexity of words defined by means of such quadratic sequences of reals [3, 1], the complexity of such permutations grows as $O(n^4)$ not $O(n^3)$. The precise formulas involve sums of the Euler's totient function and are obtained by a geometric method similar to that for words [1, 4, 5, 3].

References

[1] P. Arnoux, C. Mauduit, Complexité de suites engendrées par des recurrences unipotentes, Acta Arithmetica 76 (1996) 85–97.

[2] S. V. Avgustinovich, A. E. Frid, T. Kamae, P. Salimov, Infinite permutations of lowest maximal pattern complexity, Theoretical Computer Science 412 (2011) 2911–2921.

[3] A. Ya. Belov, G. V. Kondakov, Inverse problems of symbolic dimamics, Fundam. Prikl. Mat., 1:1 (1995), 71–79

[4] J. Berstel, M. Pocchiola, A geometric proof of the enumeration formula for Sturmian words, Internat. J. Algebra Comput. 3 (1993), 349–355.

[5] J. Cassaigne, A. Frid, On the arithmetical complexity of Sturmian words, Theoretical Computer Science 380, (2007) 304–316.

[6] D. G. Fon-Der-Flaass, A. E. Frid, On periodicity and low complexity of infinite permutations, European Journal of Combinatorics 28 (2007), 2106–2114.

[7] A. Frid, L. Zamboni, On automatic infinite permutations, accepted.

[8] M. A. Makarov, On permutations generated by infinite binary words, Siberian Electronic Math. Reports 3 (2006), 304–311.

[9] M. A. Makarov, On the permutations generated by Sturmian words, Siberian Mathematical Journal 50:4 (2009), 674–680.

[10] M. A. Makarov, On an infinite permutation similar to the Thue–Morse word, Discrete Mathematics, V. 309, no. 23–24 (2009), P. 6641–6643.

[11] M. Makarov, On the infinite permutation generated by the period doubling word, European Journal of Combinatorics V. 31 (2010) no. 1, 368–378.

[12] A. Valyuzhenich, On permutation complexity of fixed points of uniform binary morphisms, reported at WORDS 2011, submitted.

[13] S. Widmer, Permutation complexity of the Thue-Morse word, Advances in Applied Mathematics 47 (2011) 309–329.

Dynamical systems generating the extremal representations in a negative base

Tomáš Hejda, Zuzana Masáková, Edita Pelantová
Doppler Institute for Mathematical Physics and Applied Mathematics, Department of Mathematics, FNSPE, Czech Technical University in Prague

A positional number system is given by a real base β with $|\beta| > 1$ and by a finite set alphabet $\mathcal{A} \subset \mathbb{R}$. A sequence $\{x_i\}_{i \leqslant k}$ with $x_i \in \mathcal{A}$ is a β-representation of $x \in \mathbb{R}$ if $x = \sum_{i \leqslant k} x_i \beta^i$. For any x to have a β-representation the cardinality of the alphabet must satisfy $\#\mathcal{A} \geqslant |\beta|$.

It is well known that if the base β is a positive integer and the alphabet $\mathcal{A} = \{0, 1, 2, \ldots, \beta - 1\}$ then any positive real x has a β-representation and almost all positive reals (up to a countable number of exceptions) have unique representation. For example, in the decimal numeration system

$$\tfrac{1}{2} = 0.5000000\ldots = 0.4999999\ldots, \qquad \text{whereas} \qquad \tfrac{1}{3} = 0.333333\ldots$$

If the base β is not an integer and the alphabet $\mathcal{A}$ is rich enough to represent all positive reals, then almost all $x \geqslant 0$ have infinitely many representations and one can choose among them "the nicest" one from some point of view.[1]

The study of greedy representations for non-integer bases $\beta > 1$ was initialized by Rényi in 1957, his representations are lexicographically largest ("greedy") in his alphabet. An interest in lexicographically smallest ("lazy") representations for bases $\beta \in (1, 2)$ with Rényi alphabet $\{0, 1\}$ started in 1990 by the work of Erdős, Joó and Komornik. The systematic study of lazy representation for all bases $\beta > 1$ can be found in the work of Dajani and Kraaikamp from 2002.

Recently, in 2009, Ito and Sadahiro introduced numeration system with a negative base $-\beta < -1$ with the alphabet $\mathcal{A} = \{0, 1, 2, \ldots, \lfloor \beta \rfloor\}$. They gave an algorithm for computing a $(-\beta)$-representation of $x \in \left[\frac{-\beta}{1+\beta}, \frac{1}{1+\beta}\right)$ and showed that the natural order on $\mathbb{R}$ correspond to the alternate order on such $(-\beta)$-representations. Using a negative base, we can represent positive and negative numbers without an additional bit for the signum $\pm$.

In this article, we focus on the base $\beta = -\phi$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden mean. We compare several algorithms for obtaining $(-\phi)$ representations, all of which can be seen as dynamical systems on some finite sub-intervals of $\mathbb{R}$. We show that the Ito-Sadahiro algorithm produces neither minimal nor maximal $(-\beta)$-representation with respect to the alternate order and we give algorithms for determination of these extremal strings. Dajani and Kalle shown in 2010 that there exist no transformation generating the extremal representations; our algorithm regularly switches two transformations—one for the even positions and one for the odd ones. We also show that both extremal representations and even the Ito-Sadahiro representation can be obtained using the positive base ϕ^2 and an exotic non-integer alphabet $\mathcal{B} = -\mathcal{A} + \phi\mathcal{A}$.

[1] Note that although most people prefer writing $\frac{1}{2} = 0.5$, the shopkeepers consider the representation $0.4999\ldots$ nicer than $0.5000\ldots$ for $x = \frac{1}{2}$.

On an optimal randomized acceptor for graph nonisomorphism*

Edward A. Hirsch, Dmitry Itsykson
Steklov Institute of Mathematics at St.Petersburg, Russia

While most complexity theorists believe in the $\mathbf{P} \neq \mathbf{NP}$ conjecture, the existence of the fastest algorithm for any problem in $\mathbf{NP} \setminus \mathbf{P}$ is an intriguing open question. Here by "the fastest" we mean the minimum possible running time (compared to other algorithms) for every possible input, up to a polynomial; such algorithm is called *optimal*. Many years ago Levin presented an optimal algorithm [Lev73] for **NP** *search* problems, but it does not translate to *decision* problems, which can be possibly solved more efficiently for some inputs.

Solving a decision problem L can be "split" into two complementary tasks: to give the answer "yes" on the positive instances ($x \in L$) and diverge (do not stop) on the negative ones ($x \notin L$), and to give the answer "no" on the negative instances and diverge on the positive ones. The running time matters on those instances where the algorithm stops. A semi-decision procedure that performs the first task as fast as any other such procedure is called an *optimal acceptor* for L. Optimal acceptors were introduced in [KP89] and studied in connection to p-optimal proof systems (see, e.g., a survey [Hir10]).

To the date, no optimal acceptors are known for languages in $(\mathbf{co\text{-}NP} \cup \mathbf{NP}) \setminus \mathbf{P}$. The same applies to *randomized* acceptors in $(\mathbf{co\text{-}NP} \cup \mathbf{NP}) \setminus \mathbf{BPP}$, i.e., the procedures that can have either one- or two-sided bounded probability of error. The only optimal acceptors for **co-NP**-languages are *heuristic* randomized acceptors, i.e., acceptors with unbounded probability of error for a small fraction of the inputs [HIMS10].

Graph (non)isomorphism and optimality up to permutations of vertices. The problem of graph isomorphism is a natural thoroughly studied problem in **NP**. While there is a fast algorithm for it working for almost all instances according to the "uniform" distribution [BK79], the problem is not known to be in **BPP** and it is still possible that there are probability distributions that make it hard on the average.

We study acceptors for graph nonisomorphism, i.e., algorithms that give (presumably fast) answer for nonisomorphic graphs and do not stop on isomorphic ones (the latter can be, of course, compensated by running a brute-force search algorithm in parallel). Most algorithms for graph (non)isomorphism are invariant-based: that is, they compute functions (invariants) that have the same value for isomorphic graphs and different values for nonisomorphic ones. For nonisomorphic input graphs G_1 and G_2, it usually matters what are their classes under permutations of their vertices and not which members of the classes are chosen (i.e., changing the order of vertices in G_1 hardly helps such an algorithm). This gives a motivation to the following relaxed notion of the optimality. For the pair of graphs (G_1, G_2), call its *cluster* the set of pairs $(\pi_1(G_1), \pi_2(G_2))$ for all possible pairs of permutations (π_1, π_2) of their vertices. An acceptor A is called *optimal up to permutations* if for every algorithm B and every instance (G_1, G_2), the algorithm A runs in time polynomial in the size of the input and the maximum possible running time of the algorithm B on the cluster of (G_1, G_2).

We construct a randomized acceptor that is optimal up to permutations. Namely, we construct an algorithm that is always correct on nonisomorphic graphs, has a bounded probability

*Supported in part by Federal Target Programme "Scientific and scientific-pedagogical personnel of the innovative Russia" 2009-2013, by the grants NSh-5282.2010.1 and MK-4089.2010.1 from the President of RF, by the Programme of Fundamental Research of RAS, and by RFBR grant 11-01-12135-офи-м-2011. The second author is also supported by Rokhlin Fellowship.

of error on isomorphic graphs, and has the best possible median running time on every cluster. Moreover, our acceptor remains optimal if we strengthen the maximum running time on the cluster to the median running time on the cluster or even replace it by the order statistics $\min\{t \mid \Pr[\text{running time} \le t] \ge p\}$ for a constant p. This permits to reformulate the result in terms of the average-case complexity.

Average-case optimality. The basic notions of the average-case complexity were formulated by Levin [Lev73]. For a probability distribution D on the inputs, an algorithm is called average-case polynomial-time if there exists k such that the expectation of the k-th root of the running time is at most linear.

We introduce the notion of an average-case optimal algorithm: this is an algorithm that is as fast on average as any other algorithm for the same problem. The formal definition mimics Levin's definition of an average-case polynomial-time algorithm; in particular, if a problem can be solved in an average-case polynomial time, then the average-case optimal algorithm is an average-case polynomial one. Note that this notion lies in between of the conventional (pointwise) optimality and the worst-case optimality (an algorithm is worst-case optimal if it is polynomial-time if a polynomial-time algorithm for the same problem exists). For graph nonisomorphism, our optimality under permutations is a particular case of the average-case optimality for every distribution D that is stable under permutations of vertices (i.e., $D((G_1, G_2)) = D((\pi_1(G_1), \pi_2(G_2)))$ for every pair of nonisomorphic graphs (G_1, G_2) and permutations of their vertices (π_1, π_2)). Therefore, the acceptor that we constructed is average-case optimal with respect to every such distribution.

Acknowledgements

The first author is grateful to Olaf Beyesdorff, Nicola Galesi, and Massimo Lauria for discussions on the Arthur-Merlin protocol for graph nonisomorphism in Rome in December 2010.

References

[BK79] Laszlo Babai and Ludik Kucera. Canonical labelling of graphs in linear average time. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science (FOCS 1979)*, pages 39–46, 1979.

[HIMS10] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. On optimal heuristic randomized semidecision procedures, with applications to proof complexity and cryptography. Technical Report 10-193, ECCC, 2010. Extended abstract appeared in the proceedings of STACS-2010.

[Hir10] Edward A. Hirsch. Optimal acceptors and optimal proof systems. In Jan Kratochvíl, Angsheng Li, Jirí Fiala, and Petr Kolman, editors, *TAMC*, volume 6108 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2010.

[KP89] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, September 1989.

[Lev73] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973. In Russian. English translation in: B.A.Trakhtenbrot. A Survey of Russian Approaches to Perebor (Brute-force Search) Algorithms. *Annals of the History of Computing* 6(4):384-400, 1984.

On an optimal randomized acceptor for graph nonisomorphism[*†]

Edward A. Hirsch
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences
Dmitry Itsykson
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences
Valeria Nikolaenko
St.Petersburg Academic University, Russian Academy of Sciences
Alexander Smal
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

When we face a computational problem that is not known to be solved in a reasonable (say, polynomial) amount of time, we are still interested to solve it as fast as possible. The existence of an *optimal* algorithm that for *every possible input* returns its answer at least as fast (up to a polynomial) as any other algorithm for the same problem does, is an important structural feature of the problem and the model of computation (deterministic algorithms, bounded-error randomized algorithms, etc.).

While Levin's optimal algorithm for **NP** search problems is known for decades [5], it does not give an optimal algorithm for any decision problem, because, while for **NP**-complete problems the *worst-case* complexity of search and decision are polynomially related, a decision algorithm still can be exponentially faster for some inputs. Also Levin's algorithm does not stop at all on the negative instances. For many interesting languages including the language of Boolean tautologies `TAUT`, the existence of an algorithm that is optimal on the positive instances only (such algorithm is called an *optimal acceptor*) is equivalent to the existence of a p-optimal proof system (that is, a proof system that has the shortest possible proofs, and these proofs can be constructed by a polynomial-time algorithm given proofs in any other proof system) [4, 7, 6] (see [2] for survey).

Optimal heuristic randomized acceptors An obvious obstacle to constructing an optimal algorithm by enumeration is that no efficient procedure is known for enumerating the set of all correct algorithms for, say, `TAUT` or `SAT`. A possible workaround is to check the correctness for a particular input; however, even for `SAT`, a search-to-decision reduction maps the input instance to a *different* instance and thus potentially increases the complexity.

The correctness can be, however, checked in the heuristic setting. A heuristic algorithm for a language L and probability distribution D on the inputs is allowed to make errors for some

[*]The full version of the paper is available as ECCC technical report TR11-091.

[†]Supported in part by Federal Target Programme "Scientific and scientific-pedagogical personnel of the innovative Russia" 2009-2013, by the grants NSh-5282.2010.1 and MK-4089.2010.1 from the President of RF, by the Programme of Fundamental Research of RAS, and by RFBR grant 11-01-00760. The second author is also supported by Rokhlin Fellowship.

inputs; the probability of error according to D must be kept below $\frac{1}{d}$, where d is an integer parameter given to the algorithm. In [3] an optimal heuristic randomized acceptor for every r.e. language L and every polynomial-time samplable D *concentrated on* $\overline{L}$ is constructed. In other words, this is an algorithm that accepts (with bounded probability of error) every $x \in L$ in the fastest possible way, and accepts $x \notin L$ for inputs of total D-probability at most $\frac{1}{d}$.

Our results: derandomization and optimal heuristic algorithms We consider the decision problem for the image of an injective function (under the uniform distribution) that maps n-bit strings to $(n+1)$-bit strings. Its study is motivated, for example, by the fact that a particular case of this problem is the problem of recognizing the image of an injective pseudorandom generator, which has no polynomial-time heuristic randomized algorithm [3, Theorem 5.2]. It is well known that injective pseudorandom generators exist if one-way permutations exist.

For this problem, we extend the previous results in two directions. First, we devise an optimal algorithm, while [3] gave a construction of an optimal acceptor. In [3], the correctness test was performed by repeated sampling inputs in $\overline{L}$ and running a candidate acceptor on them. In our case $\overline{L}$ is the image of an injective function and we can still sample it. However, we still do not have a samplable distribution on L, i.e., on the complement to the image. The check is then done by testing the algorithm on a random input from $\{0,1\}^n$ and computing its overall probability of acceptance.

Our second result is a derandomization of this construction, namely, a deterministic algorithm that is optimal on the average. To do this, we use an expander-based construction of Goldreich and Wigderson [1] of small families of functions with good mixing properties, and also use the input as a source of pseudorandomness. It also derandomizes the construction of [3] of optimal acceptors if we consider it for the same class of problems (i.e., recognizing the complement of the image of an injective function).

References

[1] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[2] Edward A. Hirsch. Optimal acceptors and optimal proof systems. In Jan Kratochvíl, Angsheng Li, Jirí Fiala, and Petr Kolman, editors, *TAMC*, volume 6108 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2010.

[3] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. On optimal heuristic randomized semidecision procedures, with applications to proof complexity and cryptography. Technical Report 10-193, ECCC, 2010. Extended abstract appeared in the proceedings of STACS-2010.

[4] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, September 1989.

[5] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973.

[6] Jochen Messner. On optimal algorithms and optimal proof systems. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 361–372, 1999.

[7] Zenon Sadowski. On an optimal deterministic algorithm for SAT. In *Proceedings of CSL'98*, volume 1584 of *Lecture Notes in Computer Science*, pages 179–187. Springer, 1999.

A Method for Computing the Characteristic Polynomial and Determining Semidefiniteness

Mika Hirvensalo
Department of Mathematics, University of Turku

In this note we point out that to determine whether a given self-adjoint matrix A is positive semidefinite is equivalent to determining whether the characteristic polynomial of A is alternating, and present an algorithm for computing the characteristic polynomial.

The algorithm is essentially division-free, and hence it can be also used to compute the determinant for $n \times n$ matrices over a great variety of commutative rings. Together with multiplications and additions, the method requires only one $\mathbb{Z}$-module division to compute the determinant.

The correctness of the presented method is based on formula analogous to those of Newton-Girard, but the efficiency is due to the dynamic programming method presented in this note.

Minimal uncompletable words

Sandrine Julia
Université de Nice - Sophia Antipolis

Abstract

Given a rational set X, we investigate the set of *uncompletable words* in X^*. These words cannot appear as factor of any word in X^*. Some of them are called *minimal uncompletable* as soon as all their proper factors are factors of some words in X^*. We study the deep structure of the set of minimal uncompletable words and then focus on the finite case. We find a quadratic upper bound for the length of the shortest minimal uncompletable words in terms of the length of the longest words in X.

The problem

Given an alphabet Σ, a subset X of Σ^+ is *complete* if every word w in Σ^* is *completable* in a word in X^* or equivalently, if every w in Σ^* is a factor of some word in X^*. If X is complete, the set $Fact(X^*)$ is said *universal*. If not, the first words responsible of this lack are called the *minimal uncompletable words*.

For a given code, the equivalence between the properties of *maximal code* and of *complete set* is established in [6]. In [3], the authors studied *minimal complete sets* which are not necessarily codes. In [5], the problem about the length of the shortest minimal uncompletable words with respect to a finitely generated set X^* arises. The complexity is conjectured to be quadratic in terms of the length of the longest words in X. Yet, as mentionned in [4]: *The problem of the complexity of determining, given a finite set of finite words X included in Σ^*, whether $Fact(X^*) = \Sigma^*$, is still open. We can also address the question of the shortest word not in $Fact(X^*)$, given that $Fact(X^*)$ differs from Σ^*.* Minimal uncompletable words consist in a particular case of *minimal forbidden words* studied in [2, 1].

Uncompletable words

The set U of *uncompletable words* verifies: $U = \Sigma^* \setminus Fact(X^*)$. A word is minimal in U if all its proper factors belong to $Fact(X^*)$. The subset M containing the *minimal uncompletable words* verifies: $U = \Sigma^*\ M\ \Sigma^*$.
From now on, $\mathcal{A}$ denotes the minimal finite automaton recognizing X^*. We need some specific definitions and their symmetrical, when changing the direction of reading: $S = Suff(X) \setminus X^+$, $P = Pref(X) \setminus X^+$. If the sink state q_Z exists, we set: $Z = \{w \in P\Sigma$ such that w labels a path in $\mathcal{A}$ from the start state to $q_Z\}$; otherwise $Z = \varnothing$.

Proposition 1. *Let X be a rational language. The set M verifies:*

$$M = \left((SX^*Z) \setminus Pref(SX^*) \right) \bigcap \left(\overleftarrow{(S^{\leftarrow}\overleftarrow{X}^*Z^{\leftarrow}) \setminus Pref(S^{\leftarrow}\overleftarrow{X}^*)} \right)$$

Moreover, if at least one of the automata $\mathcal{A}$ and $\mathcal{A}^{\leftarrow}$ has no sink state, the set M and consequently U are empty.

The finite case

If X is finite, the length of its words is bounded by an integer k and then S is finite. Given s in S, we build a deterministic automaton $\mathcal{A}_s$ induced by $\mathcal{A}$ to recognize $Pref(\{s\}X^*)$. Every $\mathcal{A}_s$ is neither complete nor minimal but its $\mathcal{A}$-like part is complete.
If an element w does not belong to $Fact(X^*)$, none of the automata $\mathcal{A}_s$ recognizes it. So, w labels a path from the start state to q_Z in every automaton $\mathcal{A}_s$, in so far s is compatible from the start with w.
The words in Z allow to attract in the state q_Z of the automata $\mathcal{A}_s$ a word outside the set $Fact(X^*)$, for every s in S and once the word s is read.
Before giving a new characterization for complete sets, we need two more notations: for some set E and some integer n, $E^{<n}$ stands for the union $\bigcup_{0\leq i<n} E^n$. We also set $Y = \{y = q^{-1}z /\ q \in Suff(Z)$ and $z \in Z\}$.

Proposition 2. *Let X be a finite language and k be the length of the longest words in X. $Fact(X^*) = \Sigma^*$ if and only if $S\left(Z \cup (X^+Z \cap XZ(SZ \cup Y)^{<2k-1})\right)$ is a subset of $Fact(X^*)$.*

So we get an upper bound on the length of the shortest uncompletable words and consequently, on the shortest minimal uncompletable words.

Theorem 1. *Let X be a finite language such that k is the length of the longest words in X. If the set M is not empty, it contains a word of length at most $4k^2 - 3k + 1$.*

References

[1] M.-P. Béal, M. Crochemore, F. Mignosi, A. Restivo, and M. Sciortino. Computing forbidden words of regular languages. *Fundamenta Informaticae*, 56(1-2):121–135, 2003. Special issue on computing patterns in strings.

[2] M.-P. Béal, F. Mignosi, and A. Restivo. Minimal forbidden words and symbolic dynamics. In *STACS*, pages 555–566, 1996.

[3] J.-M. Boë, A. De Luca, and A. Restivo. Minimal complete sets of words. *Theoretical Computer Science*, 12:325–332, 1980.

[4] N. Rampersad, J. Shallit, and Z. Xu. The computational complexity of universality problems for prefixes, suffixes, factors, and subwords of regular languages. In *Proc.* 7^{th} *Int. Conf. on Words*, Salerno, 2009.

[5] A. Restivo. Some remarks on complete subsets of a free monoid. *Quaderni de La Ricerca Scientifica*, 109:19–25, 1981.

[6] M.-P. Schützenberger and R.-S. Marcus. Full decodable code word sets. *IRE Trans. Inf. Theory, I.T.*, 5:12–15, 1959.

On Maximal Chains of Systems of Word Equations*

Juhani Karhumäki, Aleksi Saarela
Turku Centre for Computer Science TUCS,
Department of Mathematics University of Turku

Theory of word equations is a fundamental part of combinatorics on words. It is a challenging topic of its own which has a number of connections and applications. There have also been several important achievements in the theory over the last few decades.

A fundamental property of word equations is the *Ehrenfeucht compactness property*, proved independently by Albert and Lawrence and by Guba. It guarantees that any system of word equations is equivalent to some of its finite subsystems. In free monoids an equivalent formulation is that each *independent* system is finite, independent meaning that the system is not equivalent to any of its proper subsystems.

As a related problem we define the notion of *decreasing chains* of word equations. This asks how long chains of word equations exist such that the set of solutions always properly diminishes when a new element of the chain is taken into the system. Or more intuitively, how many proper constraints we can define such that each constraint reduces the set of words satisfying these constraints. It is essentially the above compactness property which guarantees that these chains are finite.

The goal of this note is to analyze the above maximal independent systems of equations and maximal decreasing chains of word equations. An essential part is to propose open problems on this area. The most fundamental problem asks whether the maximal independent system of word equations with n unknowns is bounded by some function of n. Amazingly, the same

*Supported by the Academy of Finland under grant 121419

problem is open for three unknown equations, although we do not know larger than three equation systems in this case.

If the number of unknowns is n, then the maximal size of an independent system is denoted by $\mathrm{IS}(n)$. We use two special symbols ub and ∞ for the infinite cases: if there are infinite independent systems, then $\mathrm{IS}(n) = \infty$, and if there are only finite but unboundedly large independent systems, then $\mathrm{IS}(n) = \mathrm{ub}$. Similarly the maximal size of a decreasing chain is denoted by $\mathrm{DC}(n)$.

These definitions work in arbitrary semigroups, but from now on we will consider free monoids and semigroups. The bounds related to free monoids are denoted by IS and DC, and the bounds related to free semigroups, by IS_+ and DC_+. The maximal size of an independent system in a free monoid having a nonperiodic solution is denoted by $\mathrm{IS}'(n)$. Similar notation can be used for free semigroups.

Ehrenfeucht's compactness property means that $\mathrm{DC}(n) \leq \mathrm{ub}$ for every n. No better upper bounds are known, when $n > 2$. Even the seemingly simple question about the size of $\mathrm{IS}'(3)$ is still completely open; the only thing that is known is that $2 \leq \mathrm{IS}'(3) \leq \mathrm{ub}$.

The cases of three and four variables have been studied in an article by Czeizler. The article gives examples showing that $\mathrm{IS}'_+(3) \geq 2$, $\mathrm{DC}_+(3) \geq 6$, $\mathrm{IS}'_+(4) \geq 3$ and $\mathrm{DC}_+(4) \geq 9$. We are able to give better bounds in some cases: $\mathrm{DC}_+(3) \geq 7$ and $\mathrm{DC}(4) \geq 12$.

It was proved by Karhumäki and Plandowski that $\mathrm{IS}(n) = \Omega(n^4)$ and $\mathrm{IS}_+(n) = \Omega(n^3)$. The original bound for $\mathrm{IS}(n)$ is asymptotically $n^4/10000$. By "reusing" some of the unknowns we get a bound that is asymptotically $n^4/1536$.

To summarize, we list a few fundamental open problems.

Question 1: Is IS(3) finite?

Question 2: Is DC(3) finite?

Question 3: Is $\mathrm{IS}(n)$ finite for every n?

Question 4: Is $\mathrm{DC}(n)$ finite for every n?

We know that each of these values is at most ub. If the answer to any of the questions is "yes", a natural further question is: What is an upper bound for this value, or more sharply, what is the best upper bound, that is, the exact value? For the lower bounds the best what is known, according to our knowledge, is the following:

$$1.\ \mathrm{IS}(3) \geq 3, \qquad 2.\ \mathrm{DC}(3) \geq 7, \qquad 3.\ \mathrm{IS}(n) = \Omega(n^4), \qquad 4.\ \mathrm{DC}(n) = \Omega(n^4).$$

A natural sharpening of Question (and) asks whether these values are exponentially bounded. A related question to Question is the following amazing open problem:

Question 5: Does there exist an independent system of three equations with three unknowns having a nonperiodic solution?

As we see it, Question is a really fundamental question on word equations or even on combinatorics on words as a whole. Its intriguity is revealed by Question : we do not know the answer even in the case of three unknowns. This becomes really amazing when we recall that still the best known lower bound is only 3!

To conclude, we have considered equations over word monoids and semigroups. All of the questions can be stated in any semigroup, and the results would be different. For example, in commutative monoids the compactness property holds, but in this case the value of the maximal independent system of equations is ub.

Dynamic proper vertex colorings of a graph*

Dmitry V. Karpov
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

Abstract

Let a subdivision of the complete graph K_n be any graph, which can be constructed from K_n by replacing some edges of K_n by chains of two edges (every such chain adds to a graph a new vertex of degree 2).

Let $d \geq 8$ and G be a connected graph with maximal vertex degree d. We proof that there is a proper dynamic vertex coloring of G with d colors if and only if G is distinct from K_{d+1} and its subdivisions.

1 Introduction

We consider simple finite graphs without loops and multiple edges and their vertex colorings. A vertex coloring of a graph is *proper*, if every two adjacent vertices have different colors.

Let $V(G)$ denote the set of vertices of the graph G, $\Delta(G)$ denote the maximal vertex degree of the graph G. For any vertex $v \in V(G)$ let $d_G(v)$ denote the degree of the vertex v in the graph G and $\mathrm{N}_G(v)$ denote the *neighborhood* of the vertex v, i.e. the set of all vertices of G adjacent to v.

Definition 1. *The vertex coloring of a graph G is dynamic, if for every vertex $v \in V(G)$ with $d_G(v) \geq 2$ its neighborhood $\mathrm{N}_G(v)$ contains vertices of at least two colors.*

According to the classic notion of the *chromatic number* we define the dynamic chromatic number of a graph.

Definition 2. *The dynamic chromatic number $\chi_2(G)$ of a graph G is the least positive integer n such that there is a dynamic proper vertex coloring of G with n colors.*

The classic Brooks Theorem tells us that $\chi(G) \leq \Delta(G)$ for every connected graph G with $\Delta(G) = d \geq 3$ except for the complete graph K_{d+1} on $d+1$ vertices. In [6] (2003) it is proved, that $\chi_2(G) \leq \Delta(G) + 1$ for every graph G with $\Delta(G) \geq 3$.

2 Results

Below we formulate the main results of this paper.

Definition 3. *Let $n \geq 3$. Consider any graph H, constructed from the complete graph K_n by replacing of some edges by chains of 2 edges (each chain adds new vertex of degree 2). We call such graph a subdivision of K_n.*

Let the set $\mathcal{K}_n$ consist of the complete graph K_n and all its subdivisions.

Theorem 1. *Let $d \geq 8$.*

1) If $H \in \mathcal{K}_{d+1}$ then $\chi_2(H) = d + 1$.

2) Let G be a connected graph, $\Delta(G) \leq d$ and let G be not isomorphic to any graph in $\mathcal{K}_{d+1}$. Then $\chi_2(G) \leq d$.

*Supported by RFBR grant 11-01-00760-a

References

[1] R. L. Brooks. *On coloring the nodes of network.* Proc. Cambridge Philos. Soc. 37(1941), p.194–197.

[2] F. Harary. *Graph theory.* Addison-Wesley Publishig Company, Reading, 1969.

[3] H. Hind, M. Molloy, B. Reed. *Colouring a graph frugally.*, Combinatorica 17(4) (1997) 469-482.

[4] B. Reed, *A strengthening of Brooks' theorem.*, J. Combin. Theory Ser. B 76 (1999), no. 2, 136–149.

[5] J. A. Bondy, U. S. R. Murty. *Graph Theory with Applications*, American Elsevier, New York, 1976.

[6] H.-J. Lui, B. Montgomery, H. Poon. *Upper bounds of dynamic chromatic number.* Ars Combinatoria 68 (2003), 193–201.

[7] N. V. Gravin. *Nondegenerate colorings in the Brooks theorem.* Diskret. Mat. 21 (2009), no. 4, p.105–128 (in Russian); English translation in Discrete Math. Appl. 19 (2009), no. 5, p.533–553.

[8] D. V. Karpov. *Dymnamic proper colorings of the graph.* Zap.Nauch.Semin.POMI, 381 (2010), p.47–77. In Russian, English translation to appear in Journal of Math. Sciences.

Complementary Reset Words Problem

Alica Kelemenová

Institute of Computer Science, Silesian University, Opava, Czech Republic

The problem presented in this contribution deals with deterministic finite automata with reset words. These continue studies started in the second half of the last century.

A deterministic finite automaton posses a reset word if this word transforms all states of the automaton to the identical state. I.e. no matter in which state the automaton starts to work with the reset word, it ends in the same state. Well known long time open question for reset automata is to find definite upper bound for the length of the minimal reset words for all n state automata.

In the present contribution we will call the attention in some sense to a complementary task. Given the word w we will look for (a collection of) automata, such that w is their minimal reset word. Following questions arise:
- For given w to characterize all such (strongly connected) automata.
- To find the automaton with minimal number of states among the all reset automata with minimal reset word w.
- To establish the relation between the length of the minimal reset word w and the minimal number of states of reset automata with minimal reset word w.

Let $A = (Q, \Sigma, \delta)$ be a finite automaton with set of states Q, alphabet Σ and deterministic transition function $\delta : Q \times \Sigma \to Q$. Word $w \in \Sigma^*$ is a reset word of automaton $A = (Q, \Sigma, \delta)$ if there is a state q_w such that $\delta(q, w) = q_w$ for all $q \in Q$. Denote by $RDFA(\Sigma)$ the family of all reset automata over Σ, and by $RDFA(\Sigma, w)$ the subfamily of $RDFA(\Sigma)$ with minimal reset

word w. In the contribution we will deal with function which associate with the word w the minimal number of states of automata with minimal reset word w. Formally,

$$q(w) = min\{|Q| : Q \text{ is the set of states of A in RDFA}(\Sigma, \text{w})\}$$

$$k_Q(n) = min\{q(w) : |w| = n\}.$$

In the contribution we present following preliminary results concerning the values of the functions k_Q and q.

Lemma: For every $w \in \Sigma^*$ there is an automaton in $RDFA(\Sigma, w)$ with $|w| + 1$ states. Therefore $q(w) \leq |w| + 1$ for every $w \in \Sigma^*$ and $k_Q(n) \leq n + 1$.

Lemma: $q((10)^i 1) \leq i + 2$ for every $i \geq 1$, $q((10^{i-1})^{i-2}1) \leq i$ for every $i \geq 2$.

Corollary: $k_Q(2i + 1) \leq i + 2$ and $k_Q((i - 1)^2) \leq i$ for every $i \geq 1$.

It holds $k_Q(13) \leq 5$ using $w = (1000)^3 1$ and $k_Q(17) \leq 6$ for $w = 10010001000110001$.

This gives for small numbers n following values

n	=	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20 ...
$k_Q(n) \leq$		3	3	3	4	7	5	9	4	11	7	13	5	15	9	5	6	19	11	21 ...

On Moessner's Theorem

Dexter Kozen
Computer Science Department, Cornell University, USA
Alexandra Silva*
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Consider the following procedure for generating $n \geq 1$ infinite sequences of positive integers. To generate the first sequence, write down the positive integers 1, 2, 3, ..., then cross out every nth element. For the second sequence, compute the prefix sums of the first sequence, ignoring the crossed-out elements, then cross out every $(n - 1)$st element. For the third sequence, compute the prefix sums of the second sequence, then cross out every $(n - 2)$nd element, and so on. For example, for $n = 4$,

1	2	3	~~4~~	5	6	7	~~8~~	9	10	11	~~12~~	13	14	15	~~16~~	17	18	19	~~20~~	21	22	23	~~24~~
1	3	~~6~~		11	17	~~24~~		33	43	~~54~~		67	81	~~96~~		113	131	~~150~~		171	193	~~216~~	
1	~~4~~			15	~~32~~			65	~~108~~			175	~~256~~			369	~~500~~			671	~~864~~		
1				16				81				256				625				1296			

Moessner's theorem says that the final sequence is 1^n, 2^n, 3^n, ...

This construction is an interesting combinatorial curiosity that has attracted much attention over the years. Moessner's theorem was never proved by its eponymous discoverer. The first proof was given later by Perron. Since then, the theorem has been the subject of several popular accounts.

In the construction of Moessner's theorem, the initial step size n is constant. What happens if we increase it in each step? Let us repeat the construction starting with a step size of one and increasing the step size by one each time. Thus, in the first sequence, we cross out 1, 3, 6, 10, ..., $\binom{k+1}{2}$, ...

*Until August 2011: Computer Science Department, Cornell University, Ithaca, NY 14853-7501, USA

```
1   2   ~~3~~   4   5   ~~6~~   7   8   9   ~~10~~   11   12   13   14   ~~15~~   16   17   18   19   20   ~~21~~
    2         6   ~~11~~        18  26  ~~35~~        46   58   71   ~~85~~        101  118  136  155  ~~175~~
              6                 24  ~~50~~           96   154  ~~225~~           326  444  580  ~~735~~
                                24                    120  ~~274~~             600  1044 ~~1624~~
                                                      120                      720  ~~1764~~
                                                                               720
```

Now the final sequence consists of the factorials 1, 2, 6, 24, 120, ... = 1!, 2!, 3!, 4!, 5!, ...

Let us now *increment the increment* by one in each step, thus incrementing the step size by 1, 2, 3, 4, ... in successive steps, crossing out 1, 4, 10, 20, ..., $\binom{k+2}{3}$, ...

```
1   2   3   ~~4~~   5   6   7   8   9   ~~10~~   11   12   13   14   15   16   17   18   19   ~~20~~  ...
    2   ~~5~~       10  16  23  31  ~~40~~       51   63   76   80   95   ...
    2               12  28  51  ~~82~~          133  196  272  352  ...
                    12  40  ~~91~~              224  420  692  ...
                    12  ~~52~~                  276  696  ...
                    12                         288  984  ...
                                               288  1272 ...
                                               288  ...
                                               288  ...
                                               288  ...
```

The final sequence consists of the *superfactorials*

$$1, 2, 12, 288, \ldots = 1!, 2!1!, 3!2!1!, 4!3!2!1!, \ldots = 1!!, 2!!, 3!!, 4!!, \ldots .$$

The generalization of Moessner's theorem that handles these cases is known as *Paasche's theorem*.

Long discovered the following alternative procedure and generalization. Consider the figure illustrating the Moessner construction for $n = 4$ above. Breaking the figure into separate triangles and adding a row of 1's at the top, the first four triangles are

```
1   1   1   1   1        1   1   1   1   1        1   1   1   1   1        1   1   1   1   1
1   2   3   4            5   6   7   8            9   10  11  12           13  14  15  16
1   3   6                11  17  24               33  43  54               67  81  96
1   4                    15  32                   65  108                  175 256
1                        16                       81                       256
```

Call these the *level-n Moessner triangles*. The first triangle is the well-known Pascal triangle. However, note that all the triangles satisfy the *Pascal property*: each interior element is the sum of the elements immediately above it and to its left. Note also that the first column of each triangle consists of the prefix sums of the nth northeast-to-southwest row of the previous triangle. For example, the first column of the third triangle is 1, 9, 33, 65, 81, which are the prefix sums of 1, 8, 24, 32, 16, the last northeast-to-southwest row of the second triangle. Thus, to generate the next triangle in the sequence, let its first column be the prefix sums of the nth northeast-to-southwest row of the previous triangle, let the top horizontal row consist of all 1's, and complete the triangle using the Pascal property.

Long and Salié also generalized Moessner's result to apply to the situation in which the first sequence is not the sequence of successive integers $1, 2, 3, \ldots$ but the arithmetic progression $a, a + d, a + 2d, \ldots$. This corresponds to a sequence of triangles with $d, d, d, \ldots$ along the top and $d, a, a, a, \ldots$ as the first column of the first triangle. They showed that the final sequence obtained by the Moessner construction is $a \cdot 1^{n-1}, (a + d) \cdot 2^{n-1}, (a + 2d) \cdot 3^{n-1}, \ldots$.

Very recently, Hinze, Niqui and Rutten have given proofs involving concepts from functional programming, Hinze using calculational scans and Niqui and Rutten using coalgebra of streams. The proof of Hinze covers Moessner's and Paasche's result whereas Rutten and Niqui only provide a proof of the original Moessner's theorem.

The proof we present has the advantage of covering all the theorems mentioned above and, furthermore, opening the door to new generalizations of Moessner's original result.

Synchronization of automata with one undefined transition

Pavel V. Martyugin
Ural Federal University, Ekaterinburg, Russia

The theory of synchronizing automata is a classical area of research. A deterministic finite automaton is *synchronizing* if there is a word which maps all states of the automaton to one state. The survey of main results about synchronizing automata see in [1]. It is natural to generalize the notion of synchronizing words to the case of automata with a partial transition function (PFA) and to the case of nondeterministic finite automata (NFA).

A partial finite automaton (PFA) is a triple $\mathscr{A} = (Q, \Sigma, \delta)$, where Q is a finite set of states, Σ is a finite alphabet and δ is a partial function from $Q \times \Sigma$ to Q. The function δ can be undefined on some pairs from the set $Q \times \Sigma$. Denote by 2^Q the set of all subsets of the set Q. The function δ can be naturally extended to $2^Q \times \Sigma^*$ by a usual way. A PFA $\mathscr{A} = (Q, \Sigma, \delta)$ is called *carefully synchronizing*, if there is a word $w \in \Sigma^*$ such that the value $\delta(Q, w)$ is defined and $|\delta(Q, w)| = 1$. Clearly, DFA is a partial case of PFA and in this case any c.s.w. is also synchronizing.

The checking whether a given PFA is carefully synchronizing is harder then the checking whether a given DFA is synchronizing. A DFA can be checked in polynomial time, but the checking whether a given PFA is carefully synchronizing is PSPACE-complete (see [4]). The restriction of this problem to the class of 2-letter PFA is also PSPACE-complete.

The length of the shortest synchronizing word for a classical deterministic finite automaton is always polynomial. At the same time the length of the shortest carefully synchronizing word for PFA $\mathscr{A}$ with n states can have length $\Omega(3^{n/3})$ (see [3]). This means that the length of the carefully synchronizing word can be not polynomial in n.

The synchronization of DFA is fast and easy to check, the careful synchronization for an arbitrary PFA is slow and hard to check. We have a problem: where is the border between simplicity and hardness? In this work, we consider PFA with only one undefined transition. Already in this simple case the checking whether a given PFA with only one undefined transition carefully synchronizing is PSPACE-complete. We also prove same facts for PFA with a binary alphabet. By the same way we obtain that the length of the shortest carefully synchronizing word for PFA with one undefined transition can not be bounded by a polynomial in the size of automaton.

We also can consider an undefined transition not as a forbidden transition, but as a transition which maps a state to some unknown state. Thus, it is natural to consider synchronization for nondeterministic automata. A *nondeterministic finite automaton (NFA)* is the triple $\mathscr{A} = (Q, \Sigma, \delta)$ such that Q is a finite set of states, Σ is a finite alphabet, and δ is a function from $Q \times \Sigma$ to 2^Q. The function δ can be naturally extended to the set $2^Q \times \Sigma^*$ as usual. Let $\mathscr{A} = (Q, \Sigma, \delta)$ be an NFA and $w \in \Sigma^*$. The word w is D_1-*directing* if $\delta(q, w) \neq \emptyset$ for all $q \in Q$ and $|\delta(Q, w)| = 1$. The word w is D_2-*directing* if $\delta(q, w) = \delta(Q, w)$ for all $q \in Q$. The word w is D_3-*directing* if $\bigcap_{q \in Q} \delta(q, w) \neq \emptyset$. The D_1, D_2 and D_3-directability is a generalization of the ordinal synchronization then the considered NFA is a DFA. An additional information about the D_1, D_2 and D_3-directability and references can be found in [2].

For NFA, there is the same problem as for PFA: where is a border between the simplicity and the hardness. In this work we consider a class of NFA with a totally defined transition relation such that there exists only one ambiguous transition which map one state to a set of two states. In such NFA $\mathscr{A} = (Q, \Sigma, \delta)$ for any $q \in Q, a \in \Sigma$ we have $|\delta(q, a)| \in \{1, 2\}$ and $|\delta(q, a)| = 2$ for only one pair (q, a).

The class of such NFA is very close to the class of all DFA, but we prove that the problem of checking the D_1 and D_2-directability are PSPACE-complete for NFA with one ambiguous transition even if only 2-letter automata are considered. We also obtain that the length of the shortest D_1 or D_2-directing word for NFA with one ambiguous transition can not be bounded by a polynomial is the size of automaton.

Acknowledgement. The author acknowledges support from the Federal Education Agency of Russia, project 2.1.1/3537, and from the Russian Foundation for Basic Research, grant 09-01-12142.

References

[1] M.V. Volkov, Synchronizing automata and the Černý conjecture, Lect. Notes Comp. Sci. Vol.5196 (2008) 11–27.

[2] M. Ito, Algebraic Theory of Automata and Languages, World Scientific, Singapore, 2004.

[3] P.V. Martyugin, A Lower Bound for the Length of the Shortest Carefully Synchronizing Words, Russian Mathematics (Iz. VUZ), 2010, Vol. 54, No. 1, pp. 46–54.

[4] P. Martyugin, Complexity of Problems Concerning Carefully Synchronizing Words for PFA and Directing Words for NFA, Lect. Notes Comp. Sci. Vol.6072 (2010) pp. 288-302.

A complete one-way function based on finite rank free $(\mathbb{Z} \times \mathbb{Z})$-modules

Sergey I. Nikolenko

Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

In theoretical computer science, complete problems are an invaluable tool for studying complexity classes. A complete problem allows one to reduce all problems in a certain class to a single problem, basically reducing the whole class to one representative problem. In cryptography, on the other hand, complete problems are virtually unknown. For one-way functions, however, the first complete one-way functions were presented by Leonid A. Levin [4, 1]. The first known construction of a (weakly) one-way function, developed by Levin, is the *universal* one-way function that uses a universal Turing machine U to compute the function $f_{\text{uni}}(\text{desc}(M), x) = (\text{desc}(M), M(x))$. In [5], Levin devised a clever trick of completely forbidding indeterministic choice in the computation of the function itself, allowing it only for the inverse function, and formulated the problem of finding other combinatorial complete one-way function. In [2, 3, 7], new combinatorial complete one-way functions were presented, functions based on semi-Thue systems, Post correspondence, and tiling problems.

In this work, we propose a new combinatorial complete one-way function based on the tiling-based Turing machine simulation presented in [6]. In [6], Lohrey and Steinberg showed undecidability of several decision problems related to $(\mathbb{Z}\times\mathbb{Z})$-modules based on simulating Turing machines by tiling. Modifying their construction, we present a complete one-way function based on finite rank free $(\mathbb{Z} \times \mathbb{Z})$-modules.

Our underlying constructions follow [6] and are based on the group $(\mathbb{Z} \times \mathbb{Z})$ and the corresponding Cayley graph. By $\mathcal{E}$ we denote the edge set of the Cayley graph Γ of

the group $(\mathbb{Z} \times \mathbb{Z})$. The set of vertices of Γ is $(\mathbb{Z} \times \mathbb{Z})$, and the set of edges is $\mathcal{E} = \{\{(p,q),(r,s)\} \mid p,q,r,s \in \mathbb{Z}, |u-x|+|v-y|=1\}$. $\mathcal{M}$ is the abelian group of all finitely supported functions from $\mathcal{E} \times C$ to $\mathbb{Z}$.

Proposition 1 (formal generalization of Theorem 4, [6])**.** *Each tiling system T with an undecidable zero tiling problem corresponds to a finitely generated subsemimodule of a certain free $(\mathbb{Z} \times \mathbb{Z})$-module of finite rank with an undecidable membership problem.*

To make use of Proposition 1, Lohrey and Steinberg construct a tiling for which the zero tiling problem encodes the halting problem for a deterministic Turing machine. Therefore, this tiling problem is clearly undecidable for those tilesets for which the halting problem is undecidable.

To get a complete one-way function along the lines of [3, 7], we need to modify the tiling presented in [6, Theorem 7] so that it becomes deterministic in the sense that there is always a single candidate for the next tile to be placed. The properties of this tiling modification are summarized in Theorem 2. The proof from [6] comes through with no change at all, and this tiling also simulates the original Turing machine M. However, now this tiling is deterministic: if we begin with a correct encoding of a Turing machine input then during our simulation, every time we can place a new tile, we have no choice in what tile to place. This makes it possible for the functions F_{tile} and F_{alg} (defined below) to both be polynomial.

Theorem 2. *For every deterministic Turing machine $M = \langle Q, \Gamma, B, \Sigma, \pi, s, H\rangle$ working for at most n^2 steps on inputs of length n there exists a tileset T_M over a set of labels C and a mapping $\varphi_C : \Gamma \to C$ such that:*

1. *for every input x, the zero tiling problem for the tiling set T_M with labels on the bottom row $\varphi_C(x)$ is deterministic;*
2. *there is a correct tiling of the $n^2 \times n^2$ square with labels on the top row $\varphi_C(y)$ if and only if $M(x) = y$.*

Given Theorem 2, we can define a function that polynomially simulates the tiling process (as long as it stays deterministic, which is guaranteed by Theorem 2 for tilesets corresponding to Turing machines) and thus simulates deterministic Turing machines that work in time at most n^2.

Definition 1. *Fix an alphabet $\mathcal{A}$ and an unambiguous encoding of tilesets and tile labels as strings in $\mathcal{A}^*$. The* tiling sum function $F_{\text{tile}} : \mathcal{A}^* \to \mathcal{A}^*$ *acts as follows: for an input string $s \in \mathcal{A}^*$,*

1. *if s is an encoding of $\langle T, x\rangle$ for some tileset T and a string x, $|x| = n$, and $x' \Rightarrow^*_{n^2} y'$ where x' is x padded with $n^2 - n$ symbols "$\square$", then return y, where y is y' stripped from "$\square$" symbols at the end;*
2. *otherwise, return s.*

Theorem 3. *If one-way functions exist then F_{tile} with uniform distribution over its input strings is a weakly one-way function.*

We can now go back to subsemimodules of free $(\mathbb{Z} \times \mathbb{Z})$-modules and formulate the corresponding complete one-way function. A tiling sum $f = \sum_{i=1}^{n} \tau_{x_i,y_i} [\![t_i]\!]$ can be regarded as an element of the subsemimodule of M generated by $\{[\![t]\!] \mid t \in T\}$ (see proof of Proposition 1).

Placing a new tile t at position (x, y) corresponds to adding $\tau_{x,y}[\![t]\!]$ to that sum. Note that even if we have no access to the tiling itself, we can distinguish in polynomial time whether $\tau_{x,y}[\![t]\!]$ corresponds to placing a new tile correctly and on an empty space. Namely, $\tau_{x,y}[\![t]\!]$ is *correct* iff for some $1 \le i \le n$, $e \in \{\{(0,0),(1,0)\},\{(1,0),(1,1)\},\{(1,1),(0,1)\},\{(0,1),(0,0)\}\}$, and $c \in C$ $\tau_{x,y}[\![t]\!](e,c) = -\tau_{x_i,y_i}[\![t_i]\!](e,c)$, where $[\![t_i]\!]$ is a tile that already belongs to f, $\tau_{x_i,y_i}[\![t_i]\!](e,c) \neq 0$, and there is no $j \neq i$ such that $\tau_{x_j,y_j}[\![t_j]\!](e,c) = -\tau_{x_i,y_i}[\![t_i]\!](e,c)$ (i.e., the new tile has canceled at least one preexisting edge that has not been canceled already).

We can now define a polynomial time function on general subsemimodules of free $(\mathbb{Z} \times \mathbb{Z})$-modules that emulates correct tile placement. We define a relation $f \overset{\text{alg}}{\Rightarrow} f'$ for two formal sums f and f' with equal number of nonzero coefficients if there is a deterministic sequence of correct placements of new generators that transforms f into f'. We extend $\overset{\text{alg}}{\Rightarrow}$ to its transitive closure $\overset{\text{alg}}{\Rightarrow}^*$ and partial transitive closure $\overset{\text{alg}}{\Rightarrow}^*_k$.

Definition 2. *Fix an alphabet $\mathcal{A}$ and an unambiguous encoding of elements of $\mathcal{M}$ and their formal sums in $\mathcal{A}^*$. The* algebraic tiling sum function $F_{\text{alg}} : \mathcal{A}^* \to \mathcal{A}^*$ *acts as follows: for an input string $s \in \mathcal{A}^*$,*

1. *if s is an encoding of $\langle \mathcal{N}, f \rangle$ for some set of generators $\mathcal{N} = \{n_i\}$ and a formal sum $f = \sum_{i=1}^{n} \tau_{x_i,y_i} n_i$, and $f \overset{alg}{\Rightarrow}^*_{n^2} f'$ for some other formal sum f' then return f';*
2. *otherwise, return s.*

Theorem 4. *If one-way functions exist then F_{alg} with uniform distribution over its input strings is a weakly one-way function.*

Acknowledgements

The idea of this work resulted from discussions with Prof. Alexei Miasnikov during my visit to Stevens Mathematical Institute in Hoboken, NJ, USA; I thank Prof. Miasnikov, who was my host there, for these discussions and his hospitality. This work was financially supported by the Russian Presidential Grant Programme for Young Ph.D.'s, grant no. MK-4089.2010.1, for Leading Scientific Schools, grant no. NSh-5282.2010.1, and Russian Fund for Basic Research grants.

References

[1] Goldreich, O. *Foundations of Cryptography. Basic Tools.* Cambridge University Press, 2001.

[2] Kojevnikov, A. A., and Nikolenko, S. I. New combinatorial complete one-way functions. In *Proceedings of the 25^{th} Symposium on Theoretical Aspects of Computer Science* (2008), Bordeaux, France, pp. 457–466.

[3] Kojevnikov, A. A., and Nikolenko, S. I. On complete one-way functions. *Problems of Information Transmission 45*, 2 (2009), 108–189.

[4] Levin, L. A. One-way functions and pseudorandom generators. *Combinatorica 7*, 4 (1987), 357–363.

[5] Levin, L. A. The tale of one-way functions. *Problems of Information Transmission 39*, 1 (2003), 92–103.

[6] Lohrey, M., and Steinberg, B. Tilings and submonoids of metabelian groups. *Theory of Computing Systems 48*, 2 (2011), 411–427.

[7] Nikolenko, S. I. *Provably Secure Constructions in Cryptography.* LAP Lamberts Academic Publishing, 2011.

About vertices of degree k of minimally and contraction critically k-connected graphs: upper bounds[†]

Svetlana A. Obraztsova, Alexei V. Pastor
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

In his paper R. Halin (in "Recent Progress in Combinatorics", Academic Press, 1969) discusses, what is the constant c_k such that any minimally and contraction critically k-connected graph has at least $c_k|V(G)|$ vertices of degree k. Twenty years later the exact bound for $k = 4$ ($c_4 = 1$) was found by N. Martinov and, independently, by M. Fontet. For larger k no upper bound is known yet.

We found upper bounds for c_k for $k \geq 5$. Particularly three series of minimally and contraction critically k-connected graphs were constructed. The procedure of construction for $k > 5$ is the following.

Step 1. Take a graph on $\left\lfloor \frac{3(k-2)}{2} \right\rfloor$ vertices with empty set of edges. Using special procedure attach a triangle with vertices of degree k to this graph and repeat this operation $\left\lfloor \frac{k+1}{2} \right\rfloor$ for odd k and $\left\lfloor \frac{k}{2} \right\rfloor$ for even k. Name this graph T_0.

Step 2. Take $\left\lfloor \frac{3(k-2)}{2} \right\rfloor$ copies of graph T_i. For odd k repeat attaching the same triangle as at step 1 k times. For even k the procedure is slightly different. As a result graph T_{i+1} is obtained.

The construction for $k = 5$ is quite similar to the one considered above. Using these series of graphs, it is easy to see, that $c_5 < \frac{17}{22}$, $c_k < \frac{3k^2-2k-9}{6k^2+18k+12}$ for odd $k > 5$ and $c_k < \frac{9k^3-24k^2-28k+12}{18k^3-84k^2+104k}$ for even k.

Formal grammars: reappraising the foundations[*]

Alexander Okhotin
Department of Mathematics, University of Turku

Whenever the syntax of any language has to be rigorously described, one typically employs descriptions of such a form as "a subject followed by a predicate is a sentence" or "expression plus expression is an expression", which define the properties of longer strings on the basis of the properties of shorter strings. Such descriptions can be found, for instance, in 19th century

[†]Supported in part by RFBR grant 11-01-00760-a.
[*]Supported by the Academy of Finland under grant 134860.

grammars of the English language [8], as well as in the early drafts of Algol 60 [7]. This is the intuitively obvious way of representing syntax.

A complete formalization of such syntactic descriptions was first undertaken by Chomsky [2]. Having introduced *formal grammars* at the right time, Chomsky has influenced their subsequent mathematical study by defining the basic tools and notation. According to Chomsky, a grammar is understood as a system for nondeterministic rewriting of strings comprised of symbols of the alphabet and abstract notions defined in the grammar, and this rewriting eventually ends with a well-formed sentence of the language. The intuitively obvious grammars are obtained by using context-free rewriting, in which one abstract symbol is rewritten by a string of symbols at every step. Henceforth, they have been called *context-free grammars*. Other types of rewriting considered by Chomsky led to important models of computation unrelated to syntax (which are equivalent to finite automata, Turing machines, and the NSPACE(n) complexity class), but to no other syntactic formalisms.

Nowadays, the definition of context-free grammars by rewriting is by far the most well-known definition, and it is admittedly nice and clear. However, it does not answer the question of what makes a context-free grammar the intuitively obvious model of syntax. Furthermore, it has misdirected a whole generation of researchers into investigating numerous variants of this rewriting, none of which described anything related to syntax. Something seems to be wrong with the outlook on formal grammars based upon rewriting.

In this talk, the basics of context-free grammars shall be reinvestigated in light of their actual meaning of *an applied logic for representing the syntax*. An alternative equivalent definition shall be presented in terms of a deduction system, manipulating items of the form $[\alpha, w]$, indicating that a string w has the property α according to a grammar. This deduction can also be regarded as a fixpoint iteration in systems of equations with languages as unknowns, as defined by Ginsburg and Rice [3]. Though these might be not very convenient definitions, and though they are obviously equivalent to Chomsky's rewriting, they are useful for providing a proper outlook on the model. In particular, these definitions easily lead to two natural variants of context-free grammars: the *conjunctive grammars* [4, 5], which allow expressing logical conjunction, and *Boolean grammars* [6], in which all Boolean operations may be expressed.

The latter two families of grammars are essentially context-free, in the general meaning of the word, and can be regarded as a *variant of the definition* of the context-free grammars. They share the key properties of the standard context-free grammars, including numerous parsing algorithms. Together, these families of grammars form a hierarchy of applied logics for representing the syntax, in which the standard context-free grammars are the disjunctive fragment. In addition, the concept of context-sensitivity, that is, of grammar rules applicable in a certain context, as investigated by Chomsky [2], can be reinvestigated in terms of these logics, resulting in an inequivalent model [1]. The natural upper bound for such families is the logic proposed by Rounds [9].

It is the author's belief that the slight alteration of the outlook on grammars, proposed in this talk, puts the basic mathematical models of syntax and their variants in a proper perspective. This reappraisal of the foundations shows that the subject of formal grammars is far from being fully researched and understood, and leaves hope for further discoveries.

References

[1] M. Barash, A. Okhotin, "Defining contexts in context-free grammars", manuscript in preparation.

[2] N. Chomsky, *Syntactic Structures*, The Hague, Mouton, 1957.

[3] S. Ginsburg, H. G. Rice, "Two families of languages related to ALGOL", *Journal of the ACM*, 9 (1962), 350–371.

[4] A. Okhotin, "Conjunctive grammars", *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[5] A. Okhotin, "Conjunctive grammars and systems of language equations", *Programming and Computer Software*, 28:5 (2002), 243–249.

[6] A. Okhotin, "Boolean grammars", *Information and Computation*, 194:1 (2004), 19–48.

[7] A. J. Perlis, K. Samelson, "Preliminary report: international algebraic language", *Communications of the ACM* 1:12 (1958).

[8] A. Reed, B. Kellogg, *Higher Lessons in English*, revised edition, 1896.

[9] W. C. Rounds, "LFP: a logic for linguistic descriptions and an analysis of its complexity", *Computational Linguistics*, 14:4 (1988), 1–9.

On some new abelian properties of infinite words*

Svetlana Puzynina
University of Turku, Finland
Sobolev Institute of Mathematics, Novosibirsk, Russia

In combinatorics of words, different abelian properties of words are widely studied nowadays, such as abelian complexity, abelian avoidance, abelian powers and their generalizations. In the talk we consider abelian analogues of the Critical Factorization Theorem (joint work with S. Avgustinovich, J. Karhumäki) and abelian analog of the notion of return word (joint work with L. Zamboni).

One of the main results of combinatorics on words, the Critical Factorization Theorem, relates local periodicities of a word to its global periodicity. It was first proved by Y. Césari and M. Vincent, 1978, and in the present form it is due to J. Duval, 1979. This theorem states, roughly speaking, a connection between local and global periods of a word; the local period at any position of the word is defined as the shortest repetition centered in this position. The theorem says that the global period of a word is the maximum of its local periods. In 1998, F. Mignosi, A. Restivo and S. Salemi proposed a different notion of a local period: a local period at a position is defined as the length of the shortest repetition to the left from this position. In such a definition of local periods squares are not enough to ensure the global periodicity, but the threshold is surprisingly given by the golden ratio φ. Jointly with S. Avgustinovich and J. Karhumäki we study abelian versions of these problems. We seek for constraints for local abelian powers enforcing a word to be (ultimately) periodic. By local abelian powers we mean abelian powers with bounded periods centered at or immediately to the right/left from every position. We investigate both similarities and differences between the abelian powers and the usual powers. The results we obtained show that the constraints for abelian powers implying periodicity should be quite strong, but still natural analogies exist.

*The talk is based on joint results with S. Avgustinovich, J. Karhumäki and L. Zamboni.

Sturmian words can be defined as infinite words having the lowest subword complexity among all aperiodic words. They have been widely studied due to their fundamental importance in different fields of theoretical computer science. Sturmian words have many equivalent characterizations, e. g. using balanced words, cutting sequences, mechanical words, and via morphisms. Jointly with L. Zamboni we develop the approach based on the concept of return words. The notion of a return word is a powerful tool for studying various problems of combinatorics on words, symbolic dynamical systems and number theory. Considering each occurrence of a factor v in an infinite word, the set of return words of v is defined to be the set of all distinct words beginning with an occurrence of v and ending just before the next occurrence of v. This notion was introduced by F. Durand in 1998 and was used for a characterization of primitive substitutive sequences. Sturmian words can be characterized via return words: a word is Sturmian if and only if each of its factors has two returns (L. Vuillon, J. Justin, 2000–2001). We establish a similar characterization of Sturmian words for an abelian analogue of the notion of return word. Namely, we prove that an aperiodic recurrent infinite word is Sturmian if and only if each of its factors has two or three abelian returns.

Word Equations and Linear Algebra[*]

Aleksi Saarela
Turku Centre for Computer Science TUCS,
Department of Mathematics University of Turku

Word equations are a fundamental part of combinatorics on words. One of the basic results in the theory of word equations is that a nontrivial equation causes a defect effect. Not much is known about the additional restrictions caused by several independent relations.

In fact, even the following simple question is still unanswered: how large can an independent system of word equations on three unknowns be? The largest known examples consist of three equations. The only known upper bound comes from the Ehrenfeucht compactness property: an independent system cannot be infinite. Some results concerning independent systems on three unknowns can be found, but the open problem seems to be very difficult to approach with current techniques.

We will use polynomials to study some questions related to systems of word equations. If $\Sigma \subset \mathbb{N}_1$ is an alphabet of numbers and $w = a_0 \dots a_{n-1} \in \Sigma^n$, then we define a polynomial

$$P_w = a_0 + a_1 X^1 + \cdots + a_{n-1} X^{n-1}.$$

Similar polynomials have been used before but the way in which we use them is quite different and allows us to apply linear algebra to the problems.

One of our main contributions is the development of new methods for attacking problems on word equations. Let $\Xi = \{x_1, \dots, x_n\}$ be the set of unknowns. The *length type* of a morphism $h : \Xi^* \to \Sigma^*$ is the vector

$$L = (|h(x_1)|, \dots, |h(x_n)|) \in {}^n.$$

The first result states that if a system has a solution of length type L that has rank r, then the rank of a certain polynomial matrix is at most $n - r$, and if the rank of the matrix is 1 and at most one component of L is zero and the equations are nontrivial, then they have the same solutions of length type L.

When L is not fixed, the exponents of X in the polynomials behave like linear polynomials. These can be analyzed with tools from linear algebra, and we get the second result: If two equations don't have the same sets of solutions of rank $n-1$, then the length types of solutions of the pair of rank $n-1$ are covered by a union of $|E_1|^2$ $(n-1)$-dimensional subspaces, and if $V_1, \dots, V_m$ is a minimal such cover and $L \in V_i$ for some i, then the equations have the same solutions of length type L and rank $n-1$.

Other contributions include simplified proofs and generalizations for old results and studying maximal sizes of independent systems of equations.

As an example of the former, we state that it has been proved that if an independent pair of equations on three unknowns has a nonperiodic solution, then the equations must be balanced. The proof is very long. Using the above results we can give a short proof for a generalization of this result.

Let us finally return to independent systems. We prove that if a system is independent even when considering only solutions of rank $n-1$, then there is an upper bound for the size of the system depending quadratically on the length of the shortest equation. Even though it does not give a fixed bound even in the case of three unknowns, it is a first result of its type.

[*]Supported by the Academy of Finland under grant 121419

We hope that these theorems show that the connection between word equations and linear algebra is not only theoretically interesting, but is also very useful at establishing simple-looking results that have been previously unknown, or that have had only very complicated proofs. In addition to the results of the paper, we believe that the techniques may be useful in further analysis of word equations.

Lower bounds for weakly k-min-wise independent families of permutations

Maxim Vsemirnov
Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

Let $\mathrm{Sym}(n)$ be the set of all permutations on $\{1,\dots,n\}$ and let $G \subseteq \mathrm{Sym}(n)$, $K \subseteq \{1,\dots,n\}$. The set G is called *K-restricted min-wise independent*, if for any $X \subseteq \{1,\dots,n\}$ such that $|X| \in K$ and for any $x \in X$, we have

$$|\{\pi \in G : \min \pi(X) = \pi(x)\}| = \frac{|G|}{|X|}.$$

If $K = \{1,\dots,k\}$ then G is also called *k-restricted min-wise independent.* If $K = \{k\}$ then G is called *weakly k-restricted min-wise independent.*

The case $K = \{1,\dots,k\}$ was studied first by Broder et al. [1], [2], while J. Matoušek and M. Stojaković [5] considered $K = \{k\}$. The original motivation was related to a mathematical model used in web-indexing software, but later other applications , e.g. to derandomisation, were found; see [3].

For $k \geq 3$, polynomial (with resect to n) lower bounds on the size of k-restricted min-wise independent sets were found in [4], [6], [5]. But for weakly k-restricted min-wise independent sets only the bound

$$|G| \geq \log \log n - C.$$

was known. The aim of this talk is to report a significant improvement of that bound. Namely, we prove the following theorem.

Theorem. *Let $k \geq 3$ and $G \subseteq \mathrm{Sym}(n)$ be a weakly k-restricted min-wise independent set. Then*

$$|G| \geq \frac{n-k+2}{k-1}.$$

The proof uses some modification of the methods from [4], [6] based on simple linear algebra.

References

[1] A. Z. Broder. On the resemblance and containment of documents, *Proc. of Compression and Complexity of Sequences*, 1998, 21–29.

[2] A. Z. Broder, M. Charikar, A. M. Frieze and M. Mitzenmacher. Min-wise independent permutations, *J. Comput. System Sci.*, **60** (2000), 630–659.

[3] E. Ya. Dantsin, E. A. Hirsch, S. V. Ivanov and M. A. Vsemirnov. Algorithms for SAT and upper bounds on their complexity. *Zap. Nauchn. Semin. POMI* **277** (2001), 14–46.

[4] T. Itoh, Y. Takei and J. Tarui. On permutations with limited independence. *Proc. of SODA'2000*, 137–146.

[5] J. Matoušek and M. Stojaković. On restricted min-wise independence of permutations. *Random Struct. Algorithms* **23** (2003), 397–408.

[6] S.A.Norin. A polynomial lower bound for the size of a k-min-wise independent set of permutations. *Zap. Nauchn. Semin. POMI* **277** (2001), 104–116.

Short communications

Bounds of a number of leafs of spanning trees in graphs without triangles

Anton V. Bankevich
St.Petersburg State University

We denote the minimum degree of a vertex of G, as usual, by $\delta(G)$. *The girth of a graph* G is denoted by $g(G)$. The number of vertices of G of degree not equal to 2 is denoted by $s(G)$.

For a connected graph G we denote the maximum number of leafs in a spanning tree of G by $u(G)$.

A number of works consider spanning trees in classes of graphs with various additional constraints like ban on certain subgraph. At first Griggs, Kleitman, and Shastri ([3], 1989) proved that $u(G) \geq \frac{v(G)+4}{3}$ in a connected cubic graph with no K_4^- (complete subgraph on four vertices minus one edge). Later Bonsma ([1],2008) demonstrated two interesting bounds for a connected graph with $\delta(G) \geq 3$: $u(G) \geq \frac{v(G)+4}{3}$ for graph without triangles (that is, with $g(G) \geq 4$) and $u(G) \geq \frac{2v(G)+12}{7}$ for the graph with no K_4^-.

It is proved in [8] that $u(G) \geq \frac{v_3+4}{3}$ for a connected graph G with $g(G) \geq 4$ and v_3 vertices of degree at least 3. In [9] for a connected graph G with $\delta(G) \geq 3$, v_3 vertices of degree 3 and v_4 vertices of degree at least 4 the estimate $u(G) \geq \frac{2v_4}{5} + \frac{2v_3}{15}$ is proven.

In [1] the author and D.V.Karpov proved, that $u(G) \geq \frac{1}{4}(s(G) - 2) + 2$. One can assume that $u(G) \geq \frac{g-2}{2g-2}(s(G)-2)+2$ for a graph G with girth at least g. There are series of examples showing that if the conjecture holds for specific values of g, then the bound of the hypothesis for g is precise. The case $g = 4$ is the main result of this work.

Theorem 1. *Let G be a graph without triangles with s vertices of degree different from two. Then $u(G) \geq \frac{1}{3}(s - 2) + 2$.*

But despite a good start, it turned out that our conjecture is false for $g \geq 10$, which will be proved in the following theorem.

Theorem 2. *For any positive integer g there exists an arbitrarily large graph G with girth at least g, for which $u(G) \leq \frac{1}{2}s(G) - \frac{1}{16}s(G)$.*

The question for $5 \leq g \leq 9$ remains open.

References

[1] A. V. Bankevich, D. V. Karpov *Bounds of a number of leafs in spanning trees.* POMI Preprint 2/2011 (in Russian).

[2] J. A. Storer. *Constructing full spanning trees for cubic graphs.* Inform. Process. Lett. 13 (1981), 1, p. 8-11.

[3] J. R. Griggs, D. J. Kleitman, A. Shastri. *Spanning trees with many leaves in cubic graphs.* J. Graph Theory 13 (1989) 6, p. 669-695.

[4] D. J. Kleitman, D. B. West. *Spanning trees with many leaves.* SIAM J. Discrete Math. 4 (1991), 1, p. 99-106.

[5] J. R. Griggs, M. Wu. *Spanning trees in graphs of minimum degree 4 or 5.* Discrete Math. 104 (1992) p. 167–183.

[6] N. Alon. *Transversal numbers of uniform hypergraphs.* Graphs and Combinatorics 6 (1990), p. 1-4.

[7] G. Ding, T. Johnson, P. Seymour *Spanning trees with many leaves.* J. Graph Theory 37 (2001), . 4, p. 189-197.

[8] P. S. Bonsma, F. Zickfeld *Spanning trees with many leaves in graphs without diamonds and blossoms.* LATIN 2008: Theoretical informatics, p. 531-543, Lecture Notes in Comput. Sci., 4957, Springer, Berlin, 2008.

[9] N. V. Gravin. *Constructing spanning tree with many leaves.* Zap. Nauchn. Semin. POMI, v. 381 (2010), p.31-46.

[10] D. V. Karpov. *Spanning tree with many leaves.* Zap. Nauchn. Semin. POMI, v. 381 (2010), p.78-87.

Straight-line Programs: A Practical Test*

Ivan Burmistrov, Lesha Khvorost
Ural State University, Ekaterinburg

Abstract

We present an improvement of Rytter's algorithm that constructs a straight-line program for a given text and show that the improved algorithm is optimal in the worst case with respect to the number of AVL-tree rotations. Also we compare Rytter's and ours algorithms on various data sets and provide a comparative analysis of compression ratio achieved by these algorithms, by LZ77 and by LZW.

Nowadays searching algorithms on huge data sets attract much attention. To reduce the input size one needs algorithms that can work directly with a compressed representation of input data.

Various compressed representations of strings are known: straight-line programs (SLPs) [4], collage-systems [1], string representations using antidictionaries [5], etc. Nowadays text compression based on context-free grammars such as SLPs has become a popular research direction. The reason for this is not only that grammars provide well-structured compression but also that the SLP-based compression is, in a sense, polynomially equivalent to the compression achieved by the Lempel-Ziv algorithm that is widely used in practice. It means that, given a text S, there is a polynomial relation between the size of an SLP that derives S and the size of the dictionary stored by the Lempel-Ziv algorithm, see [4]. It should also be noted that classical LZ78 [8] and LZW [7] algorithms can be considered as special cases of grammar compression. (At the same time other compression algorithms from the Lempel-Ziv family—such as LZ77 and run-length compression—do not fit directly into grammar compression model.)

*The authors acknowledge support from the Ministry for Education and Science of Russia, grant 2.1.1/13995, and from the Russian Foundation for Basic Research, grant 10-01-00524.

Using the fact that SLPs are nicely structured, several researchers keep developing analogues of classical string algorithms that (at least theoretically) perform quite well on SLP-compressed representations: **Pattern matching** [2], **Longest common substring** [3], **Computing all palindromes** [3], some versions of **Longest common subsequence** [6]. At the same time, constants hidden in big-O notation for algorithms on SLPs are often very big. Also the aforementioned polynomial relation between the size of an SLP for a given text and the size of the LZ77-dictionary for the same text does not yet guarantee that SLPs provide good compression ratio in practice. Thus, a major questions is whether or not there exist SLP-based compression models suitable to practical usage? This question splits into two sub-questions addressed in the present paper: How difficult is it to compress data to an SLP-representation? How large compression ratio do SLPs provide as compared to classic algorithms used in practice?

In this paper we present an improved algorithm for SLP construction. The algorithm is similar to Rytter's algorithm proposed in [4] but practical results show that it is more efficient on large inputs.

In the paper we also present practical tests of SLPs as compression model. As a result we should conclude that the existing ways of SLP construction are quite slow. On the other hand, the tests confirm that compression ratio provided by SLPs is close to compression ratio provided by the family of Lempel-Ziv algorithms.

References

[1] T. Kida, T. Matsumoto, Y. Shibata, M. Takeda, A. Shinohara, and S. Arikawa. Collage system: a unifying framework for compressed pattern matching. *Theor. Comput. Sci.*, 1(298):253–272, 2003.

[2] Y. Lifshits. Processing compressed texts: A tractability border. In *CPM*, volume 4580 of *Lecture Notes in Computer Science*, pages 228–240. Springer, 2007.

[3] W. Matsubara, S. Inenaga, A. Ishino, A. Shinohara, T. Nakamura, and K. Hashimoto. Computing longest common substring and all palindromes from compressed strings. In *SOFSEM*, volume 4910 of *Lecture Notes in Computer Science*, pages 364–375. Springer, 2008.

[4] W. Rytter. Application of lempel-ziv factorization to the approximation of grammar-based compression. *Theor. Comput. Sci.*, 302(1-3):211–222, 2003.

[5] Y. Shibata, M. Takeda, A. Shinohara, and S. Arikawa. Pattern matching in text compressed by using antidictionaries. In *CPM*, volume 1645 of *Lecture Notes in Computer Science*, pages 37–49. Springer, 1999.

[6] A. Tiskin. Faster subsequence recognition in compressed strings. *CoRR*, abs/0707.3407, 2007.

[7] T. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.

[8] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.

Generalized flowers in k-connected graphs. Application to the case $k = 4$

Alexander L. Glazman
Steklov Institute of Mathematics at St.Petersburg

1 Introduction

For every graph G, we denote the set of its vertices by $V(G)$ and the set of its edges by $E(G)$. As usual, for any set $F \subseteq E(G)$ we denote the graph with the set of vertices $V(G)$ and the set of edges $E \setminus F$ by $G - F$. For any set $U \subset V$ of vertices, let $G - U$ be the induced subgraph of the graph G with the set of vertices $V \setminus U$.

Definition 1. *A set R of vertices of a graph G is a cutset if the graph $G - R$ is disconnected. If necessary to indicate the number of elements of a cutset, a cut consisting of x elements is called an x-cutset.*

A cut R splits a set $X \subset V(G)$ of vertices if the vertices of $X \setminus R$ are disconnected in the graph $G - R$.

The decomposition of a connected graph by its cut-vertices is well-known. It's convenient to use a tree of blocks and cut-vertices to describe this structure. W.T.Tutte [1] described the structure of 2-vertex cutsets in biconnected graphs. It happened to have much common with the structure of a connected graph, separated by cut-vertices. A construction of a tree of blocks for biconnected graphs was also proposed in [1].

We study k-connected graphs. Every cutset in k-connected graph has at least k vertices.

Definition 2. *1) We say that cuts R and T are independent if R does not split T and T does not split R. Otherwise the cuts R and T are dependent.*

2) Let $\mathfrak{S} \subset \mathfrak{R}_k(G)$. The dependence graph $\mathrm{Dep}(\mathfrak{S})$ *is a graph with vertex set $\mathfrak{S}$ and two vertices are adjacent iff corresponding sets are dependent.*

It is proved in [4, 2], that if R and T are k-cutsets in k-connected graph G such that R does not split T, then T does not split R, i. e., the cuts R and T are independent. The main obstacle to construct something like the tree of blocks and cut-vertices is dependency of k-cutsets.

D. Karpov [5] invented a new method for studying a structure of k-connected graphs. The power of this method is well-illustrated in case $k = 2$ — in fact the structure is the same as in W.T.Tutte's work. After this D.Karpov with A.Pastor [4] described the structure of 3-connected graph using the new invented method. All cutsets are divided in some groups which are called *complexes*, and due to this definition it is possible to construct a hypertree on these complexes.

Definition 3. *A part of the decomposition of G by cutset S is a subgraph of G, induced on a maximal (by inclusion) set of vertices which is not split by S. The set of all such parts we will denote by* $\mathrm{Part}(S)$.

A part of the decomposition of G by a set of cutsets $\mathfrak{S}$ is a subgraph of G, induced on a maximal set of vertices which is not split by any cutset from $\mathfrak{S}$. The set of all such parts we will denote by $\mathrm{Part}(\mathfrak{S})$.

The boundary of a part $A \in \mathrm{Part}(\mathfrak{S})$ is the set of all vertices of A which lie in any set from $\mathfrak{S}$. We will denote it by $\mathrm{Bound}(A)$.

2 Definition of a flower

The most important construction in the method of [5] is a *flower*. It can be drawn as several equal petals lying on a circle with several vertices in the center of this circle, two nonneighboring petals together with the center forms a cutset. But the strict definition of a flower is much more complicated.

Let $m \geq 4$, and $P, Q_1, \ldots, Q_m \in V(G)$ satisfy the following conditions for all $i \in \{1, \ldots, m\}$:

$$0 \leq |P| < k, \qquad Q_i \cap P = \varnothing, \qquad |Q_i| = \frac{k - |P|}{2}.$$

Let's consider $F = (P; Q_1, \ldots, Q_m)$. Sets $Q_1, \ldots, Q_m$ (we name them *petals*) are cyclic ordered, i.e. the cyclic shift of them doesn't change F. Let's consider $Q_{i,j} = Q_i \cup Q_j \cup P$. We say that petals Q_i and Q_j are *close*, if for all k from i to j we have $Q_k \subset Q_{i,j}$ or for all k from j to i we have $Q_k \subset Q_{i,j}$.

Definition 4. *Assume that exist such $\mathfrak{S} \subset \mathfrak{R}_k(G)$ consisting of sets $Q_{i,j}$ (where i and j are not neighbors), that decomposition $\mathrm{Part}(\mathfrak{S}) = \{G_{1,2}, G_{2,3}, \ldots, G_{m,1}\}$, and $\mathrm{Bound}(G_{i,i+1}) = Q_{i,i+1}$ for every i.*

Besides assume that if $Q_i \cap Q_j \neq \varnothing$ for some i, j then the petals Q_i and Q_j are close.

Then we say that F is a flower. The set P is the center and sets $Q_1, \ldots, Q_m$ are petals of this flower.

Definition 5. *The decomposition of G by flower F is $\mathrm{Part}(F) = \{G_{1,2}, \ldots, G_{m,1}\}$, subgraphs $G_{i,i+1}$ are parts of this decomposition. If no two petals of F intersect then the flower F is regular. We say that $\mathfrak{S}$ generates a flower F.*

It turned out that for studying k-connected graphs, where $k > 3$, we have to make a generalization. At first, it is convenient to generalize the notion of part.

Definition 6. *Let $T \in \mathfrak{R}_k(G)$, $\mathrm{Part}(T) = \{A_1, \ldots, A_m\}$, $\ell \geq 2$. Let $I_1 \cup \cdots \cup I_\ell = \{1, \ldots, m\}$ be a disjunctive union and $B_1, \ldots, B_\ell$ be induced subgraphs of the graph G with $V(B_j) = \cup_{i \in I_j} V(A_i)$. Then $\mathrm{Part}_I(T) = \{B_1, \ldots, B_k\}$ is a generalized decomposition of G by cutset S.*

When generalized parts are defined it is easy to define the notion of generalized flower – the definition is the same but instead of parts we consider generalized parts.

3 Results

Lemma 1. *Let $\mathfrak{S} \subset \mathfrak{R}_k(G)$ generate a generalized flower. Then the dependence graph $D(\mathfrak{S})$ is connected.*

Definition 7. *Let $\mathfrak{R}(F)$ be the set of inner sets of flower, i.e. sets $Q_{i,j}$, where petals Q_i and Q_j are not close. Bounds of a flower are sets $Q_{i,i+1}$.*

Theorem 1. *Let $F = (P; Q_1, \ldots, Q_m)$ be a generalized flower. Then $\mathfrak{R}(F) \subset \mathfrak{R}_k(G)$, and $Q_{i,j} \in \mathfrak{R}(F)$ separates $G_{i,j}$ from $G_{j,i}$.*

Theorem 2. *Let sets $\mathfrak{S}, \mathfrak{T} \in \mathfrak{R}_k(G)$ generate generalized flowers F_S and F_T respectively with the same center and set of petals. Then $\mathrm{Part}(\mathfrak{S}) = \mathrm{Part}(\mathfrak{T})$ and $F_S = F_T$ (i.e. cyclic order of petals in these two flowers is the same).*

Definition 8. *There are two types of flowers in a 4-connected graph — with an empty center and with a 2-vertex center. Those with an empty center we will call 0-flowers and others are called 2-flowers.*

Theorem 3. *Every 4-cutset lying in a set of vertices of a 2-flower F contains the center of F, i.e. is either inner set or boundary of F.*

Definition 9. *1) Let us call a petal Q_i of 0-flower $F = (P; Q_1, \dots, Q_m)$ a switch if $Q_i = \{x, y\}$, $Q_i \subset Q_{i-1} \cup Q_{i+1}$.*

If Q_i is a swich of 0-flower F then a switching is a replacement of Q_i by $Q'_i = Q_{i-1} \cup Q_{i+1} \setminus Q_i$.

2) Two 0-flowers are similar if one can be obtained from the other by several operations of switching.

3) Quasiinner sets of a 0-flower are inner sets of flowers similar to it.

It is not very difficult to show that similarity of 0-flowers is an equivalence.

Theorem 4. *Every 4-cutset T lying in a set of vertices of a 0-flower F, contains the center of F. Moreover, T is an inner set or quasiinner set, or a boundary of F.*

References

[1] W. T. Tutte. *Connectivity in graphs.* Toronto, Univ. Toronto Press, 1966.

[2] W. Hohberg. *The decomposition of graphs into k-connected components.* Discr. Math., **109**, 1992, p. 133-145.

[3] D. V. Karpov, A. V. Pastor. *On the structure of k-connected graph.* Zap. Nauchn. Semin. POMI, **266**, 2000, p. 76-106. In Russian. English translation in Journal of Math. Sci. **113**, i.4 (2003), p. 584-597.

[4] D. V. Karpov, A. V. Pastor *The structure of a 3-connected graph.* POMI Preprint 19/2008, in Russian.

[5] D. V. Karpov *Cutsets in a k-connected graph.* Zap. Nauchn. Semin. POMI, v. 340 (2006), p. 33-60, in Russian. English translation in Journal of Math. Sci. **145**, i.3 (2007), p. 4953–4966.

Observations and Problems on k-abelian avoidability*

Mari Huova, Juhani Karhumaki

Department of Mathematics and TUCS, University of Turku, Finland

Theory of avoidability is among the oldest and the most studied topics in combinatorics on words. The first results in this area are the well-known results by Axel Thue, [Th1, Th2], showing the existence of an infinite cube-free binary word and an infinite square-free ternary word. Since late 1960's commutative, i.e. abelian, variants of the above problems were studied. For example, Evdokimov [Ev], Pleasant [Pl] and finally Keranen [Ke] with the optimal result

*Supported by the Academy of Finland under the grant 121419 and by the Vaisala Foundation.

showed, respectively, that there exists an infinite word over a 25-, 5- and 4-letter alphabet avoiding abelian squares. The optimal value three for the size of the alphabet avoiding abelian cubes was proved by Dekking [De].

We introduce new variants of the problems by defining repetitions via new equivalence relations which lie properly in between equality and commutative equality, i.e. abelian equality.

Let $k \geq 1$ be a natural number. We say that words u and v in Σ^+ are *k-abelian equivalent*, in symbols $u \equiv_{a,k} v$, if

1. $\mathrm{pref}_{k-1}(u) = \mathrm{pref}_{k-1}(v)$ and $\mathrm{suf}_{k-1}(u) = \mathrm{suf}_{k-1}(v)$, and
2. for all $w \in \Sigma^k$, the number of occurrences of w in u and v coincide.

Here pref_{k-1} (resp. suf_{k-1}) denotes the prefix (resp. suffix) of length $k-1$.

Now, notions like *k-abelian repetitions* are naturally defined. For instance, $w = uv$ is a *k-abelian square* if and only if $u \equiv_{a,k} v$. In the binary case 2- and 3-abelian words are fairly easy to characterize which allows us to estimate the sizes of the corresponding equivalence classes. They are of order $\Theta(n^2)$ and $\Theta(n^4)$, see [HKSS].

The essential goal of this presentation is to point out that the natural variants of the Thue's problems asking the sizes of the smallest alphabets avoiding k-abelian squares and cubes are not trivial, even in the case $k = 2$. The following table 1 summarizes the results we mentioned at the beginning and at the same time tells the limits of our problems.

Avoidability of squares				Avoidability of cubes			
	type of rep.				type of rep.		
size of the alph.	$=$	$\equiv_{a,2}$	$\equiv_a$	size of the alph.	$=$	$\equiv_{a,2}$	$\equiv_a$
2	$-$	$-$	$-$	2	$+$	?	$-$
3	$+$	?	$-$	3	$+$	$+$	$+$
4	$+$	$+$	$+$				

Table 1: Avoidability of different types of repetitions in infinite words.

We were able to settle the first one of the question marks in table 1 by computer checking and the result was that the longest ternary word which is 2-abelian square-free has length 537 showing that there does not exist an infinite 2-abelian square-free word over any ternary alphabet. To solve the other question mark we also did some computer checking - and obtained evidence that the answer is likely to be different compared to the first one. For example, we were able to construct a binary word of more than 100000 letters that still avoids 2-abelian cubes. This shows that there exist, at least, very long binary 2-abelian cube-free words. We counted also the number of 2-abelian square-free words with respect to their lengths and compared these to the sizes of different sets of 2-abelian cube-free words. To mention an example we have that the maximal number of ternary 2-abelian square-free words with a fixed length (105) is 404286 and the number of binary 2-abelian cube-free words of length 60 is already 478456030. In addition, in many cases the number of binary 2-abelian cube-free words grows exponentially with a factor approximately 1,3 with respect to the length of words.

As a conclusion, our two considered problems would seem to behave differently: one like words and the other like abelian words.

References

[De] F. M. Dekking: *Strongly non-repetitive sequences and progression-free sets*. J. Combin. Theory Ser. A 27(2), 181-185 (1979).

[Ev] A. A. Evdokimov: *Strongly asymmetric sequences generated by a finite number of symbols*. Dokl. Akad. Nauk SSSR 179, 1268-1271 (1968); English translation in Soviet Math. Dokl. 9, 536-539 (1968).

[HKSS] M. Huova, J. Karhumaki, A. Saarela, K. Saari: *Local squares, periodicity and finite automata*. In: C. Calude, G. Rozenberg, A. Salomaa (eds.) Rainbow of Computer Science, 90-101, Springer, 2011.

[Ke] V. Keranen: *Abelian squares are avoidable on 4 letters*. In: W. Kuich (ed.) ICALP 1992. LNCS, vol. 623, 41-52. Springer, Heidelberg, 1992.

[Pl] P. A. B. Pleasant: *Non-repetitive sequences*. Proc. Cambridge Philos. Soc. 68, 267-274 (1970).

[Th1] A. Thue: *Uber unendliche Zeichenreihen*. Norske vid. Selsk. Skr. Mat. Nat. Kl. 7, 1-22 (1906).

[Th2] A. Thue: *Uber die gegenseitige Lage gleicher Teile gewisser Zeichenreihen*. Norske vid. Selsk. Skr. Mat. Nat. Kl. 1, 1-67 (1912).

Lower bounds for myopic DPLL algorithms with a cut heuristic

Dmitry Itsykson*, Dmitry Sokolov†

Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences

We introduce a lower bounds on the time complexity of DPLL algorithms that solve the satisfiability problem using a splitting strategy. We prove the theorem about effectiveness vs. correctness trade-off for deterministic myopic DPLL algorithms with cut heuristic.

DPLL (are named by the authors: Davis, Putnam, Logemann and Loveland) algorithms are one of the most popular approach to the problem of satisfiability of Boolean formulas (SAT). DPLL algorithm is a recursive algorithm that takes the input formula ϕ, uses a procedure $\mathbf{A}$ to choose a variable x, uses a procedure $\mathbf{B}$ that chooses the value $a \in \{0, 1\}$ for the variable x that would be investigated first, and makes two recursive calls on inputs $\phi[x := a]$ the $\phi[x := 1 - a]$. Note that the second call is not necessary if the first one returns the result, that the formula is satisfiable.

There is a number of works concerning lower bounds for DPLL algorithms: for unsatisfiable formulas exponential lower bounds follow from lower bounds on the complexity of resolution proofs [1], [2]. In case of satisfiable formulas we have no hope to prove superpolynomial lower bound since if $\mathbf{P} = \mathbf{NP}$, then procedure $\mathbf{B}$ may always choose the correct value of the variable according to some satisfying assignment. The paper [3] gives exponential lower bounds on

*Partially supported by Federal Target Programme "Scientific and scientific-pedagogical personnel of the innovative Russia" 2009-2013, RAS Program for Fundamental Research, the president grants NSh-5282.2010.1 and MK-4089.2010.1 and by RFBR.

†Partially supported by CS Club Scholarship.

satisfiable formulas for two wide enough classes of DPLL algorithms: myopic and drunken algorithms. In the myopic case procedures **A** and **B** can see formula with erased signs of negation, they can request the number of positive and negative occurrences for every variable and also may read $K = n^{1-\epsilon}$ clauses precisely.

All lower bounds for satisfiable instances are based on the fact that during several first steps algorithm falls into a hard unsatisfiale formula, and algorithm should investigate the whole it' splitting tree. In this work we extend the class of DPLL algorithms by adding the procedure **C** that may decide that some branch of the splitting tree will not be investigated since it is not too "perspective". More precisely, before each recursive call an algorithm calls the procedure **C** that decides whether to make this recursive call or not. DPLL algorithms with cut heuristic are always give a correct answer on unsatisfiable formulas; however they may err on satisfiable formulas. On the other hand if the presence of a cut heuristic gives the substantial improvement on the time complexity while the bad instances (i.e. instances on which the algorithm errs) are not easy to find, then such algorithms become reasonable.

In this work we show that it is possible to construct the family of unsatisfiable formulas $\Phi^{(n)}$ in polynomial time such that for every myopic deterministic heuristics **A** and **C** there exists a polynomial time samplable ensemble of distributions R_n such that the DPLL algorithm based on procedures $\mathbf{A}, \mathbf{B}$ and **C** for some **B** either errs on 99% of random inputs according R_n or runs exponential time on formulas $\Phi^{(n)}$. In case **A** and **C** are not restricted we show that a statement similar to above is equivalent to $\mathbf{P} \neq \mathbf{NP}$. The case of randomized myopic procedures **A** and **C** is left open.

Heuristic acceptors. The study of DPLL algorithms with cut heuristic was also motivated by the study of heuristic acceptors [4] The distributional proving problem is a pair (L, D) of a language L and a polynomial time samplable distribution D concentrated on the complement of D. An algorithm A is called a heuristic acceptor if it has additional input d that represents the parameter of the error and for every $x \in L$ and $d \in \mathbb{N}$, $A(x, d)$ returns 1 and $\Pr_{x \leftarrow D_n}[A(x) = 1] < 1/d$ for every integer n. We call an acceptor polynomially bounded if for every $x \in L$ running time of $A(x, d)$ is bounded by polynomial in $|x| \cdot d$. The paper [4] shows that the existence of distributed proving problems that have no polynomially bounded acceptors is equivalent to the existence of infinitely often one-way functions.

Let D be some distribution concentrated on satisfiable formulas. We consider DPLL algorithm with a cut heuristic supplied with an additional parameter d that is available for procedures $\mathbf{A}, \mathbf{B}, \mathbf{C}$. We call such an algorithm a heuristic DPLL acceptor if it satisfies the definition of a heuristic acceptor. Our result implies that there are no deterministic polynomially bounded myopic DPLL acceptors for the proving problem $(UNSAT, Q)$.

References

[1] A. Urquhart. Hard examples for resolution. *JACM*, 34(1):209–219, 1987.

[2] G. S. Tseitin. On the complexity of derivation in the propositional calculus. *Zapiski nauchnykh seminarov LOMI*, 8:234–259, 1968. English translation of this volume: Consultants Bureau, N.Y., 1970, pp. 115–125.

[3] Michael Alekhnovich, Edward A. Hirsch, and Dmitry Itsykson. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. *J. Autom. Reason.*, 35(1-3):51–72, 2005.

[4] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. On optimal heuristic randomized semidecision procedures, with applications to proof complexity and cryptography. Technical Report 10-193, ECCC, 2010. Extended abstract appeared in the proceedings of STACS-2010.

Reset complexity of ideal languages

Marina I. Maslennikova
Ural Federal University, Ekaterinburg, Russia

We present a new characteristic of a regular ideal language called *reset complexity*. We find some bounds on the reset complexity in terms of the state complexity of a given language. We also compare the reset complexity and the state complexity for languages related to slowly synchronizing automata and study the uniqueness question for automata yielding the minimum of reset complexity.

A DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is called *synchronizing* if there exists a word $w \in \Sigma^*$ which leaves the automaton in one particular state no matter which state in Q it starts at: $\delta(q, w) = \delta(q', w)$ for all $q, q' \in Q$. Any such word is said to be *synchronizing* (or *reset*) for the DFA $\mathscr{A}$. By $Syn(\mathscr{A})$ we denote the language of all words synchronizing $\mathscr{A}$.

Synchronizing automata are of interest, motivated mostly by the Černý conjecture. Černý [2] produced for each $n > 1$ a synchronizing automaton $\mathscr{C}_n$ with n states over a binary alphabet whose shortest synchronizing word has length $(n-1)^2$. Later he conjectured that each synchronizing automaton with n states possesses a synchronizing word of length at most $(n-1)^2$. This conjecture has been proved for various classes of synchronizing automata, nevertheless in general it remains one of the most longstanding open problems in automata theory. For more details on synchronizing automata see the survey [3].

Recall that a DFA with a distinguished initial state and a distinguished set of final states is called *minimal* if it contains no (different) equivalent states and all states are reachable from the initial state. For a given regular language L, the minimal automaton recognizing L is unique up to isomorphism. The number of states in the minimal DFA is denoted by $sc(L)$ and is called *state complexity* of the language L.

In what follows we consider only *ideal* languages, that is, languages L satisfying the property $L = \Sigma^* L \Sigma^*$. It is obvious that the language of synchronizing words of a given synchronizing automaton satisfies this property. We have the following

Lemma 1. *Let L be an ideal language and $\mathscr{A}$ the minimal automaton recognizing L. Then $\mathscr{A}$ is synchronizing and $Syn(\mathscr{A}) = L$.*

Lemma 1 shows that for every ideal language L there is a synchronizing automaton $\mathscr{A}$ such that $Syn(\mathscr{A}) = L$. Thus, it is rather natural to find out how many states such an automaton $\mathscr{A}$ may have. We define the *reset complexity* $rc(L)$ of an ideal language L as the minimal possible number of states in a synchronizing automaton $\mathscr{A}$ such that $Syn(\mathscr{A}) = L$. By Lemma 1 we have $rc(L) \leq sc(L)$. Now it is of interest how big a gap between $rc(L)$ and $sc(L)$ can be. For a unary alphabet we obtain that the two numbers coincide.

Proposition 1. *Let L be an ideal language over a unary alphabet. Then $sc(L) = rc(L) = \ell + 1$, where ℓ is the minimum length of words in L.*

In contrast, for a binary alphabet the gap between $rc(L)$ and $sc(L)$ can be exponentially large. To prove this fact we consider examples of "slowly" synchronizing automata, i.e. automata whose shortest synchronizing words have length close to $(n-1)^2$. The first example belongs to Černý [2], the others are taken from [1]. For all these examples we have the following

Proposition 2. *For every "slowly" synchronizing automaton $\mathscr{A}_n$ with n states,*

$$sc(Syn(\mathscr{A}_n)) = 2^n - n \text{ and } rc(Syn(\mathscr{A}_n)) = n.$$

Thus, we see that the description of an ideal language L by means of an automaton for which L serves as the language of synchronizing words can be exponentially more succinct than the "standard" description via minimal automaton recognizing L.

Another interesting question concerns the uniqueness of automata yielding the minimum of reset complexity. Here we exhibit a strongly connected 6-state synchronizing automaton $\mathscr{C}_6$ and a 6-state synchronizing automaton $\mathscr{S}_6$ having a *sink* state (a state fixed by all letters) such that $\mathscr{C}_6$ and $\mathscr{S}_6$ have the same language of synchronizing words, namely $L = (a+b)^*(b^3ab^2a + a^2b^3a + abab^3a + ab^2ab^3a)(a+b)^*$. By an exhaustive computer search we have shown that L is not the language of synchronizing words for any synchronizing automaton with less than 6 states whence both $\mathscr{C}_6$ and $\mathscr{S}_6$ are minimal in terms of reset complexity. However the question that remains open is whether or not the uniqueness takes places within the class of automata with sink and within the class of strongly connected automata.

References

[1] D. S. Ananichev, V. V. Gusev, M. V. Volkov, *Slowly synchronizing automata and digraphs.* In P. Hliněný, A. Kučera (eds.), Mathematical Foundations of Computer Science. MFCS 2010. Lect. Notes Comp. Sci. **6281**, Berlin, Springer, 2010, 55–65.

[2] J. Černý, *Poznámka k homogénnym eksperimentom s konečnými automatami.* Mat.-Fiz. Čas. Slovensk. Akad. 1964. **14** (1964) 208–216 [in Slovak].

[3] M. V. Volkov. *Synchronizing automata and the Černý conjecture.* In C. Martín-Vide, F. Otto, H. Fernau (eds.), Languages and Automata: Theory and Applications. LATA 2008. Lect. Notes Comp. Sci. **5196**, Berlin, Springer, 2008, 11–27.

An Upper bound on the chromatic number of circle graphs without K_4

Gleb V. Nenashev

St.Petersburg State University

Let G be a circle graph without clique on 4 vertices. We proof that the chromatic number of G doesn't exceed 30.

Definition 1. *Let us fix a circle. A circle graph is an intersection graph of a set of chords of the circle.*

Definition 2. *A vertex coloring of a graph is proper, if any two adjacent vertices have different colors. The chromatic number $\chi(G)$ of a graph G is the least positive integer n such that there is a proper vertex coloring of G with n colors.*

As usual $w(G)$ denote the clique number of a graph G (the size of maximal clique of a graph G).

Let G be a circle graph. A lot of results about the chromatic numbers of circle graphs are known. In 1988 A. V. Kostochka [1] proved that $\chi(G) \leq 5$ for a circle graph G without K_3. In 1996 A. A. Ageev [2] constructed a circle graph G without K_3 and $\chi(G) = 5$, i.e. the bound from [1] is tight. In 1997 A. V. Kostochka and J. Kratochvil [3] proved that $\chi(G) \leq 2^{w(G)+6}$. In 1999 A. A. Ageev [4] proved the bound $\chi(G) \leq 3$ for a circle graph G without K_3 with girth at least 5 (i.e. without cicles of length 3 and 4).

In 2011 A. V. Kostochka and K. G. Milans [7] showed, that a circle graph without K_4 can be colored in 38 colors. We prove a stronger theorem.

Theorem 1. *Let G be a circle graph without K_4. Then $\chi(G) \leq 30$.*

In fact, we prove a more general statement.

Theorem 2. *Suppose that every circle graph with clique number at most k can be colored in n colors. Then every circle graph with clique number $k+1$ can be colored in $6n$ colors.*

This statement together with the results of [1] provide the result of theorem 1.

References

[1] A. V. Kostochka. *On upper bounds for the chromatic numbers of graphs.* Trudy Instituta Mathematiki 10, p.204-–226, 1988.

[2] A. A. Ageev. *A triangle-free circle graph with chromatic number 5.* Discrete Math. **152**, p. 295—298, 1996.

[3] A. V. Kostochka, J. Kratochvil. *Covering and coloring polygon-circle graphs.* Discrete Math. **163**, p.299—305, 1997.

[4] A. A. Ageev. *Every circle graph of girth at least 5 is 3-colourable.* Discrete Math. **195**, p.229—233, 1999.

[5] A. V. Kostochka. *Coloring intersection graphs of geometric figures with a given clique number.* Contemp. Math. **342**, p.127-–138, 2004.

[6] J. Cerny. *Coloring circle graphs.* Electronic Notes in Discrete Mathematics **29**, p. 457–461, 2007.

[7] A. V. Kostochka, K. G. Milans *Coloring clean and K_4-free circle graphs.* 2011, submitted.

Constructing Premaximal Binary Cube-free Words of Any Level

Elena A. Petrova, Arseny M. Shur
Ural State University Ekaterinburg, Russia

Abstract

We study the structure of the language of binary cube-free words. Namely, we are interested in the cube-free words that cannot be infinitely extended preserving cube-freeness. We show the existence of such words with arbitrarily long finite extensions, both to one side and to both sides.

The study of repetition-free words and languages remains quite popular in combinatorics of words: lots of interesting and challenging problems are still open. The most popular repetition-free binary languages are the *cube-free* language CF and the *overlap-free* language OF. The language CF is much bigger and has much more complicated structure. For example, the number of overlap-free binary words grows only polynomially with the length [7], while the language of cube-free words has exponential growth [3]. The most accurate bounds for the growth of OF is given in [6] and for the growth of CF in [11]. Further, there is essentially unique nontrivial morphism preserving OF [8], while there are uniform morphisms of any length preserving CF [5]. The sets of two-sided infinite overlap-free and cube-free binary words also have quite different structure, see [10].

Any repetition-free language can be viewed as a poset with respect to prefix, suffix, or factor order. In case of prefix [suffix] order, the diagram of such a poset is a tree; each node generates a subtree and is a common prefix [respectively, suffix] of its descendants. The following questions arise naturally. *Does a given word generate finite or infinite subtree? Are the subtrees generated by two given words isomorphic? Can words generate arbitrarily large finite subtrees?* For some power-free languages, the decidability of the first question was proved in [4] as a corollary of interesting structural properties. The third question for ternary square-free words constitutes Problem 1.10.9 of [1]. For all kth power-free languages, it was shown in [2] that the subtree generated by any word has at least one leaf. Note that considering the factor order instead of the prefix or the suffix one, we get a more general acyclic graph instead of a tree, but still can ask the same questions about the structure of this graph. For the language OF, all these questions were answered in [9, 12], but almost nothing is known about the same questions for CF.

In this paper, we answer the third question for the language CF in the affirmative. Namely, we construct cube-free words that generate subtrees of any prescribed depth and then extend this result for the subgraphs of the diagram of factor order.

Let $L \subset \Sigma^*$ and $W \in L$. Any word $U \in \Sigma^*$ such that $UW \in L$ is called a *left context* of W in L. The word W is *left maximal* [*left premaximal*] if it has no nonempty left contexts [respectively, finitely many left contexts]. The *level* of the left premaximal word W is the length of its longest left context; thus, left maximal words are of level 0. The right counterparts of the above notions are defined in a symmetric way. We say that a word is *maximal [premaximal]* if it is both left and right maximal [respectively, premaximal]. The *level* of a premaximal word W is the pair $(n, k) \in \mathbb{N}$ such that n and k are the length of the longest left context of W and the length of its longest right context, respectively.

The aim of this paper is to prove the following theorems:

Theorem 1. *In* CF*, there exist left premaximal words of any level* $n \in \mathbb{N}_0$.

Theorem 2. *In* CF*, there exist premaximal words of any level* $(n, k) \in \mathbb{N}_0^2$.

References

[1] J.-P. Allouche, J. Shallit (2003): *Automatic Sequences: Theory, Applications, Generalizations*, Cambridge Univ. Press.

[2] D. R. Bean, A. Ehrenfeucht, G. McNulty (1979): *Avoidable patterns in strings of symbols*, Pacific J. Math. **85**, 261–294.

[3] F.-J. Brandenburg (1983): *Uniformly growing k-th power free homomorphisms*, Theor. Comput. Sci. **23**, 69–82.

[4] J. D. Currie (1995): *On the structure and extendability of k-power free words*, European J. Comb. **16**, 111–124.

[5] J. D. Currie, N. Rampersad (2009): *There are k-uniform cubefree binary morphisms for all* $k \geq 0$, Discrete Appl. Math. **157**, 2548–2551.

[6] R. M. Jungers, V. Y. Protasov, V. D. Blondel (2009): *Overlap-free words and spectra of matrices*, Theor. Comput. Sci. **410**, 3670–3684.

[7] A. Restivo, S. Salemi (2002): *Words and Patterns*, Proc. 5th Int. Conf. Developments in Language Theory. Springer, Heidelberg, 117–129. (LNCS Vol. **2295**).

[8] P. Séébold (1984): *Overlap-free sequences*, Automata on Infinite Words. Ecole de Printemps d'Informatique Theorique, Le Mont Dore. Springer, Heidelberg, 207–215. (LNCS Vol. **192**).

[9] A. M. Shur (1998): *Syntactic semigroups of avoidable languages*, Sibirskii Matematicheskii Zhurnal, **39**, 683–702. [Russian; Engl. Transl. in Siberian Math. J. **39** (1998), 594–610.]

[10] A. M. Shur (2000): *The structure of the set of cube-free Z-words over a two-letter alphabet*, Izv. Math. **64**(4), 847–871.

[11] A. M. Shur (2009): *Two-sided bounds for the growth rates of power-free languages*, Proc. 13th Int. Conf. on Developments in Language Theory. Springer, Berlin, 466–477. (LNCS Vol. **5583**).

[12] A. M. Shur (2011): *Deciding context equivalence of binary overlap-free words in linear time*, Semigroup Forum. (Submitted)

Function and Image Manipulation with Automata

Turo Sallinen
Department of Mathematics and Turku Centre for Computer Science (TUCS)
University of Turku

Finite automata are among the simplest models of conventional computing. They operate on words over a finite alphabet. Instead of mere acceptance, we can use nondeterministic automata equipped with weights, first introduced by Schützenberger, to compute real valued functions.

We start by giving the basic definitions for computing functions in one and two dimensions. In the one dimensional case we see that integration is an easy operation and our class of automata is closed with respect to it. Even a fractal type function is simple to integrate, unlike in traditional calculus. In two dimensions functions are depicted in grayscale images and we see that many basic image transformations – such as product, adding, change of contrast, squeezing, zooming, rotation, integration and (partial) derivation – are simple to implement. Most importantly, all the transformations can be done directly on the automata representation without computing the function itself.

On Abelian Repetition Threshold

Alexey V. Samsonov, Arseny M. Shur
Ural State University, Ekaterinburg, Russia

Abstract

We study the avoidance of Abelian powers of words and consider three reasonable generalizations of the notion of Abelian power to fractional powers. Our main goal is to find an Abelian analogue of the repetition threshold, i. e. a numerical value separating k-avoidable and k-unavoidable Abelian powers for each size k of the alphabet. We prove lower bounds for the Abelian repetion threshold for large alphabets and all definitions of Abelian fractional power. We develop a method estimating the exponential growth rate of Abelian-power-free languages. Using this method, we get non-trivial lower bounds for Abelian repetition threshold for small alphabets. We suggest that some of the obtained bounds are the exact values of Abelian repetition threshold.

Introduction

The study of avoidable powers of words has more than a centennial history since the paper by Thue [15]. If w is a word, $|w|$ is its length, $\beta > 1$ is a number, then w^β is a unique prefix v of the infinite word $www\ldots$, whose length satisfies the conditions $|v|/|w| \geq \beta$, $(|v|-1)/|w| < \beta$. A word u is *β-free*, if none of its factors, including u itself, is a β-power. A β-power is said to be *k-avoidable* if there are infinitely many β-free words (or, equivalently, an infinite β-free word) over the k-letter alphabet, and *k-unavoidable* otherwise.

For any k-letter alphabet ($k \geq 2$), the *repetition threshold* is the number $RT(k)$ which separates k-unavoidable and k-avoidable powers of words. Famous Dejean's conjecture [8] (proven in [3, 5, 12]) states that $RT(3) = 7/4$, $RT(4) = 7/5$, and $RT(k) = k/(k-1)$ otherwise.

Abelian powers of words were first considered in [10]. The word $w_1w_2\dots w_n$ is an *Abelian n-th power*, if each of the words $w_2,\dots,w_n$ is an anagram of w_1. The avoidability of Abelian integral powers is well studied ([1, 4, 4, De, Ke]). In contrast with the usual powers, there are several ways to generalize the notion of Abelian power to fractional exponents. We define weak, semistrong and strong Abelian fractional powers and then work with all three definitions. Once a definition of Abelian fractional power is chosen, Abelian repetition threshold can be defined in the same way as the "usual" repetition threshold. We study the values of Abelian repetition threshold for all alphabets and three suggested definitions.

Definitons

Let $\Sigma=\{1,\dots,k\}$ be an alphabet and $w\in\Sigma^*$ be an arbitrary k-ary word. The Parikh vector $\vec{p}(w)$ is the vector of length k whose ith component equals the number of occurrences of the letter i in w, for any $i=1,\dots,k$. If $v\in\Sigma^*$, then the notation $\vec{p}(w)\le\vec{p}(v)$ means that the ith component of $\vec{p}(w)$ is not greater than the ith component of $\vec{p}(v)$, for any $i=1,\dots,k$.

Let $m\ge 2$ be an integer. An *Abelian m-power* is a word of the form $w_1w_2\dots w_m$, where w_i is an anagram of w_1 for $2\le i\le m$, or $\vec{p}(w_1)=\ldots=\vec{p}(w_m)$. Now we extend this definition to the rational numbers in the range $(1,\infty)$. Let $\beta>1$, $|w_1|=q$, $m=\lfloor\beta\rfloor$, $t=\lceil\{\beta\}q\rceil$, where $\{\beta\}$ stands for the fractional part of β. Consider a word of the form $w=w_1\dots w_mv$, where $w_1\dots w_m$ is an abelian m-power and $|v|=t$. The terms *root* and *tail* denote the words w_1 and v respectively. We consider three different restrictions upon the Parikh vector of the tail, thus obtaining three definitions of Abelian fractional power. Let $\mathrm{pref}(u,l)$ be the prefix of length l of the word u.

A *weak Abelian β-power* is a word w of the form described above such that $\vec{p}(v)\le\vec{p}(w_1)$. That is, the tail is a prefix of an anagram of the root.

A *strong Abelian β-power* is a word w of the form described above such that $\vec{p}(v)=\vec{p}(\mathrm{pref}(w_1,t))$. That is, the tail is an anagram of a prefix of the root.

A *semistrong Abelian β-power* is a word w of the form described above such that $\vec{p}(v)\le\bigvee\limits_{i=\overline{1,m}}\vec{p}(\mathrm{pref}(w_i,t))$, where $\bigvee$ is the operation of taking maximum componentwise.

Example. The word *abc cba ac* is a semistrong and a weak Abelian (8/3)-power, but not a strong Abelian (8/3)-power, because *ac* is not a permutation of *ab*. The word *abcaa* is not even a weak Abelian (5/3)-power, but is a strong, semistrong and weak Abelian (5/4)-power.

Abelian exponent of a word w is the maximal rational number β such that w is an Abelian β-power. A word w is *Abelian-β-free* if all its factors have Abelian exponents less than β. By *Abelian-β-free languages* we mean the languages of *all* Abelian-β-free words over a given alphabet. We consider three types of Abelian-power-free languages (weak, semistrong and strong).

Results

We prove uniform lower bounds for both strong and weak Abelian repetition threshold (denoted by $ART_s(k)$ and $ART_w(k)$, respectively). In view of the numerical results, it looks highly probable that our bound for $ART_s(k)$ is exact, while the bound for $ART_w(k)$ can be improved.

Theorem 1. $ART_w(k)\ge\frac{k}{k-2}$ for all $k\ge 10$.

Theorem 2. $ART_s(k)\ge\frac{k-2}{k-3}$ for all $k\ge 5$.

We use method proposed in [14] to obtain upper bounds for the growth rates of certain Abelian power-free languages. These bounds provide a strong evidence that Abelian repetition thresholds for strong and semistrong Abelian powers coincide, so we are able to formulate the following

Conjecture 1. The Abelian repetition threshold for strong and semistrong Abelian powers is given by

$$ART_s(k) = \begin{cases} 11/3, & k = 2, \\ 2, & k = 3, \\ 9/5, & k = 4, \\ (k-2)/(k-3), & k \geq 5. \end{cases}$$

References

[1] A. Aberkane, J. D. Currie, N. Rampersad, *The number of ternary words avoiding Abelian cubes grows exponentially*, J. Int. Seq., **7** (2004), #04.2.7, 13 pp. (electronic).

[2] F.-J. Brandenburg, *Uniformly growing k-th power free homomorphisms*, Theor. Comput. Sci., **23** (1983), 69-82.

[3] A. Carpi, *On Dejean's conjecture over large alphabets*, Theor. Comput. Sci., **385** (2007), 137–151.

[4] A. Carpi, *On the number of Abelian square-free words on four letters*, Discr. Appl. Math., **81** (1998), 155–167.

[5] M. Crochemore, F. Mignosi, A. Restivo, *Automata and forbidden words*, Inform. Processing Letters, **67**(3) (1998), 111-117.

[6] J. D. Currie, *The number of binary words avoiding Abelian fourth powers grows exponentially*, Theor. Comput. Sci., **319**(1–3) (2004), 441–446.

[7] J. D. Currie, N. Rampersad, *A proof of Dejean's conjecture*, Math. Comp. **80** (2011), 1063–1070.

[8] F. Dejean, *Sur un Théorème de Thue*, J. Comb. Theory A **13**(1) (1972), 90-99.

[9] F. M. Dekking, *Strongly non-repetitive sequences and progression-free sets*, J. Combin. Theory A **27** (1979), 181-185.

[10] P. Erdös, *Some unsolved problems*, Magyar Tud. Akad. Mat. Kutató Int. Közl. **6** (1961), 221–264.

[11] V. Keränen, *Abelian squares are avoidable on 4 letters*, Proc. ICALP'92, 41–52. Springer, Berlin, 1992. (LNCS **623**).

[12] M. Rao, *Last Cases of Dejean's Conjecture*, Theor. Comput. Sci. **412** (2011), 3010–3018. Combinatorics on Words (WORDS 2009), 7th International Conference on Words.

[13] A.M. Shur, *Comparing complexity functions of a language and its extendable part*, RAIRO Theor. Inf. Appl. **42** (2008), 647–655.

[14] A. M. Shur, *Growth rates of complexity of power-free languages*, Theor. Comput. Sci. **411** (2010), 3209–3223.

[15] A. Thue, *Über unendliche Zeichenreihen*, Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl., Christiana **7** (1906), 1-22.

Synchronizing random automata on 4-letter alphabet

Evgeny Skvortsov, Yulia Zaks
Department of Mathematics and Mechanics, Ural Federal University

Abstract

The paper deals with the synchronization of a random automaton that is sampled uniformly at random from the set of all automata with n states and m letters. We show that for $m = 4$ the probability that a random automaton is synchronizing is larger than a positive constant.

Let $\mathcal{A} = (Q, \Sigma, \delta)$ be a *deterministic finite automaton* (DFA), where Q denotes a state set, Σ stands for an input alphabet, and $\delta : Q \times \Sigma \to Q$ is a transition function defining an action of the letters in Σ on Q. A word w is said to be a *reset word* for DFA $\mathcal{A}$ if its action leaves $\mathcal{A}$ in one particular state no matter what state it starts at: $\delta(q_1, w) = \delta(q_2, w)$ for all $q_1, q_2 \in Q$. A DFA $\mathcal{A}$ is called *synchronizing* if it possesses a reset word.

Since synchronizing automata are applied in different areas: robotics, model-based testing of reactive systems, symbolic dynamics, DNA computing and many others (for example, [6]), efficient algorithms finding a reset word are of utmost necessity. One of the parameters defining the quality of such algorithms is the length of the reset word and this draws attention to the problem of estimating this length.

The best up do date upper bound on the length of the shortest reset word for DFA with n states equals $(n^3 - n)/6$; it was obtained by Pin [4] in 1983. The conjecture that the length cannot be larger than $(n - 1)^2$ formulated by Černý has been proved for some classes of the automata but remains unproven in the general case.

In fact, *slowly synchronizing automata*, i.e. automata with the shortest reset word of length $\Theta(n^2)$ are known to be exceptional. For a long time the only infinite series of such automata was the original one proposed by Černý [3]. The other substantially different ones [1, 2] have only recently been constructed. This fact in combination with the results of the numerical experiments make the random case more important in practical sense than the extremal one.

Let us define the notion of random automaton and state the questions of interest concerning its synchronization properties.

Consider a set of states Q and an alphabet Σ. Let us pick uniformly at random a transition function δ from the set $\{\delta : Q \times \Sigma \to Q\}$. A resulting triple (Q, Σ, δ) defines a *random deterministic finite automaton*. It is important to note that a random automaton can be constructed as follows: for each $q \in Q$ and for each $a \in \Sigma$ we choose $q' = \delta(q, a)$ uniformly at random from Q.

We are interested in the following questions:

- What size of an alphabet does imply that a random automaton with the alphabet of this size is synchronizing *with high probability* (whp) and what is the length of the shortest reset word in this case? (By "high probability" we mean that the probability tends to 1 with n going to infinity.)
- What size of an alphabet does imply that a random automaton with the alphabet of this size is synchronizing and complies with the Černý conjecture whp?
- What size of an alphabet does imply that a random automaton with the alphabet of this size is synchronizing *with constant probability* (wcp)? (By "constant probability" we mean that the probability is bounded from below by a positive constant with n going to infinity.)

In [5] we give partial answers to the first two questions for automata with n states and $m(n)$ letters. In this paper we address the third question and show that a random automaton with the alphabet size independent of the number of states is synchronizing though with constant probability. Our main result is the following theorem.

Theorem 1. *There is a constant $p_0 > 0$ such that for any natural number n a random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = n, |\Sigma| = 4$ is synchronizing with the probability greater than p_0.*

References

[1] Ananichev D.S., Gusev V., Volkov M.V. Slowly synchronizing automata and digraphs. In Proc. Conf. Math. Found. Comp. Sci., Lect. Notes Comp. Sci. 2010. V.6281. P.55–65.

[2] Ananichev D. S., Volkov M. V., Zaks Yu.I., Synchronizing automata with a letter of deficiency 2, Theoret. Comput. Sci. 2007 V.376 P.30–41.

[3] Černý J. Poznámka k homogénnym eksperimentom s konečnými auto-matami. Mat.-Fyz. Čas. Slovensk. Akad. Vied. 1964. V.14. P.208–216. [in Slovak]

[4] Pin J.-E. On two combinatorial problems arising from automata theory. Ann. Discrete Math. 1983. V.17. P.535–548.

[5] Skvortsov E., Zaks Yu. Synchronizing random automata. DMTCS. 2010 V.12:4. P.95–108.

[6] Volkov M. V. Synchronizing automata and the Černý conjecture. In C. Martín-Vide, F. Otto, H. Fernau (eds.), Languages and Automata: Theory and Applications. LATA 2008, Lect. Notes Comp. Sci., Springer-Verlag, Berlin-Heidelberg-New York. 2008. V.5196. P.11–27.