

Combinatorics and Algorithmics of Strings

Edited by

Maxime Crochemore¹, James Currie², Gregory Kucherov³, and Dirk Nowotka⁴

1 King's College – London, GB, maxime.crochemore@kcl.ac.uk

2 University of Winnipeg, CA, j.currie@uwinnipeg.ca

3 Université Paris-Est – Marne-la-Vallée, FR, gregory.kucherov@univ-mlv.fr

4 Christian-Albrechts-Universität zu Kiel, DE, dn@zs.uni-kiel.de

Abstract

Strings (aka sequences or words) form the most basic and natural data structure. They occur whenever information is electronically transmitted (as bit streams), when natural language text is spoken or written down (as words over, for example, the Latin alphabet), in the process of heredity transmission in living cells (through DNA sequences) or the protein synthesis (as sequence of amino acids), and in many more different contexts. Given this universal form of representing information, the need to process strings is apparent and is actually a core purpose of computer use. Algorithms to efficiently search through, analyze, (de-)compress, match, encode and decode strings are therefore of chief interest. Combinatorial problems about strings lie at the core of such algorithmic questions. Many such combinatorial problems are common in the string processing efforts in the different fields of application.

The purpose of this seminar is to bring together researchers from different disciplines whose interests are string processing algorithms and related combinatorial problems on words. The two main areas of interest for this seminar are Combinatorics on Words and Stringology. This report documents the program and the outcomes of Dagstuhl Seminar 14111 “Combinatorics and Algorithmics of Strings”.

Seminar March 9–14, 2014 – <http://www.dagstuhl.de/14111>

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics

Keywords and phrases combinatorics on words, string algorithms, automata

Digital Object Identifier 10.4230/DagRep.4.3.28

Edited in cooperation with Robert Mercas

1 Executive Summary

Maxime Crochemore

James Currie

Gregory Kucherov

Dirk Nowotka

License © Creative Commons BY 3.0 Unported license

© Maxime Crochemore, James Currie, Gregory Kucherov, and Dirk Nowotka

Processing strings efficiently is of concern in practically every application field. Understanding the combinatorial properties of sequences is a prerequisite for designing efficient algorithms on them. The Dagstuhl seminar 14111 has been concerned with exactly that: *Combinatorics and Algorithmics of Strings*.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Combinatorics and Algorithmics of Strings, *Dagstuhl Reports*, Vol. 4, Issue 3, pp. 28–46

Editors: Maxime Crochemore, James Currie, Gregory Kucherov, and Dirk Nowotka



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This Dagstuhl seminar was attended by 41 researchers from 12 countries representing the two fields, algorithmics and combinatorics, about equally, although it needs to be mentioned that the overlap of these two communities is rather large. Inviting these close communities to Dagstuhl gave us the opportunity to start from substantial common ground and to work on scientific problems right from the beginning. Given that background, tutorials or other introductory sessions were not considered to be suitable elements for this seminar. Instead, much time was spent for problem posing and solving sessions. This seminar has clearly been research oriented.

The first seminar day, Monday, was entirely devoted to posing open problems. Based on those, the participants were able to form interest groups and engage into research activities early on. In the next days regular research talks and some more open problems were presented. However, time slots for research work were also allocated. On the last day of the seminar, Friday, we were able to already present some solutions to the problems posed in the beginning. In general, it is not to be expected that research problems are solved within a week (and most weren't), but it illustrates the impact of the meeting on catalysing research and collaboration between the participants.

The following two are great examples of such collaboration. Florin Manea asked about the complexity of deciding whether or not two words u and w are k -binomial equivalent, that is, is the number of occurrences of all scattered subwords up to length k equal in u and w ? Contributions by Paweł Gawrychowski (polynomial Monte-Carlo algorithm in the logarithmic word-size RAM model), Juhani Karhumäki, and Wojciech Rytter (polynomial time on a unit-cost RAM model), and discussions with Dominik Freydenberger and Manfred Kufleitner finally led to the conclusion that the problem can be solved in polynomial time in the logarithmic word-size RAM model. Another problem was posed by Juhani Karhumäki and Michaël Rao (not present at the seminar) on the avoidability of shuffle squares. They asked: Does there exist an infinite word over some finite alphabet which avoids all factors that are a shuffle product of a word with itself? James Currie realized that shuffle squares can indeed be avoided applying the Lovász Local Lemma in his argument. However, this solution of avoidability in principle led to a proof for a very large alphabet, the size of which being a number of more than 40 digits. A few days after this Dagstuhl seminar Mike Müller improved that result by giving a rather low upper bound on the alphabet size of 10 on which shuffle squares can be avoided using a recent result by Joseph Miller. In general, it has to be noted that progress was made in many more areas and several papers in preparation were announced already.

Another notable highlight of the seminar was a session dedicated to word equations. Senior researchers of that particular research area, like Wojciech Plandowski and Volker Diekert, and young protagonists, like Aleksa Saarela, Štěpán Holub, and Artur Jež, who talked about their recent efforts in developing the field, contributed and exchanged ideas. Such a unique assembly of major experts in word equations and their contributions at Dagstuhl was rather unique and a remarkable event.

In the light of such developments, it can be safely claimed that this seminar was a great success. Given the quality of presentations on this seminar and the constructive intensity of discussions, it is self-evident that a follow-up should be organised. We are grateful to all participants for their contributions to this successful seminar as well as to the staff of Schloss Dagstuhl for their great service.

2 Table of Contents

Executive Summary

Maxime Crochemore, James Currie, Gregory Kucherov, and Dirk Nowotka 28

Overview of Talks, Open Problems, and Solutions

Near Real-Time Suffix Tree Construction via the Fringe Marked Ancestor Problem <i>Danny Breslauer</i>	32
Avoidability of Shuffle Squares <i>James D. Currie</i>	32
Hairpins and unambiguous context-free languages <i>Volker Diekert</i>	33
On The Minimum Number of Abelian Squares in a Word <i>Gabriele Fici</i>	34
On The Maximum Number of Abelian Squares in a Word <i>Gabriele Fici</i>	35
Are there better measures of compressibility than Empirical Entropy? <i>Johannes Fischer</i>	36
Two open problems on pattern languages <i>Domínik D. Freydenberger</i>	36
Decomposition to palindromes <i>Anna E. Frid</i>	37
Order-preserving pattern matching with k mismatches <i>Paweł Gawrychowski</i>	38
Algebraic properties of word equations <i>Štěpán Holub</i>	38
Local Recompression for Word Equations <i>Artur Jež</i>	38
String Range Matching <i>Juha Kärkkäinen</i>	39
Sum of Digits of n and n^2 <i>Steffen Kopecki</i>	39
The Burrows-Wheeler Transform with Permutations <i>Manfred Kufleitner</i>	40
Text Indexing: Easy and Difficult <i>Moshe Lewenstein</i>	40
Testing k -binomial equivalence <i>Florin Manea</i>	41
k -Abelian Pattern Matching <i>Robert Mercas</i>	41
Bell numbers modulo 8 <i>Eric Rowland</i>	42

Maximum Number of Distinct and Nonequivalent Nonstandard Squares in a Word <i>Wojciech Rytter</i>	43
Parametric solutions of word equations <i>Aleksi Saarela</i>	43
Efficient generation of repetition-free words <i>Arseny M. Shur</i>	44
Participants	46

3 Overview of Talks, Open Problems, and Solutions

3.1 Near Real-Time Suffix Tree Construction via the Fringe Marked Ancestor Problem

Danny Breslauer (*University of Haifa, Israel*)

License © Creative Commons BY 3.0 Unported license
© Danny Breslauer

Joint work of Danny Breslauer; Giuseppe F. Italiano

Main reference D. Breslauer, G. F. Italiano, “Near Real-Time Suffix Tree Construction via the Fringe Marked Ancestor Problem,” *Journal of Discrete Algorithms*, 18:32–48, 2013.

URL <http://dx.doi.org/10.1016/j.jda.2012.07.003>

We contribute a further step towards the plausible *real-time* construction of suffix trees by presenting an on-line suffix tree algorithm that spends only $\mathcal{O}(\log \log n)$ time processing each input symbol and takes $\mathcal{O}(n \log \log n)$ time in total, where n is the length of the input text. Our results improve on a previously published algorithm that takes $\mathcal{O}(\log n)$ time per symbol and $\mathcal{O}(n \log n)$ time in total. The improvements are obtained by adapting Weiner’s suffix tree construction algorithm to use a new data structure for the fringe marked ancestor problem, a special case of the nearest marked ancestor problem, which may be of independent interest.

3.2 Avoidability of Shuffle Squares

James D. Currie (*University of Winnipeg, CA*)

License © Creative Commons BY 3.0 Unported license
© James D. Currie

A **shuffle square** is a word w such that for some word $v = \prod_{i=1}^n a_i = \prod_{i=1}^n b_i$ with $a_i, b_i \neq \epsilon, 1 \leq i \leq n-1, a_n \neq \epsilon$, we can write

$$w = \prod_{i=1}^n (a_i b_i).$$

We then write $w \in v \sqcup v$. On the first day of the 2014 Dagstuhl seminar, *Combinatorics and Algorithmics of Strings*, J. Karhumäki asked the following question:

► **Question 1.** *Are shuffle squares avoidable?*

That is, whether for a large enough alphabet Σ , there is a word of Σ^ω in which no factor is a shuffle square. On the last day of the seminar, I pointed out that a very basic application of the Lovász Local Lemma gives avoidability.

► **Theorem 1.** *Shuffle squares are k -avoidable, where $k = \lceil e^{95} \rceil$.*

Evidently, it would be desirable to have a construction, and it remained to bring k down to some reasonable size. Much better bounds on the alphabet size (currently, $k = 10$) have been obtained by Mike Müller, cleverly using the criterion of Miller, recently promoted by Rampersad.

Probabilistic methods will also show that **shuffle powers** are avoidable. A shuffle r -power is a word $w \in x \sqcup p$ for some words x and p with p a prefix of x and $|xp|/|x| \geq r$.

The question of minimal alphabet sizes for avoidability remains open, and a construction is needed.

3.3 Hairpins and unambiguous context-free languages

Volker Diekert (Universität Stuttgart, DE)

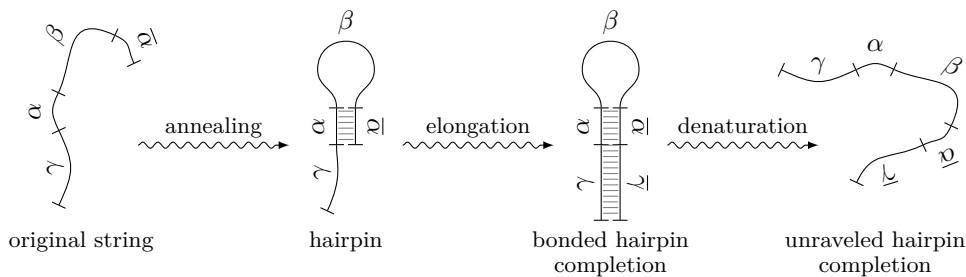
License © Creative Commons BY 3.0 Unported license
© Volker Diekert

Joint work of Volker Diekert; Steffen Kopecki; Victor Mitrana

Main reference V. Diekert, S. Kopecki, V. Mitrana, “Deciding regularity of hairpin completions of regular languages in polynomial time,” Information and Computation, 217:12–30, 2012.

URL <http://dx.doi.org/10.1016/j.ic.2012.04.003>

In DNA computing one deals with strings over the bases $A, C, G,$ and T . The Watson-Crick base pairing connects the bases A and T (resp. C and G) via hydrogen bonds; and the bases A and T (resp. C and G) are complementary. On an abstract level, $\{A, C, G, T\}$ forms a finite alphabet with *involution* Σ . That is for each $a \in \Sigma$ there is a unique $\bar{a} \in \Sigma$ such that $\bar{\bar{a}} = a$ for all a . In the case $\Sigma = \{A, C, G, T\}$ we have $\bar{A} = T$ and $\bar{C} = G$. A string of the form $\gamma\alpha\beta\bar{\alpha}$, where α is not too short (say $|\alpha| \geq 9$), may create a *hairpin* during annealing. This process may lead to elongation and denaturation; and new strings may occur as follows:



In an abstract setting, a *hairpin completion* transforms a string $\gamma\alpha\beta\bar{\alpha}$ into $\gamma\alpha\beta\bar{\alpha}\bar{\gamma}$ for $|\alpha| \geq \kappa$ where κ is some fixed small constant. This yields a transformation on formal languages:

$$L \mapsto H(L) = \{\gamma\alpha\beta\bar{\alpha}\bar{\gamma} \in \Sigma^* \mid \gamma\alpha\beta\bar{\alpha} \in L\}.$$

There is also a more symmetric (and more interesting) variant

$$L \mapsto \{\gamma\alpha\beta\bar{\alpha}\bar{\gamma} \in \Sigma^* \mid \gamma\alpha\beta\bar{\alpha} \in L \vee \alpha\beta\bar{\alpha}\bar{\gamma} \in L\}.$$

However, for simplicity of the presentation it is enough to consider the case $L \mapsto H(L)$. The following two facts are known for regular languages $L \subseteq \Sigma^*$ by [1]:

- Given a DFA for L with n states, we can decide in time $\mathcal{O}(n^2)$ whether or not the hairpin completion $H(L)$ is regular.
- If L is regular then $H(L)$ is an unambiguous linear context-free language.

Analogous results hold also for the the symmetric variants, but proofs are much more demanding and complexities increase. This leads to the following two problems where the second one is motivated by our study of hairpin completions. Clearly, it has its interest in formal languages theory in its own right.

► **Question 2.** *Is the following problem PSPACE-complete? The input is an NFA for L with n states. The question is whether or not the hairpin completion $H(L)$ is regular.*

► **Question 3.** *Is the following problem decidable? The input is an unambiguous (linear) context-free grammar G . The question is whether the generated language $L(G)$ is regular.*

A positive solution of the second problem would show that the decidability of the regularity for $H(L)$ (for regular L) is merely a special case of a more general situation. There are many more interesting open problems about hairpin formations, for example it is not known whether or not regularity of the *iterated hairpin completion of a singleton language* is decidable. For the exact statement of the problem and related questions we refer to [1, 2].

References

- 1 V. Diekert, S. Kopecki, and V. Mitrana. Deciding regularity of hairpin completions of regular languages in polynomial time. In *Information and Computation*, 217:12–30, 2012.
- 2 L. Kari, S. Kopecki, and S. Seki. Iterated hairpin completions of non-crossing words. In M. Bieliková et al. (editors) *38th Conference on Current Trends in Theory and Practice of Computer Science*, volume 7147 of LNCS, pages 337–348, 2012.

3.4 On The Minimum Number of Abelian Squares in a Word

Gabriele Fici (University of Palermo, IT)

License  Creative Commons BY 3.0 Unported license
© Gabriele Fici

Joint work of Gabriele Fici; Aleksii Saarela

An abelian square is a word that can be written as uv , where v is obtained from u by permuting letters (i.e., v is an anagram of u). Given an alphabet size k and an integer n , let $f_k(n)$ be the least number of abelian square factors that a word of length n over an alphabet of size k must contain. It is known that $f_k(n) = 0$ for every n if $k \geq 4$ [2], and it has been conjectured by Mäkelä in 1992 that $f_3(n) = 3$ for sufficiently large n , but this conjecture seems hard to prove¹. On the other side, it is easy to see that $f_1(n) = \lfloor n/2 \rfloor$. For the case $k = 2$, we have partial results supporting the conjecture that $f_2(n) = \lfloor n/4 \rfloor$. More details follow.

► **Definition 1.** A word w is Abelian Square Minimal (ASM) if no other word of the same length over the same alphabet contains less abelian squares than w .

► **Definition 2.** An abelian square of the form a^{2i} , for some letter a and integer $i > 0$, is called a trivial abelian square.

We have proved the following result:

► **Lemma 3.** *Let w be a binary word of length n containing only trivial abelian squares. Then $|AS(w)| \geq \lfloor n/4 \rfloor$.*

On the other hand, a sequence of binary words of length n containing only $\lfloor n/4 \rfloor$ distinct abelian squares is easy to show (take a word with only one b , placed in the middle). Hence, to prove that $f_2(n) = \lfloor n/4 \rfloor$, it is sufficient to prove the following:

► **Conjecture 4.** *Let w be a binary ASM word of length $|w| > 4$. Then w contains only trivial abelian squares.*

¹ Actually, Mäkelä asked if there exist arbitrarily large ternary words containing no abelian square of length greater than 2 [3]. Rampersad performed computer searches yielding words of length at least 3160 satisfying this condition [4].

Should the formula $f_2(n) = \lfloor n/4 \rfloor$ be true, we would have that a longest binary word containing only n abelian squares has length $4n + 3$. More precisely, it would be the word $a^{2n+1}ba^{2n+1}$ or its complement $b^{2n+1}ab^{2n+1}$.

This is related to a conjecture of Fraenkel, Simpson and Paterson [1], who considered the minimum number of *inequivalent* abelian squares (that is, having different Parikh vectors) in a binary word:

► **Conjecture 5.** *A longest word containing only n inequivalent abelian squares has length $4n + 3$, and has one of the forms: $(ab)^{2n+1}a$, $a^{2n+1}ba^{2n+1}$ or their complements.*

References

- 1 A. S. Fraenkel, J. Simpson, and M. Paterson. On weak circular squares in binary words. In: *CPM 1997, LNCS 1264*, 76–82. Springer, 1997.
- 2 V. Keränen. Abelian squares are avoidable on 4 letters. In: *ICALP 1992, LNCS 623*, 41–52. Springer-Verlag, 1992.
- 3 S. Mäkelä. Patterns in words. Master’s thesis, University of Turku, Finland, 2002.
- 4 N. Rampersad. Infinite sequences and pattern avoidance. Master’s thesis, University of Waterloo, Canada, 2004.

3.5 On The Maximum Number of Abelian Squares in a Word

Gabriele Fici (University of Palermo, IT)

License © Creative Commons BY 3.0 Unported license
© Gabriele Fici

An abelian square is a word that can be written as uv , where v is obtained from u by permuting letters (i.e., v is an anagram of u). Given a word w of length n , we investigate the maximum number of factors of w that are abelian squares. Contrarily to the case of ordinary squares, where this number is linear in n , it is easy to show that a word of length n can contain $\Theta(n^2)$ many distinct abelian squares. Take for example $w_n = a^n ba^n ba^n$. For any $0 \leq i, j \leq n$, if the factor $a^i ba^n ba^j$ has even length, then it is an abelian square. Therefore, w_n contains $(n^2 + 3n + 1 + (-1)^n)/2$ many distinct abelian squares. This example motivates us to search for infinite words all factors of which contain a quadratic (in their length) number of distinct abelian squares.

► **Definition 1.** Let $as_w(n)$ denote the minimum number of distinct abelian squares in a factor of w of length n . An infinite word w is Abelian Square Rich if $as_w(n) = \Theta(n^2)$.

Together with Julien Cassaigne, we proved that the Thue-Morse word (that is the fixed point of the substitution $\mu : 0 \mapsto 01, 1 \mapsto 10$) is Abelian Square Rich.

We raise the following question:

► **Question 4.** *Is every Sturmian word Abelian Square Rich?*

First, we can prove that a factor of a Sturmian word is an abelian square (resp. an abelian k -power) if and only if both of its Parikh vector components are divisible by 2 (resp. by k). Then, using standard techniques of Number Theory based on results of approximation of irrationals by rationals, we can prove that the number of factors of length n of a Sturmian word s that are abelian squares is, on average, linear in n . This implies the following result:

► **Theorem 2.** *Let s be a Sturmian word. If s is k -power free for some $k \in \mathbb{R}^+$, then s is Abelian Square Rich.*

For example, the Fibonacci word (that is the fixed point of the substitution $\phi : 0 \mapsto 01, 1 \mapsto 0$) is $\frac{5+\sqrt{5}}{2}$ -power free, and therefore is Abelian Square Rich.

A slight different point of view consists in considering two abelian squares inequivalent if they have different Parikh vectors, and not simply if they are different words [1]. Sturmian words only have a linear number of inequivalent abelian squares, since they have abelian complexity equal to 2 for every $n > 0$. Nevertheless, computations support the following conjecture (also proposed by W. Rytter [2]).

► **Conjecture 3.** *A word of length n contains $O(n\sqrt{n})$ many inequivalent abelian squares.*

We propose the following open problem:


► **Question 5.** *Let $ias_w(n)$ denote the minimum number of distinct inequivalent abelian squares in a factor of w of length n . Does an infinite word w exist such that $ias_w(n) = \Theta(n\sqrt{n})$?*

References

- 1 A. S. Fraenkel, J. Simpson, and M. Paterson. On weak circular squares in binary words. In: *CPM 1997, LNCS 1264*, 76–82. Springer, 1997.
- 2 W. Rytter. Personal communication, 2014.

3.6 Are there better measures of compressibility than Empirical Entropy?

Johannes Fischer (TU Dortmund, DE)

License  Creative Commons BY 3.0 Unported license
© Johannes Fischer

Empirical entropy as a complexity measure is widely used in the analysis of data structures and algorithms, although it does not capture very common types of text regularities. We ask what one should expect from a better measure of compressibility, and propose a measure based on longest common prefixes.

3.7 Two open problems on pattern languages

Dominik D. Freydenberger (Goethe-Universität Frankfurt am Main, DE)

License  Creative Commons BY 3.0 Unported license
© Dominik D. Freydenberger

A *pattern* is a word $\alpha \in (\Sigma \cup X)^+$, where Σ and X are disjoint alphabets (of *terminals* and *variables*, respectively). A pattern α generates the language

$$L_{NE,\Sigma}(\alpha) := \{\sigma(\alpha) \mid \sigma \text{ is a substitution}\},$$

where a *substitution* is a morphism $\sigma : (\Sigma \cup X)^+ \rightarrow \Sigma^+$ with $\sigma(a) = a$ for all $a \in \Sigma$.

3.7.1 Degrees of ambiguity

For every pattern α and every word $w \in L_{\text{NE},\Sigma}(\alpha)$, the *degree of ambiguity of w (w. r. t. α)* is the number of distinct substitutions σ with $\sigma(\alpha) = w$. The degree of ambiguity of α is the maximal degree of ambiguity of any word $w \in L_{\text{NE},\Sigma}(\alpha)$. As shown by Mateescu and Salomaa [1], for every $k = 2^m 3^n$ ($m, n \geq 0$), a pattern with degree of ambiguity k can be effectively constructed. For all other finite degrees of ambiguity, even the existence of such patterns is unknown:

► **Question 6.** *Are there patterns with degree of ambiguity k such that k is not of the form $k = 2^m 3^n$ ($m, n \geq 0$)?*

3.7.2 Inclusion depth

For a pattern α , we define its *inclusion depth* $\text{ID}_\Sigma(\alpha)$ as the largest n for which there exist patterns $\beta_1, \dots, \beta_{n-1}$ with

$$\Sigma^+ \supset L_{\text{NE},\Sigma}(\beta_1) \supset \dots \supset L_{\text{NE},\Sigma}(\beta_{n-1}) \supset L_{\text{NE},\Sigma}(\alpha).$$

By definition, $\text{ID}_\Sigma(\alpha)$ is always finite, and Luo [2] gives the lower bound $\text{ID}_\Sigma(\alpha) \geq 2|\alpha| - |\text{var}(\alpha)| - 1$, where $\text{var}(\alpha)$ is the set of variables in α . Apart from this, little is known about $\text{ID}_\Sigma(\alpha)$. In particular, the following question is open:

► **Question 7.** *Given a pattern α , can we compute $\text{ID}_\Sigma(\alpha)$?*

References

- 1 A. Mateescu and A. Salomaa, Finite Degrees of Ambiguity in Pattern Languages. *RAIRO ITA*, 28(3–4):233–253, 1994.
- 2 W. Luo, Compute Inclusion Depth of a Pattern. In *Proc. COLT 2005*, LNAI 3559, pp. 689–690, 2005.

3.8 Decomposition to palindromes

Anna E. Frid (Aix-Marseille University, FR)

License © Creative Commons BY 3.0 Unported license
© Anna E. Frid

Joint work of Anna Frid; Svetlana Puzynina; Luca Q. Zamboni

Main reference A. E. Frid, S. Puzynina, L. Q. Zamboni, “On palindromic factorization of words,” *Advances in Applied Mathematics* 50:737–748, 2013.

URL <http://dx.doi.org/10.1016/j.aam.2013.01.002>

Given a non-periodic infinite word, is it true that for each k it contains a factor (version: a prefix) which cannot be decomposed as a concatenation of at most k palindromes? In a 2013 paper with S. Puzynina and L. Zamboni, we have proved this conjecture for the case of overlap-free words and for a wider class containing in particular the Sierpinski word. However, the general case remains open, and moreover, there is no proof even for general Sturmian words.

3.9 Order-preserving pattern matching with k mismatches

Paweł Gawrychowski (MPI für Informatik – Saarbrücken, DE)

License © Creative Commons BY 3.0 Unported license
© Paweł Gawrychowski

Joint work of Paweł Gawrychowski; Przemysław Uznanski

Main reference P. Gawrychowski, P. Uznanski, “Order-preserving pattern matching with k mismatches,” in Proc. of the 25th Annual Symp. on Combinatorial Pattern Matching (CPM’14), to appear; pre-print available as arXiv:1309.6453v2 [cs.DS].

URL <http://arxiv.org/abs/1309.6453v2>

We study a generalisation of the recently introduced order-preserving pattern matching, where instead of looking for an exact copy of the pattern, we only require that the relative order between the elements is the same. In our variant, we additionally allow up to k mismatches between the pattern and the text, and the goal is to construct an efficient algorithm for small values of k . For a pattern of length m and a text of length n , our algorithm detects an order-preserving occurrence with up to k mismatches in $\mathcal{O}(n(\log \log m + k \log \log k))$ time.

3.10 Algebraic properties of word equations

Štěpán Holub (Charles University – Prague, CZ)

License © Creative Commons BY 3.0 Unported license
© Štěpán Holub

Joint work of Štěpán Holub; Jan Žemlička

Main reference Š. Holub, J. Žemlička, “Algebraic properties of word equations,” arXiv:1403.1951v1 [cs.FL], 2014.

URL <http://arxiv.org/abs/1403.1951v1>

In [1], Aleksa Saarela has introduced a new approach to word equations that is based on linear-algebraic properties of polynomials encoding the equations and their solutions. We develop further this approach and take into account other algebraic properties of polynomials, namely their factorization.

It turns out, that a special factor of Saarela’s determinant corresponds to each length type of a solution. This, in particular, allows to improve the bound for the number of independent equations with minimal defect effect from quadratic to linear.

References

- 1 Aleksa Saarela. Systems of word equations, polynomials and linear algebra: A new approach. *CoRR*, abs/1401.7498, 2014.

3.11 Local Recompression for Word Equations

Artur Jež (MPI für Informatik – Saarbrücken, DE)

License © Creative Commons BY 3.0 Unported license
© Artur Jež

Main reference A. Jež, “Recompression: a simple and powerful technique for word equations,” in Proc. of the 30th Int’l Symp. on Theoretical Aspects of Computer Science (STACS’13), LIPIcs, Vol. 20, pp. 233–244, 2013.

URL <http://dx.doi.org/10.4230/LIPIcs.STACS.2013.233>

In this talk I will present an application of a simple technique of local recompression to word equations. The technique is based on local modification of variables (replacing X by aX or Xa) and iterative replacement of pairs of letters occurring in the equation by a ‘fresh’ letter,

which can be seen as a bottom-up compression of the solution of the given word equation, to be more specific, building an SLP (Straight-Line Programme) for the solution of the word equation. Using this technique we give a new, independent and self-contained proofs of many known results for word equations. To be more specific, the presented (nondeterministic) algorithm runs in $\mathcal{O}(n \log n)$ space and in time polynomial in n and $\log N$, where n is the size of the input equation and N the size of the length-minimal solution of the word equation.

The obtained algorithm is easy to explain and generalises to many extension of word equations: free monoids with involution, free groups, context unification.

3.12 String Range Matching

Juha Kärkkäinen (University of Helsinki, FI)

License © Creative Commons BY 3.0 Unported license
© Juha Kärkkäinen

Joint work of Juha Kärkkäinen; Dominik Kempa; Simon J. Puglisi

Main reference J. Kärkkäinen, D. Kempa, S. J. Puglisi, “String Range Matching,” in Proc. of the 25th Annual Symp. on Combinatorial Pattern Matching (CPM’14), to appear.

Given strings X and Y the exact string matching problem is to find the occurrences of Y as a substring of X . An alternative formulation asks for the lexicographically consecutive set of suffixes of X that begin with Y . We introduce a generalisation called string range matching where we want to find the suffixes of X that are in an arbitrary lexicographical range bounded by two strings Y and Z . The problem has applications in distributed suffix sorting, where Y and Z are themselves suffixes of X .

Exact string matching can be solved in linear time using constant extra space. The open question is:

► **Question 8.** *What is the time-space complexity of string range matching?*

We have described algorithms for string range matching that have an extra logarithmic factor in either the time or the space [CPM 2014].

► **Question 9.** *Are there algorithms with a better time-space complexity? Or can one show that string range matching cannot be solved in linear time and constant extra space?*

3.13 Sum of Digits of n and n^2

Steffen Kopecki (University of Western Ontario – London, CA)

License © Creative Commons BY 3.0 Unported license
© Steffen Kopecki

Joint work of Steffen Kopecki; Thomas Stoll

Main reference K. G. Hare, S. Laishram, T. Stoll, “The sum of digits of n and n^2 ,” International Journal of Number Theory, 07(7):1737–1752, 2011.

URL <http://dx.doi.org/10.1142/S1793042111004319>

I am presenting a problem that has been presented on last year’s workshop *Challenges in Combinatorics on Words* at the Fields Institute by Thomas Stoll. Since last year we made some progress in solving the problem, but despite our efforts there are still some cases left open.

For $n \in \mathbb{N}$, let $s_2(n)$ denote the sum of digits in the binary expansion of n . In other words, if $s_2(n) = k$, then n can be written as $n = 2^{r_0} + 2^{r_1} + \dots + 2^{r_{k-1}}$ for integers

$0 \leq r_0 < r_1 < \dots < r_{k-1}$. For $k \in \mathbb{N}$, we are investigating the set of positive odd integers n which satisfy the equation $s_2(n) = s_2(n^2) = k$. We let

$$\mathcal{S}_k = \{n \text{ odd} \mid s_2(n) = s_2(n^2) = k\}$$

and ask the question for which numbers $k \in \mathbb{N}$ the set \mathcal{S}_k is infinite. Our investigation is restricted to odd numbers because for every odd number n which satisfies the equation, there is an infinite family of even numbers $\{n \cdot 2^i \mid i > 0\}$ which satisfy the equation as well.

From [1] we obtain that

- for $k = 1, \dots, 8$ the set \mathcal{S}_k is finite, and
 - for $k = 12, 13$ and $k \geq 16$ the set \mathcal{S}_k is infinite.
- Furthermore, since last year's workshop, we could prove that
- for $k = 9, 10$ the set \mathcal{S}_k is finite.

► **Question 10.** *Is the set \mathcal{S}_k finite or infinite for $k = 11, 14, 15$.*

References

- 1 K. G. Hare, S. Laishram, and T. Stoll. The sum of digits of n and n^2 . *International Journal of Number Theory*, 7(07):1737–1752, 2011.

3.14 The Burrows-Wheeler Transform with Permutations


Manfred Kufleitner (Universität Stuttgart, DE)

License  Creative Commons BY 3.0 Unported license
© Manfred Kufleitner

We present a new variant of the Burrows-Wheeler Transform (BWT). It involves an action of a group G on an ordered alphabet Σ . We write a^g for the letter obtained by applying the element $g \in G$ to $a \in \Sigma$. For $u = a_1 \dots a_n$ we let $u^g = a_1^g \dots a_n^g$ be the homomorphic extension to words $u \in \Sigma^*$. Let \tilde{u} denote the lexicographically minimal element in $\{u^g \mid g \in G\}$. Let $(\tilde{v}_1, \dots, \tilde{v}_n)$ be the sorted list of the conjugates v_i of u . The BWT with permutations (BWTP) of u is $\text{BWTP}_G(u) = (w, i, g)$ where w is the sequence of the last letters in the sorted list of the words \tilde{v}_i , the number i is an index with $\tilde{u} = \tilde{v}_i$, and $g \in G$ satisfies $\tilde{u} = u^g$. It is easy to show that BWTP is injective. It would be desirable to find efficient algorithms for computing the BWTP and its inverse. Moreover, for some fixed compression algorithm, it would be interesting to identify the groups G such that $\text{BWTP}_G(u)$ compresses best; this could help in revealing hidden symmetries of u .

3.15 Text Indexing: Easy and Difficult

Moshe Lewenstein (Bar-Ilan University – Haifa, IL)

License  Creative Commons BY 3.0 Unported license
© Moshe Lewenstein

Joint work of Amihod Amir; Timothy Chan; Moshe Lewenstein; Noa Lewenstein

Text indexing refers to the problem of preprocessing a text for future queries. The goal is to construct a data structure quickly in minimum space in order to answer queries quickly.

For exact match queries data structures, such as suffix trees, suffix arrays, and others, are well known to be constructible in linear time and space for later linear (pattern length) queries.

We first show several examples where this is still the case for extended query definitions. We then show an interesting separation between the definition of sum-queries and product-queries. For small constant sized alphabet $1, \dots, c$ sum-queries is solvable efficiently. On the other hand, we show that product-queries, under the 3SUM-Hardness assumption, need either $\mathcal{O}(n^2)$ preprocessing time or $\mathcal{O}(n)$ query time.

This has consequences for the problem of histogram (or jumbled) indexing which has garnered much interest lately.

3.16 Testing k -binomial equivalence

Florin Manea (Universität Kiel, DE)

License © Creative Commons BY 3.0 Unported license
© Florin Manea

Joint work of Dominik Freydenberger; Paweł Gawrychowski; Juhani Karhumäki; Manfred Kufleitner; Florin Manea; Wojciech Rytter

The binomial coefficient of two words u and v is the number of times v occurs as a scattered factor of u , and it is denoted as $\binom{u}{v}$. Two words u and w over an alphabet Σ are k -binomial equivalent if $\binom{u}{v} = \binom{w}{v}$ for all words $v \in \Sigma^{\leq k}$. In this setting, it seems interesting to show that one can decide in polynomial time for a pair of words u and w and a number k whether u and w are k -binomial equivalent. As a first result, Paweł Gawrychowski showed that the problem can be solved efficiently by a polynomial Monte-Carlo algorithm in the logarithmic word-size RAM model. Then, Juhani Karhumäki and Wojciech Rytter noted that the problem can be reduced to the problem of testing whether two nondeterministic finite automata without λ -transitions are path equivalent. It is known that this problem can be solved in polynomial time on a unit-cost RAM model. Further discussions involving Dominik Freydenberger and Manfred Kufleitner led to a final solution of this problem, concluding that in fact the problem can be solved in polynomial time in the logarithmic word-size RAM model. As an open problem, we ask the following:

► **Question 11.** *What is the complexity of finding, for two words w and u and a number k all the factors of w that are k -binomial equivalent to u . Can this problem be solved more efficiently than just checking whether each factor of w is k -binomial equivalent to u ?*

3.17 k -Abelian Pattern Matching

Robert Mercaş (Universität Kiel, DE)

License © Creative Commons BY 3.0 Unported license
© Robert Mercaş

Joint work of Thorsten Ehlers; Florin Manea; Robert Mercaş; Dirk Nowotka

Main reference T. Ehlers, F. Manea, R. Mercaş, D. Nowotka, “ k -Abelian Pattern Matching,” in Proc. of the 18th Int’l Conf. on Developments in Language Theory (DLT’14), to appear.

Two words are called k -abelian equivalent, if they share the same multiplicities for all factors of length at most k . We present an optimal linear time algorithm for identifying all occurrences of factors in a text that are k -abelian equivalent to some pattern P . Moreover,


an optimal algorithm for finding the largest k for which two words are k -abelian equivalent is given. The complexity of algorithms for online versions of the k -abelian pattern matching problem is also considered. In particular we show results regarding the investigation of the pattern matching problem for k -abelian equivalences in the setting of online algorithms, and propose a series of real-time solutions of this problem. One of the questions we propose is to:

► **Question 12.** *Identify an optimal linear time complexity algorithm for the pattern matching problem for k -abelian equivalences.*

We also show results for an extended form of k -abelian equivalence.

3.18 Bell numbers modulo 8

Eric Rowland (University of Liège, BE)

License  Creative Commons BY 3.0 Unported license
© Eric Rowland

The n th Bell number $B(n)$ is the number of partitions of an n -element set. The sequence $B(n)_{n \geq 0}$ is 1, 1, 2, 5, 15, 52, 203, 877, . . .

► **Question 13.** *Is it true that $B(n)$ is not divisible by 8 for all $n \geq 0$?*

Experiments suggest that $(B(n) \bmod 8)_{n \geq 0}$ is a 2-automatic sequence, meaning that there is a deterministic finite automaton with output that outputs $B(n) \bmod 8$ when fed the standard base-2 representation of n . Recently, Yassawi and I [2] showed how to automatically compute automata for many sequences modulo prime powers, thereby giving such congruences purely mechanically. However, the sequence of Bell numbers appears to not be accessible by this method.

During the workshop, Mike Müller found a paper of Lunnon, Pleasants, and Stephens [1] which shows that $(B(n) \bmod p^\alpha)_{n \geq 0}$ is in fact periodic. Modulo 8, the sequence of Bell numbers has period 24. Computing the first 24 terms then gives a proof that no Bell number is divisible by 8. Also, no Bell number is congruent to 6 modulo 8. A comment in the OEIS entry for the Bell numbers, <https://oeis.org/A000110>, referencing the Lunnon–Pleasants–Stephens paper has been clarified with the proper theorem. Steffen Kopecki found an independent proof, using a Pascal-like triangle for the Bell numbers.

References

- 1 W. F. Lunnon, P. A. B. Pleasants and N. M. Stephens, Arithmetic properties of Bell numbers to a composite modulus I, *Acta Arithmetica* **35** (1979) 1–16.
- 2 Eric Rowland and Reem Yassawi, Automatic congruences for diagonals of rational functions, to appear in *Journal de Théorie des Nombres de Bordeaux*, available from <http://arxiv.org/abs/1310.8635>.

3.19 Maximum Number of Distinct and Nonequivalent Nonstandard Squares in a Word

Wojciech Rytter (*University of Warsaw, PL*)

License © Creative Commons BY 3.0 Unported license

© Wojciech Rytter

Joint work of Tomasz Kociumaka; Jakub Radoszewski; Wojciech Rytter; Tomasz Waleń

Main reference T. Kociumaka, J. Radoszewski, W. Rytter, T. Waleń, “Maximum Number of Distinct and Nonequivalent Nonstandard Squares in a Word,” in Proc. of the 18th Int’l Conf. on Developments in Language Theory (DLT’14), to appear.

The combinatorics of squares in a word depends on how the equivalence of halves of the square is defined. We consider Abelian squares, parameterized and order-preserving squares. The word uv is an Abelian (parameterized, order-preserving) square if u and v are equivalent in the Abelian (parameterized, order-preserving) sense. The maximum number of ordinary squares is known to be asymptotically linear, but the exact bound is still investigated.

We present several results on the maximum number of distinct squares for nonstandard subword equivalence relations. Let $SQ_{Abel}(n, k)$ and $SQ'_{Abel}(n, k)$ denote the maximum number of Abelian squares in a word of length n over alphabet of size k , which are distinct as words and which are nonequivalent in the Abelian sense, respectively.

We prove that

$$SQ_{Abel}(n, 2) = \Theta(n^2), \quad SQ'_{Abel}(n, 2) = \Omega(n^{1.5}/\log n).$$

We also give linear bounds for parameterized and order-preserving squares for small alphabets:

$$SQ_{param}(n, 2) = \Theta(n), \quad SQ_{op}(n, O(1)) = \Theta(n).$$

As a side result we construct infinite words over the smallest alphabet which avoid nontrivial order-preserving squares and nontrivial parameterized cubes (nontrivial parameterized squares cannot be avoided in an infinite word).

3.20 Parametric solutions of word equations

Aleksi Saarela (*University of Turku, FI*)

License © Creative Commons BY 3.0 Unported license

© Aleksi Saarela

By Hmelevskii’s theorem [1], every constant-free word equation on three unknowns has a parametric solution. In [3], an exponential upper bound was proved for the length of such a parametric solution.

► **Question 14.** *How many parametric formulas do we need in such a solution, at most?*

The best known lower bound for the number of formulas is three: The equation $xyxzyz = zxzyxy$ has a parametric solution

$$(x, y, z) = (p, q, \varepsilon), \quad (x, y, z) = (p, q, pq), \quad (x, y, z) = (p^i, p^j, p^k),$$

but it can be proved that it does not have a parametric solution with just two formulas. As another example, consider the equation $xy = z^n$. It has a parametric solution with $\lceil n/2 \rceil$ formulas, but it is not known whether this is optimal.

The above-mentioned equation $xyxzyz = zxzyxy$ is also related to the following big open problem:

► **Question 15.** *How long sequences E_1, \dots, E_n of non-trivial word equations on three unknowns do we have such that the systems E_1, \dots, E_i ($i = 1, \dots, n$) are pairwise non-equivalent and have non-periodic solutions?*

The best known example is the sequence $xyz = zxy, xyxzyz = zxzyxy, xz = zx$. For more information, see [2].

References

- 1 Ju. I. Hmelevskii. *Equations in free semigroups*. American Mathematical Society, 1976. Translated by G. A. Kandall from the Russian original: Trudy Mat. Inst. Steklov. 107 (1971).
- 2 Juhani Karhumäki and Aleksi Saarela. On maximal chains of systems of word equations. *Proc. Steklov Inst. Math.*, 274:116–123, 2011.
- 3 Aleksi Saarela. On the complexity of Hmelevskii’s theorem and satisfiability of three unknown equations. In *Proceedings of the 13th DLT*, volume 5583 of *LNCS*, pages 443–453. Springer, 2009.

3.21 Efficient generation of repetition-free words

Arseny M. Shur (Ural Federal University – Ekaterinburg, RU)

License  Creative Commons BY 3.0 Unported license
© Arseny M. Shur

When some repetition is proved to be avoidable over some alphabet, we usually get an explicit construction of infinite repetition-free words. Normally, such constructions are based on substitutions satisfying certain restrictions (in the simplest case, just on morphisms). As a result, the obtained words have some “additional” properties, like ultimate recurrence, which do not follow from repetition-freeness. Hence it is quite useful to have a generator which can produce any word from a given repetition-free language. Such generators can rely on the general “local resampling” idea used by Moser and Tardos for the constructive proof of the Lovasz Local Lemma [2].

An algorithm for square-like repetitions was first proposed by Grytczuk, Kozik, and Witkowski [1] and reformulated for squares by Rampersad. We modified this algorithm to convert random words over $\Sigma_k = \{1, \dots, k\}$ to square-free words over Σ_{k+1} ; this can be done more efficiently than the conversion over the same alphabet. Without falling into implementation details, our algorithm works as follows. On each step, one letter is appended to the right end of the square-free word under construction. If the resulting word ends with a square, then the right half of this square is dismissed, otherwise we just proceed to the next step. To get the letter for appending, we take a random letter over Σ_k , say i , sort the letters of Σ_{k+1} by the recency of their last occurrence in the square-free word, and take the $(i+1)$ th element of the sorted list.

We proved the following

► **Theorem 1.** *The expected number of random k -ary letters used by the above algorithm to construct a $(k+1)$ -ary square-free word of length n is*

$$N = n(1 + 2/k^2 + 1/k^3 + 4/k^4 + \mathcal{O}(1/k^5)) + \mathcal{O}(1).$$

Thus, if k is not small, then the algorithm converts random words to square-free words of nearly the same length. However, for the extremal case of ternary square-free words we have

no theoretical bound on the conversion rate. From experiments we learned that the expected value of N is linear in n in this case and, moreover, $N \approx 12.5n$. The following problems naturally develop the obtained results.

► **Question 16.** *Give an upper bound on the conversion ratio of the above algorithm for the case of ternary square-free words.*

► **Question 17.** *Give efficient algorithms generating cube-free words; words, avoiding fractional powers; Abelian square-free words.*

References

- 1 J. Grytczuk, J. Kozik, and M. Witkowski. Nonrepetitive sequences on arithmetic progressions. *Electronic J. Combinatorics*, 18(1):# P209, 2011.
- 2 R. A. Moser and G. Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM*, 57:11:1–11:15, 2010.

Participants

- Dany Breslauer
University of Haifa, IL
- Julien Cassaigne
CNRS – Marseille, FR
- Julien Clément
Caen University, FR
- Maxime Crochemore
King’s College London, GB
- James D. Currie
University of Winnipeg, CA
- Volker Diekert
Universität Stuttgart, DE
- Gabriele Fici
University of Palermo, IT
- Johannes Fischer
TU Dortmund, DE
- Dominik D. Freydenberger
Goethe-Universität Frankfurt am
Main, DE
- Anna E. Frid
Aix-Marseille University, FR
- Paweł Gawrychowski
MPI für Informatik –
Saarbrücken, DE
- Amy Glen
Murdoch University, AU
- Štěpán Holub
Charles University – Prague, CZ
- Artur Jež
MPI für Informatik –
Saarbrücken, DE
- Juha Kärkkäinen
University of Helsinki, FI
- Juhani Karhumäki
University of Turku, FI
- Steffen Kopecki
University of Western Ontario –
London, CA
- Gregory Kucherov
University Paris-Est –
Marne-la-Vallée, FR
- Manfred Kufleitner
Universität Stuttgart, DE
- Gad M. Landau
University of Haifa, IL
- Alessio Langiu
King’s College London, GB &
University of Palermo, IT
- Thierry Lecroq
University of Rouen, FR
- Moshe Lewenstein
Bar-Ilan University, IL
- Florin Manea
Universität Kiel, DE
- Giancarlo Mauri
University of Milan-Bicocca, IT
- Robert Mercas
Universität Kiel, DE
- Fillippo Mignosi
University of L’Aquila, IT
- Mike Müller
Universität Kiel, DE
- Dirk Nowotka
Universität Kiel, DE
- Wojciech Plandowski
University of Warsaw, PL
- Ely Porat
Bar-Ilan University, IL
- Svetlana Puzynina
University of Turku, FI
- Antonio Restivo
University of Palermo, IT
- Eric Rowland
University of Liège, BE
- Wojciech Rytter
University of Warsaw, PL
- Aleksi Saarela
University of Turku, FI
- Arseny M. Shur
Ural Federal University –
Ekaterinburg, RU
- Jamie Simpson
Curtin University of Technology –
Perth, AU
- German Tischler
Wellcome Trust Sanger Institute –
Hinxton, GB
- Esko Ukkonen
University of Helsinki, FI
- Mikhail V. Volkov
Ural Federal University –
Ekaterinburg, RU

