CAI 2017

7th Conference on Algebraic Informatics

June 25-28, 2017
Elite City Resort, Kalamata, Greece
http://www.cargo.wlu.ca/CAI2017

Tracks

Track 1: Automata Theory and Logic
Chair: **Manfred Droste** (Universität Leipzig, Germany)
Invited Speaker: **Heiko Vogler** (TU Dresden, Germany)

Track 2: Cryptography and Coding Theory
Chair: **Stephane Ballet** (Université d'Aix-Marseille, France),
Dimitrios Poulakis (Aristotle University of Thessaloniki, Greece),
Robert Rolland (Aix-Marseille Université, France)
Invited Speaker: **Claude Carlet** (Université Paris 8, France)

Track 3: Computer Algebra
Chair: **Rafael Sendra** (Universidad de Alcala, Spain),
Franz Winkler (RISC-Linz, Austria)
Invited Speaker: **Michael Wibmer** (University of Pennsylvania, USA)

Track 4: Design Theory
Chair: **Lucia Moura** (University of Ottawa, Canada),
Dimitris Simos (SBA Research, Austria)
Invited Speaker: **Charles Colbourn** (Arizona State University, USA)

Track 5: Natural and Quantum Computing
Chair: **Mika Hirvensalo** (University of Turku, Finland)
Invited Speaker: **Lila Kari** (University of Waterloo, Canada)

General Chair:
Ilias Kotsireas
Wilfrid Laurier University, Canada

Editors: (CAI Book of Abstracts)
Scott King and **Ilias Kotsireas**

Publicity Chairs:
Ioannis Haranas
Wilfrid Laurier University, Canada

Dragomir Stanojevic
MedCurrent Corporation, Canada

Computer Algebra Research Group of Wilfrid Laurier University

SBA Research

Aix*Marseille université

Turun yliopisto
University of Turku

Universidad de Alcalá

uOttawa

UNIVERSITÄT LEIPZIG

Table of Contents

Track 1: Automata Theory and Logic

Chair: Manfred Droste (Germany)

Invited Speaker: Heiko Vogler

Parsing of Natural Languages

Technische UniversitÃt Dresden (Faculty of Computer Science)

Abstract

The syntax of natural languages copes with at least three facets: the sequencing of words (in German: kleines Brot, in French: pain petit), the constituency structure (like subject - predicate - object), and dependencies (governor: house, dependent: cosy). In this talk we will focus on the latter two facets.

We define the concept of hybrid tree, which captures both, phrase structure trees and dependency trees. Formally, a hybrid tree is a tree together with a linear order on a subset of the set of its positions. For a phrase structure tree the subset is the set of its leaves, for a dependency tree the subset is the set of all positions of the tree.

We discuss formal grammars which generate hybrid tree languages, indicate algorithms how to induce the grammars automatically from corpora, and motivate a new grammar model – the hybrid grammars.

Languages and formations generated by D_4 and Q_8[1]

Jean-Éric Pin[1], Xaro Soler-Escrivà[2]
[1] *IRIF, CNRS and Université Paris-Diderot, Case 7014, 75205 Paris Cedex 13, France. Jean-Eric.Pin@irif.fr*
[2] *Dpt. de Matemàtiques, Universitat d'Alacant, Sant Vicent del Raspeig, Ap. Correus 99, E – 03080 Alacant. xaro.soler@ua.es*

Abstract
We describe the two classes of languages recognized by the groups D_4 and Q_8, respectively. Then we show that the formations of languages generated by these two classes are the same. We also prove that these two formations are closed under inverses of morphisms, which yields a language theoretic proof of the fact that the group formations generated by D_4 and Q_8, respectively, are two equal varieties.

Most monoids and groups considered in this paper are finite. In particular, we use the term *variety of groups* for *variety of finite groups*. Similarly, all languages considered in this paper are regular languages and hence their syntactic monoid is finite.

1 Introduction

A nontrivial question is to describe the regular languages corresponding to well-studied families of finite groups. Only a few cases have been investigated in the literature: abelian groups [6], p-groups [6, 16, 17, 18], nilpotent groups [6, 15], soluble groups [14, 17] and supersoluble groups [4]. More recently [2], the authors addressed the following question: is it possible to obtain a reasonable description of the languages corresponding to a given formation of groups? Recall that a *formation of groups* is a class of finite groups closed under taking quotients and subdirect products.

This question was motivated by the importance of formations in finite group theory, notably in the development of a generalised Sylow theory. The theory of formations was born with the seminal paper [7] of Gaschütz in 1963, where a broad extension of Sylow and Hall theory was presented. The new theory was not an arithmetic one, that is, based on the orders of subgroups. Instead, the important idea was concerned with group classes having the same properties. In that way, the formations of groups appeared and since that time they have played a fundamental role in the study of groups [5, 1].

[1] The first author is supported by Proyecto MTM2014-54707-C3-1-P from MINECO (Spain) and FEDER (European Union) and partially funded from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 670624) and by the DeLTA project (ANR-16-CE40-0007).

In [2], the authors extended Eilenberg's correspondence theorem between varieties of monoids and varieties of languages [6] to the setting of formations. More precisely, they spotted a bijective correspondence between formations of finite monoids and the so-called *formations of languages*. Using this "formation theorem" the authors not only recovered the previously mentioned results on nilpotent groups, soluble groups and supersoluble groups, but, relying on the local definition of a saturated formation [5], they exhibited new examples, like the class of groups having a Sylow tower [3].

This new paper focuses on the language interpretation of two results dealing with the dihedral group D_4 and the quaternion group Q_8. The first result asserts that D_4 and Q_8 generate the same formation [5, Exercise 9, p. 344]. The second one states that this formation is a variety of groups, that is, is closed under taking subgroups. This latter result is actually an instance of a more general result, due to Neumann [9], which states that any formation generated by a single nilpotent group is a variety (see [5, IV.1.16, p. 342] for an alternative proof).

The main result of this paper provides a purely language theoretic proof of these two results on D_4 and Q_8. To do so, we first translate them in terms of languages: the formations of languages $\mathscr{F}_1$ and $\mathscr{F}_2$ associated to D_4 and Q_8, respectively, are the same (first result) and they form a variety of languages (second result). The main difficulty in proving these results by pure language theoretic means is to establish the inclusion $\mathscr{F}_1 \subseteq \mathscr{F}_2$. The lengthy proof of Theorem 2 should convince the reader that it is a nontrivial property.

Our proofs rely on a systematic use of the binomial coefficients of two words. This is not really a surprise, since binomial coefficients modulo p are the main tool for describing languages recognized by p-groups, and D_4 and Q_8 are 2-groups. In this paper, we present two explicit formulas with an algorithmic flavour. First, we discuss the behaviour of binomial coefficients under morphisms (Formula 5). Next, we show that a language of A^* is recognized by a p-group if and only if it is a finite union of languages defined by linear algebraic constraints involving the binomial coefficients. Finally, we give an algorithm to obtain such a decomposition when the p-group is a group of unitriangular matrices over $\mathbb{F}_p$.

Our paper is organised as follows. In order to keep the paper self-contained, prerequisites (Section 2) include formations and varieties, syntactic monoids and the Formation Theorem. Section 3 is devoted to binomial coefficients on words. We present in Section 4 various descriptions of the languages recognized by p-groups and the corresponding algorithms. Section 5 contains the proof of our main theorem.

2 Prerequisites

2.1 Formations and varieties

A *formation of groups* is a class of groups **F** satisfying the two conditions:

(1) any quotient of a group of **F** also belongs to **F**,

(2) the subdirect product of any finite family of groups of **F** is also in **F**.

Formations of finite algebras can be defined in the same way [11, 13, 12]. In particular, a *formation of monoids* is a class of finite monoids closed under taking quotients and subdirect products.

A *variety of groups* is a class of groups **V** satisfying the three conditions:

(1) any subgroup of a group of **V** also belongs to **V**,

(2) any quotient of a group of **V** also belongs to **V**,

(3) the direct product of any finite family of groups of **V** is also in **V**.

Varieties of monoids are defined in the same way. It follows from the definition that a formation of groups [monoids] is a variety if and only if it is closed under taking subgroups [submonoids]. Therefore a formation is not necessarily a variety. For instance, the formation of groups generated by the alternating group A_5 is known to be the class of all direct products of copies of A_5, which is not a variety [1, Lemma 2.2.3, p. 91], [5, II.2.13].

2.2 Syntactic monoids

Let L be a regular language and let x and y be words. The *quotient* $x^{-1}Ly^{-1}$ of L by x and y is defined by the formula

$$x^{-1}Ly^{-1} = \{u \in A^* \mid xuy \in L\}$$

The *syntactic monoid* of a regular language L of A^* is the finite monoid obtained as the quotient of A^* by the *syntactic congruence* of L, defined on A^* as follows: $u \sim_L v$ if and only if, for every $x, y \in A^*$,

$$xvy \in L \iff xuy \in L$$

The natural morphism $\eta : A^* \to A^*/\sim_L$ is the *syntactic morphism* of L.

A *class* of regular languages $\mathscr{C}$ associates with each finite alphabet A a set $\mathscr{C}(A^*)$ of regular languages of A^*. It is *closed under quotients* if for each language $L \in \mathscr{C}(A^*)$ and for each pair of words (x, y) of A^*, the language $x^{-1}Ly^{-1}$ belongs to $\mathscr{C}$.

2.3 The Formation Theorem

Just as formations of finite monoids extend the notion of a variety of finite monoids, formations of languages are more general than varieties of languages. Like varieties, formations are classes of regular languages closed under Boolean operations and quotients. But while varieties are closed under inverse of morphisms, formations of languages only enjoy a weak version of this property — Property (F_2) — and thus comprise more general classes of languages than varieties.

The following definition was first given in [2]. A *formation of languages* is a class of regular languages $\mathscr{F}$ satisfying the following conditions:

- (F_1) for each alphabet A, $\mathscr{F}(A^*)$ is closed under Boolean operations and quotients,
- (F_2) if L is a language of $\mathscr{F}(B^*)$ and $\eta : B^* \to M$ denotes its syntactic morphism, then for each monoid morphism $\alpha : A^* \to B^*$ such that $\eta \circ \alpha$ is surjective, the language $\alpha^{-1}(L)$ belongs to $\mathscr{F}(A^*)$.

Observe that a formation of languages is closed under inverse of surjective morphisms, but this condition is not equivalent to (F_2).

To each formation of monoids $\mathbf{F}$, let us associate the class of languages $\mathscr{F}(\mathbf{F})$ defined as follows: for each alphabet A, $\mathscr{F}(\mathbf{F})(A^*)$ is the set of languages of A^* whose syntactic monoid belongs to $\mathbf{F}$.

Given a formation of languages $\mathscr{F}$, let $\mathbf{F}(\mathscr{F})$ denote the formation of monoids generated by the syntactic monoids of the languages of $\mathscr{F}$. The following statement is the main result of [2].

Theorem 1 (Formation Theorem). *The correspondences $\mathbf{F} \to \mathscr{F}(\mathbf{F})$ and $\mathscr{F} \to \mathbf{F}(\mathscr{F})$ are two mutually inverse, order preserving, bijections between formations of monoids and formations of languages.*

3 Binomial coefficients on words

Binomial coefficients on words were first defined in [6, p. 238]. Useful references include [8, Chapter 6] and [10].

3.1 Definition of binomial coefficients on words

A word $u = a_1 a_2 \cdots a_n$ (where $a_1, \ldots, a_n$ are letters) is a *subword* of a word v if v can be factored as $v = v_0 a_1 v_1 \cdots a_n v_n$. For instance, *ab* is a subword of *cacbc*. Given two words u and v, we denote by $\binom{v}{u}$ the number of distinct ways to write u as a subword of v.

More formally, if $u = a_1 a_2 \cdots a_n$, then

$$\binom{v}{u} = \mathrm{Card}\{(v_0, v_1, \ldots, v_n) \mid v_0 a_1 v_1 \cdots a_n v_n = v\}$$

Observe that if u is a letter a, then $\binom{v}{a}$ is simply the number of occurrences of the letter a in v, also denoted by $|v|_a$. These binomial coefficients satisfy the following recursive formula, where $u, v \in A^*$ and $a, b \in A$:

$$\begin{cases} \binom{u}{1} = 1 \\ \binom{1}{u} = 0 \text{ if } u \neq 1 \\ \binom{va}{ub} = \begin{cases} \binom{v}{ub} & \text{if } a \neq b \\ \binom{v}{ub} + \binom{v}{u} & \text{if } a = b \end{cases} \end{cases} \tag{1}$$

We shall later use the following elementary result.

Proposition 1. *Let $u \in \{a,b\}^*$. Then the following formula holds*

$$\binom{u}{a}\binom{u}{b} + \binom{u}{ab} + \binom{u}{ba} \equiv 0 \bmod 2 \tag{2}$$

Proof. Let us prove (2) by induction on $|u|$. The result is trivial if $|u| = 0$. For the induction step, it suffices to prove the result for ua, the case ub being symmetrical.

$$\begin{aligned} \binom{ua}{a}\binom{ua}{b} + \binom{ua}{ab} + \binom{ua}{ba} &= \left(\binom{u}{a} + 1\right)\binom{u}{b} + \binom{u}{ab} + \binom{u}{ba} + \binom{u}{b} \\ &\equiv \binom{u}{a}\binom{u}{b} + \binom{u}{ab} + \binom{u}{ba} \equiv 0 \bmod 2. \end{aligned}$$

3.2 Binomial coefficients and morphisms

Let $\mathbb{Z}\langle A\rangle$ be the ring of noncommutative polynomials with coefficients in $\mathbb{Z}$ and variables in A (see [8, Chapter 6] or [10]). Given a polynomial $P \in \mathbb{Z}\langle A\rangle$ and a word x, we let $\langle P, x\rangle$ denote the coefficient of P in x. Thus all but a finite number of these coefficients are null and $P = \sum_{x \in A^*} \langle P, x\rangle x$.

In this section, we study the behaviour of binomial coefficients under monoid morphisms. More precisely, given a monoid morphism $\varphi : A^* \to B^*$ and words $u \in A^*$ and $x \in B^*$, we give a formula to compute $\binom{\varphi(u)}{x}$.

The proof of this result relies on properties of the *Magnus automorphism* of the ring $\mathbb{Z}\langle A\rangle$. This automorphism μ_A is defined, for each letter $a \in A$, by $\mu_A(a) = 1 + a$. Its inverse is defined by $\mu_A^{-1}(a) = a - 1$. The following *binomial identity* [8, Formula 6.3.4]

$$\text{for all } u \in A^*, \quad \mu_A(u) = \sum_{x \in A^*} \binom{u}{x} x \tag{3}$$

can be used to give an alternative definition of the binomial coefficients.

If $\varphi : A^* \to B^*$ be a monoid morphism, then φ can be extended by linearity to a ring morphism from $\mathbb{Z}\langle A\rangle$ to $\mathbb{Z}\langle B\rangle$. Let $\gamma : \mathbb{Z}\langle A\rangle \to \mathbb{Z}\langle B\rangle$ be the ring morphism defined by $\gamma = \mu_B \circ \varphi \circ \mu_A^{-1}$.

Then for each $s \in A^*$, $\gamma(s)$ is a polynomial of $\mathbb{Z}\langle B\rangle$.

$$\gamma(s) = \sum_{x \in B^*} \langle \gamma(s), x\rangle x \tag{4}$$

We are now ready to present the announced formula:

Proposition 2. *If $\varphi : A^* \to B^*$ is a morphism, then*

$$\binom{\varphi(u)}{x} = \sum_{s \in A^*} \binom{u}{s} \langle \gamma(s), x\rangle = \sum_{|s| \leqslant |x|} \binom{u}{s} \langle \gamma(s), x\rangle \tag{5}$$

Proof. Observing that $\mu_A^{-1}(a) = a - 1$ for each letter $a \in A$, one gets

$$\gamma(a) = \mu_B(\varphi(a) - 1) = \mu_B(\varphi(a)) - 1 = \Big(\sum_{x \in B^*} \binom{\varphi(a)}{x} x \Big) - 1 = \sum_{x \in B^+} \binom{\varphi(a)}{x} x$$

and thus $\langle \gamma(a), 1\rangle = 0$. It follows that $\langle \gamma(s), x\rangle = 0$ if $|x| < |s|$. Furthermore, for each $u \in A^*$, one gets on the one hand from (3)

$$\mu_B(\varphi(u)) = \sum_{x \in B^*} \binom{\varphi(u)}{x} x$$

and on the other hand, using (3) and (4)

$$\gamma(\mu_A(u)) = \gamma\Big(\sum_{s \in A^*} \binom{u}{s} s \Big) = \sum_{s \in A^*} \binom{u}{s} \gamma(s) = \sum_{s \in A^*} \sum_{x \in B^*} \binom{u}{s} \langle \gamma(s), x\rangle x$$

Now since $\gamma \circ \mu_A = \mu_B \circ \varphi$, the polynomials $\mu_B(\varphi(u))$ and $\gamma(\mu_A(u))$ have the same coefficients, which gives (5).

Example 1. To illustrate the use of (5), let us show how to compute $\binom{\varphi(u)}{ab}$. Let $A = \{a, b, c\}$, $B = \{a, b\}$ and let $\varphi : A^* \to B^*$ be the morphism defined by $\varphi(a) = a$, $\varphi(b) = ab$ and $\varphi(c) = a^2 b$. First, $\gamma = \mu_B \circ \varphi \circ \mu_A^{-1}$ is defined as follows:

$$\begin{aligned}
\gamma(a) &= \mu_B(\varphi(a-1)) = \mu_B(a-1) = a \\
\gamma(b) &= \mu_B(\varphi(b-1)) = \mu_B(ab-1) = (1+a)(1+b) - 1 = a + b + ab \\
\gamma(c) &= \mu_B(\varphi(c-1)) = \mu_B(a^2 b - 1) = \mu_B(a^2 b) - 1 \\
&= (1+a)(1+a)(1+b) - 1 = 2a + aa + b + 2ab + aab
\end{aligned}$$

Thus we get by (5)

$$\binom{\varphi(u)}{ab} = \sum_{s \in A^*} \binom{u}{s} \langle \gamma(s), ab\rangle = \sum_{|s| \leqslant 2} \binom{u}{s} \langle \gamma(s), ab\rangle$$

We now need to compute the coefficients $\langle \gamma(s), ab \rangle$ for $|s| \leqslant 2$. The non-zero coefficients are the following:

$$\begin{array}{llll} \langle \gamma(b), ab \rangle = 1 & \langle \gamma(c), ab \rangle = 2 & \langle \gamma(ab), ab \rangle = 1 & \langle \gamma(ac), ab \rangle = 1 \\ \langle \gamma(bb), ab \rangle = 1 & \langle \gamma(bc), ab \rangle = 1 & \langle \gamma(cb), ab \rangle = 2 & \langle \gamma(cc), ab \rangle = 2 \end{array}$$

and finally

$$\binom{\varphi(u)}{ab} = \binom{u}{b} + 2\binom{u}{c} + \binom{u}{ab} + \binom{u}{ac} + \binom{u}{bb} + \binom{u}{bc} + 2\binom{u}{cb} + 2\binom{u}{cc}.$$

4 Languages recognized by p-groups

Let p be a prime number. A *p-group* is a group whose order is a power of p. A *p-group language* is a language whose syntactic monoid is a p-group.

4.1 Two descriptions of the p-group languages

The following result is credited to Eilenberg and Schützenberger in [6].

Proposition 3. *A language of A^* is a p-group language if and only if it is a Boolean combination of languages of the form*

$$L(x, r, p) = \{u \in A^* \mid \binom{u}{x} \equiv r \bmod p\}, \tag{6}$$

where $0 \leqslant r < p$ and $x \in A^*$.

We now give another characterization. A function $f : A^* \to \mathbb{Z}$ is said to be a *linear combination of binomial coefficients* if there exist $c_1, \ldots, c_n \in \mathbb{Z}$ and $x_1, \ldots, x_n \in A^*$ such that, for all $u \in A^*$,

$$f(u) = c_1\binom{u}{x_1} + \cdots + c_n\binom{u}{x_n} \tag{7}$$

Since the function $f(u) = c\binom{u}{1}$ maps every word to the constant c, every constant function is a linear combination of binomial coefficients.

Proposition 4. *A language of A^* is a p-group language if and only if it is a finite union of languages of the form*

$$L(f_1, \ldots, f_r, p) = \{u \in A^* \mid f_1(u) \equiv \cdots \equiv f_r(u) \equiv 0 \bmod p\} \tag{8}$$

where $f_1, \ldots, f_r$ are linear combinations of binomial coefficients.

Proof. Let $\mathscr{G}_p$ be the Boolean algebra generated by the languages of the form $L(x,r,p)$ and let $\mathscr{S}_p$ be the set of languages that are finite unions of languages of the form $L(f_1,\ldots,f_r,p)$.

Step 1. $\mathscr{S}_p$ is a Boolean algebra. First, $\mathscr{S}_p$ is closed under union by definition. It is also closed under intersection since

$$L(f_1,\ldots,f_r,p)\cap L(g_1,\ldots,g_s,p) = L(f_1,\ldots,f_r,g_1,\ldots,g_s,p). \tag{9}$$

In particular,

$$L(f_1,\ldots,f_r,p) = L(f_1,p)\cap\cdots\cap L(f_r,p). \tag{10}$$

It remains to show that $\mathscr{S}_p$ is closed under complementation. Since $\mathscr{S}_p$ is closed under union and intersection, it suffices to prove that the complement of each language of the form $L(f,p)$, where f is a linear combination of binomial coefficients, belongs $\mathscr{S}_p$. Now

$$\begin{aligned} L(f,p)^c &= \{u\in A^* \mid f(u)\not\equiv 0 \bmod p\} \\ &= \bigcup_{c\in\mathbb{F}_p\setminus\{0\}} \{u\in A^* \mid f(u)\equiv c \bmod p\} \\ &= \bigcup_{c\in\mathbb{F}_p\setminus\{0\}} \{u\in A^* \mid (f-c)(u)\equiv 0 \bmod p\} \end{aligned}$$

It remains to observe that $f-c$ is a linear combination of binomial coefficients to conclude.

Step 2: $\mathscr{S}_p\subseteq\mathscr{G}_p$. It suffices to show that every language of the form $L(f,p)$ belongs to $\mathscr{G}_p$. Now if f is given by (7), one gets

$$L(f,p) = \bigcup_{\{(r_1,\ldots,r_n)\mid c_1r_1+\cdots+c_nr_n\equiv 0 \bmod p\}} \big(L(x_1,r_1,p)\cap\cdots\cap L(x_n,r_n,p)\big) \tag{11}$$

and thus $L(f,p)\in\mathscr{G}_p$ as required. Thus $\mathscr{S}_p\subseteq\mathscr{G}_p$.

Step 3: $\mathscr{G}_p\subseteq\mathscr{S}_p$. This immediately follows from the formula

$$L(x,r,p) = L(f,p) \text{ where } f(u) = -r\binom{u}{1} + \binom{u}{x}.$$

Thus $\mathscr{G}_p=\mathscr{S}_p$ and it now suffices to apply Proposition 3 to conclude the proof.

Note that one can compute the minimal automaton of a language of the form $L(f_1,\ldots,f_r,p)$ by computing its derivatives as follows:

$$u^{-1}L = \{x\in A^* \mid f_1(ux) = f_2(ux) = \cdots = f_n(ux)\equiv 0 \bmod p\}.$$

4.2 An algorithm for p-group languages

Let p be a prime number and let $U_n(\mathbb{F}_p)$ be the group of unitriangular[2] $n \times n$-matrices with coefficients in $\mathbb{F}_p$, the finite field of order p. Then $U_n(\mathbb{F}_p)$ is a p-group and it is a well-known fact that every p-group is isomorphic to a subgroup of some $U_n(\mathbb{F}_p)$, for a suitable choice of n.

Let $\pi : A \to U_{n+1}(\mathbb{F}_p)$ be a map[3] and let G be the subgroup of $U_{n+1}(\mathbb{F}_p)$ generated by $\pi(A)$. Then π extends to a surjective monoid morphism $\pi : A^* \to G$ which maps every word $a_1 \cdots a_k \in A^*$ to the matrix $\pi(a_1) \cdots \pi(a_k)$. For $1 \leqslant i < j \leqslant n+1$, we let $\pi_{i,j} : A^* \to \mathbb{F}_p$ be the map defined, for all $u \in A^*$, by

$$\pi_{i,j}(u) = (\pi(u))_{i,j} \tag{12}$$

By definition, a language K is recognized by π if there exists a subset S of G such that $K = \pi^{-1}(S)$. According to Proposition 4, K is a finite union of languages of the form $L(f_1, \ldots, f_r, p)$. We now give an algorithm to obtain this representation explicitly.
Setting, for each $s \in S$, $K_s = \pi^{-1}(s)$, one gets

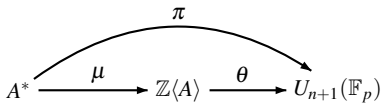
$$K = \bigcup_{s \in S} K_s \quad \text{and}$$
$$K_s = \{\, u \in A^* \mid \text{for } 1 \leqslant\ i < j \leqslant n+1,\ \pi_{i,j}(u) = s_{i,j} \,\}$$

It just remains to verify that the languages K_s are of the form $L(f_1, \ldots, f_r, p)$. But this follows immediately from the following result:

Proposition 5. *Each function $\pi_{i,j}$ is a linear combination of binomial coefficients.*

Proof. Let $\theta : A \to U_{n+1}(\mathbb{F}_p)$ be the map defined by $\theta(a) = \pi(a) - 1$ for all $a \in A$. Then θ extends to a ring morphism $\theta : \mathbb{Z}\langle A \rangle \to U_{n+1}(\mathbb{F}_p)$ and for $1 \leqslant i < j \leqslant n+1$, the maps $\theta_{i,j} : A^* \to \mathbb{F}_p$ are defined as in (12). Since $\theta(a)$ is a strictly triangular matrix for all $a \in A$, it follows that $\theta(x) = 0$ for all words x of length $> n$. Note however that $\theta(x)$ is not in general equal to $\pi(x) - 1$.

Let also $\mu : A^* \to \mathbb{Z}\langle A \rangle$ be the monoid morphism defined by $\mu(a) = 1 + a$ for all $a \in A$. Thus μ is the restriction to A^* of the Magnus automorphism introduced in Section . Since the formula $\theta(\mu(a)) = \theta(1 + a) = 1 + \theta(a) = \pi(a)$ holds for all $a \in A$, one has $\pi = \theta \circ \mu$.



[2] An $n \times n$-matrix is *unitriangular* if its diagonal coefficients are all equal to 1 and all its coefficients below the diagonal are equal to 0.

[3] The switch from n to $n+1$ will be justified later on.

It follows by (3) that

$$\pi(u) = \theta(\mu(u)) = \theta\Big(\sum_{x\in A^*} \binom{u}{x} x\Big) = \sum_{x\in A^*} \binom{u}{x}\theta(x) = \sum_{|x|\leqslant n} \binom{u}{x}\theta(x)$$

and hence

$$\pi_{i,j}(u) = \sum_{|x|\leqslant n} \theta_{i,j}(x)\binom{u}{x} \tag{13}$$

which shows that $\pi_{i,j}$ is a linear combination of binomial coefficients.

An interesting special case occurs if the language is defined by constraints on the first row of the matrix, for instance for a language of the form

$$L = \{\, u \in A^* \mid \pi_{1,2}(u) = \cdots = \pi_{1,n}(u) = 0 \,\}$$

Observing that L can also be written as

$$L = \{\, u \in A^* \mid (1,0,\ldots,0)\pi(u) = (1,0,\ldots,0) \,\}$$

one can directly obtain a deterministic automaton for L by taking $\mathbb{F}_p^n$ as set of states, the state $(0,\ldots,0)$ as initial and unique final state and by defining the transitions, for each $(z_1,\ldots,z_n) \in \mathbb{F}_p^n$ and each letter a, by setting

$$\begin{aligned}(z_1,\ldots,z_n)\cdot a = (z'_1,\ldots,z'_n),&\\ \text{where } (1,z_1,\ldots,z_n)\pi(a) = (1,z'_1,\ldots,z'_n),&\end{aligned} \tag{14}$$

that is,

$$\begin{aligned} z'_1 &= \pi_{1,2}(a) + z_1,\\ z'_2 &= \pi_{1,3}(a) + \pi_{2,3}(a)z_1 + z_2,\\ z'_3 &= \pi_{1,4}(a) + \pi_{2,4}(a)z_1 + \pi_{3,4}(a)z_2 + z_3, \text{ etc.}\end{aligned}$$

This algorithm is illustrated by the examples presented in Section .

4.3 Three examples

These three examples will be used in Section . The languages of the first two examples were also considered by Thérien [15].

Example 2. The subgroup of $U_3(\mathbb{F}_2)$ generated by the two matrices

$$a = \begin{pmatrix} 1 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 & 0 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}$$

is isomorphic to D_4. A confluent rewriting system for this group is $a^2 \to 1$, $b^2 \to 1$ and $baba \to abab$. The group consists of the matrices

$$1=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix} \quad a=\begin{pmatrix}1&1&0\\0&1&0\\0&0&1\end{pmatrix} \quad b=\begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix} \quad ab=\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\end{pmatrix}$$
$$ba=\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix} \quad aba=\begin{pmatrix}1&0&1\\0&1&1\\0&0&1\end{pmatrix} \quad bab=\begin{pmatrix}1&1&1\\0&1&0\\0&0&1\end{pmatrix} \quad abab=\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\end{pmatrix}$$

Let $\pi : A^* \to D_4$ be the natural morphism and let

$$L_1 = \{u \in A^* \mid \pi_{1,2}(u) = \pi_{1,3}(u) = 0\}.$$

To obtain a deterministic automaton for L_1, we take $\mathbb{F}_2^2$ as the set of states and define the transitions, for all $(z_1, z_2) \in \mathbb{F}_2^2$, by setting

$$\begin{cases} (z_1,z_2)\cdot a = (1+z_1, z_2) & (15)\\ (z_1,z_2)\cdot b = (z_1, z_1+z_2) & (16)\end{cases}$$

The resulting automaton, which turns out to be minimal, is the following:

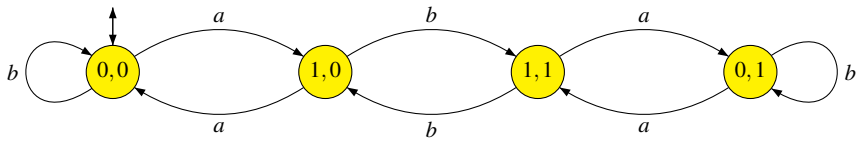


Figure 1 The minimal automaton of L_1.

The syntactic monoid of L_1 is the group D_4 presented by the relations $a^2 = 1$, $b^2 = 1$ and $(ba)^2 = (ab)^2$. Its syntactic image is $\{1, b\}$.

	1	2	3	4
$*\ 1$	1	2	3	4
a	2	1	4	3
b	1	3	2	4
ab	3	1	4	2

	1	2	3	4
ba	2	4	1	3
aba	4	2	3	1
bab	3	4	1	2
$abab$	4	3	2	1

Applying (13) with $n = 2$ one gets

$$\begin{aligned}
\pi_{1,2}(u) &= \sum_{|x|\leqslant 2} \binom{u}{x}\theta_{1,2}(x) = \binom{u}{1}\theta_{1,2}(1) + \binom{u}{a}\theta_{1,2}(a) + \binom{u}{b}\theta_{1,2}(b) \\
&\quad + \binom{u}{aa}\theta_{1,2}(aa) + \binom{u}{ab}\theta_{1,2}(ab) + \binom{u}{ba}\theta_{1,2}(ba) + \binom{u}{bb}\theta_{1,2}(bb) \\
&= \binom{u}{a} \\
\pi_{1,3}(u) &= \sum_{|x|\leqslant 2} \binom{u}{x}x_{1,3} = \binom{u}{1}\theta_{1,3}(1) + \binom{u}{a}\theta_{1,3}(a) + \binom{u}{b}\theta_{1,3}(b) \\
&\quad + \binom{u}{aa}\theta_{1,3}(aa) + \binom{u}{ab}\theta_{1,3}(ab) + \binom{u}{ba}\theta_{1,3}(ba) + \binom{u}{bb}\theta_{1,3}(bb) \\
&= \binom{u}{ab}
\end{aligned}$$

It follows that

$$L_1 = \left\{ u \in \{a,b\}^* \mid \binom{u}{a} \equiv \binom{u}{ab} \equiv 0 \bmod 2 \right\} \tag{17}$$

Moreover, for all $u \in \{a,b\}^*$,

$$(0,0)\cdot u = \left(\binom{u}{a}, \binom{u}{ab} \right)$$

where the binomial coefficients are computed modulo 2. Thus the states of the minimal automaton of L_1 encode the possible values modulo 2 of these two binomial coefficients. Now, one can recover (15) and (16) by observing that, if

$$(0,0)\cdot u = (z_1,z_2) = \left(\binom{u}{a}, \binom{u}{ab} \right)$$

then

$$\begin{aligned}
(0,0)\cdot ua = (z_1,z_2)\cdot a &= \left(\binom{ua}{a}, \binom{ua}{ab} \right) = \left(\binom{u}{a} + 1, \binom{u}{ab} \right) \\
&= (z_1+1, z_2)
\end{aligned}$$

and

$$\begin{aligned}
(0,0)\cdot ub = (z_1,z_2)\cdot b &= \left(\binom{ub}{a}, \binom{ub}{ab} \right) = \left(\binom{u}{a}, \binom{u}{ab} + \binom{u}{a} \right) \\
&= (z_1, z_1+z_2)\,.
\end{aligned}$$

Example 3. The group D_4 is also generated by the two matrices

$$a = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A confluent rewriting system for this group is $b^2 \to 1$, $aba \to b$, $ba^2 \to a^2b$, $bab \to a^3$, $a^4 \to 1$ and $a^3b \to ba$. The group consists of the matrices

$$1=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix} \qquad a=\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix} \qquad b=\begin{pmatrix}1&1&1\\0&1&0\\0&0&1\end{pmatrix} \qquad a^2=\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\end{pmatrix}$$
$$ab=\begin{pmatrix}1&0&1\\0&1&1\\0&0&1\end{pmatrix} \qquad ba=\begin{pmatrix}1&0&0\\0&1&1\\0&0&1\end{pmatrix} \qquad a^3=\begin{pmatrix}1&1&1\\0&1&1\\0&0&1\end{pmatrix} \qquad a^2b=\begin{pmatrix}1&1&0\\0&1&0\\0&0&1\end{pmatrix}$$

Let $\pi : A^* \to D_4$ be the natural morphism and let

$$L_2 = \{u \in A^* \mid \pi_{1,2}(u) = \pi_{1,3}(u) = 0\}.$$

To obtain a deterministic automaton for L_2, we take $\mathbb{F}_2^2$ as the set of states and define the transitions, for all $(z_1, z_2) \in \mathbb{F}_2^2$, by setting

$$\begin{cases} (z_1, z_2) \cdot a = (1 + z_1, z_1 + z_2) & (18) \\ (z_1, z_2) \cdot b = (1 + z_1, 1 + z_2) & (19) \end{cases}$$

The resulting automaton, which turns out to be minimal, is the following:

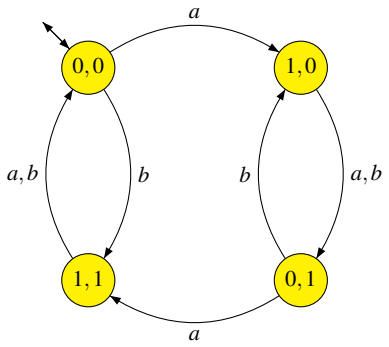


Figure 2 The minimal automaton of L_2.

Applying (13) with $n = 2$ one gets[4]

[4] It is easy to make mistakes in this computation. Recall that in general $\theta(x) \neq \pi(x) - 1$. Thus for instance $\theta(ba) = \theta(b)\theta(a) = \left(\begin{smallmatrix}0&0&1\\0&0&0\\0&0&0\end{smallmatrix}\right)$ and $\pi(ba) - 1 = \left(\begin{smallmatrix}0&0&0\\0&0&1\\0&0&0\end{smallmatrix}\right)$, whence $\theta_{1,3}(ba) = 1$.

$$\begin{aligned}
\pi_{1,2}(u) &= \sum_{|x|\leqslant 2} \binom{u}{x}\theta_{1,2}(x) = \binom{u}{1}\theta_{1,2}(1) + \binom{u}{a}\theta_{1,2}(a) + \binom{u}{b}\theta_{1,2}(b) \\
&\quad + \binom{u}{aa}\theta_{1,2}(aa) + \binom{u}{ab}\theta_{1,2}(ab) + \binom{u}{ba}\theta_{1,2}(ba) + \binom{u}{bb}\theta_{1,2}(bb) \\
&= \binom{u}{a} + \binom{u}{b} \\
\pi_{1,3}(u) &= \sum_{|x|\leqslant 2} \binom{u}{x}\theta_{1,3}(x) = \binom{u}{1}\theta_{1,3}(1) + \binom{u}{a}\theta_{1,3}(a) + \binom{u}{b}\theta_{1,3}(b) \\
&\quad + \binom{u}{aa}\theta_{1,3}(aa) + \binom{u}{ab}\theta_{1,3}(ab) + \binom{u}{ba}\theta_{1,3}(ba) + \binom{u}{bb}\theta_{1,3}(bb) \\
&= \binom{u}{b} + \binom{u}{aa} + \binom{u}{ba}
\end{aligned}$$

It follows that

$$L_2 = \left\{ u \in \{a,b\}^* \mid \binom{u}{a} + \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ab} \equiv 0 \bmod 2 \right\} \tag{20}$$

Moreover, for all $u \in \{a,b\}^*$,

$$(0,0)\cdot u = \left(\binom{u}{a} + \binom{u}{b}, \binom{u}{b} + \binom{u}{aa} + \binom{u}{ba} \right)$$

where the binomial coefficients are computed modulo 2. Thus the states of the minimal automaton of L_1 encode the possible values modulo 2 of these two linear combinations of binomial coefficients. Now, one can recover (18) and (19) by observing that, if

$$(0,0)\cdot u = (z_1, z_2) = \left(\binom{u}{a} + \binom{u}{b}, \binom{u}{b} + \binom{u}{aa} + \binom{u}{ba} \right)$$

then

$$\begin{aligned}
(0,0)\cdot ua = (z_1,z_2)\cdot a &= \left(\binom{ua}{a} + \binom{ua}{b}, \binom{ua}{b} + \binom{ua}{aa} + \binom{ua}{ba} \right) \\
&= \left(\binom{u}{a} + 1 + \binom{u}{b}, \binom{u}{b} + \binom{u}{aa} + \binom{u}{a} + \binom{u}{ba} + \binom{u}{b} \right) \\
&= (z_1 + 1, z_1 + z_2)
\end{aligned}$$

and

$$\begin{aligned}
(0,0)\cdot ub = (z_1,z_2)\cdot b &= \left(\binom{ub}{b} + \binom{ub}{a}, \binom{ub}{b} + \binom{ub}{aa} + \binom{ub}{ba} \right) \\
&= \left(\binom{u}{a} + \binom{u}{b} + 1, \binom{u}{b} + 1 + \binom{u}{aa} + \binom{u}{ba} \right) \\
&= (z_1 + 1, z_2 + 1)\,.
\end{aligned}$$

The syntactic monoid of L_2 is the group D_4, but this time presented by the group relations $b^2 = 1$, $a^4 = 1$ and $a^3 b = ba$. Its syntactic image is $\{1, ba\}$.

$*$	1	2	3	4
1	1	2	3	4
a	2	3	4	1
b	4	3	2	1
a^2	3	4	1	2

	1	2	3	4
ab	3	2	1	4
ba	1	4	3	2
a^3	4	1	2	3
a^2b	2	1	4	3

Example 4. The subgroup of $U_4(\mathbb{F}_2)$ generated by the two matrices

$$a = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

is isomorphic to Q_8. A confluent rewriting system for this group is $b^2 \to a^2$, $aba \to b$, $ba^2 \to a^2b$, $bab \to a$, $a^4 \to 1$ and $a^3b \to ba$. The group consists of the matrices of the following form, where $\varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathbb{F}_2$.

$$\begin{pmatrix} 1 & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \\ 0 & 1 & 0 & \varepsilon_1 + \varepsilon_2 \\ 0 & 0 & 1 & \varepsilon_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Let $\pi : A^* \to Q_8$ be the natural morphism and let

$$L_3 = \{u \in A^* \mid \pi_{1,2}(u) = \pi_{1,3}(u) = \pi_{1,4}(u) = 0\}.$$

To obtain a deterministic automaton for L_2, we take $\mathbb{F}_2^3$ as the set of states and define the transitions, for all $(z_1, z_2, z_3) \in \mathbb{F}_2^3$, by setting

$$\begin{cases} (z_1, z_2, z_3) \cdot a = (z_1 + 1, z_2, z_1 + z_3) & (21) \\ (z_1, z_2, z_3) \cdot b = (z_1, z_2 + 1, z_1 + z_2 + z_3) & (22) \end{cases}$$

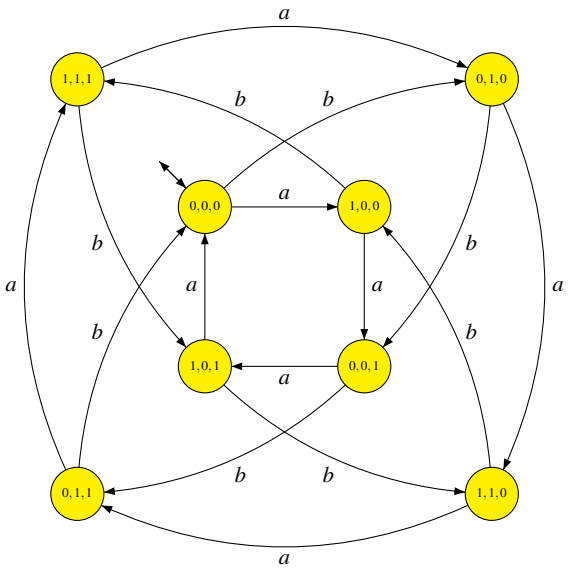


Figure 3 The minimal automaton of L_3.

Applying (13) with $n = 3$ one gets

$$\begin{aligned}
\pi_{1,2}(u) &= \sum_{|x|\leqslant 3} \binom{u}{x} \theta_{1,2}(x) = \binom{u}{1}\theta_{1,2}(1) + \binom{u}{a}\theta_{1,2}(a) + \binom{u}{b}\theta_{1,2}(b) \\
&\quad + \binom{u}{aa}\theta_{1,2}(aa) + \binom{u}{ab}\theta_{1,2}(ab) + \binom{u}{\theta}(ba)_{1,2}ba + \binom{u}{bb}\theta_{1,2}(bb) \\
&= \binom{u}{a} \\
\pi_{1,3}(u) &= \sum_{|x|\leqslant 3} \binom{u}{x} \theta_{1,3}(x) = \binom{u}{1}\theta_{1,3}(1) + \binom{u}{a}\theta_{1,3}(a) + \binom{u}{b}\theta_{1,3}(b) \\
&\quad + \binom{u}{aa}\theta_{1,3}(aa) + \binom{u}{ab}\theta_{1,3}(ab) + \binom{u}{ba}\theta_{1,3}(ba) + \binom{u}{bb}\theta_{1,3}(bb) \\
&= \binom{u}{b} \\
\pi_{1,4}(u) &= \sum_{|x|\leqslant 3} \binom{u}{x} \theta_{1,4}(x) = \binom{u}{1}\theta_{1,4}(1) + \binom{u}{a}\theta_{1,4}(a) + \binom{u}{b}\theta_{1,4}(b) \\
&\quad + \binom{u}{aa}\theta_{1,4}(aa) + \binom{u}{ab}\theta_{1,4}(ab) + \binom{u}{ba}\theta_{1,4}(ba) + \binom{u}{bb}\theta_{1,4}(bb) \\
&= \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb}
\end{aligned}$$

It follows that

$$L_3 = \left\{ u \in \{a,b\}^* \mid \binom{u}{a} \equiv \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} \equiv 0 \bmod 2 \right\} \tag{23}$$

Moreover, for all $u \in \{a,b\}^*$,

$$(0,0,0)\cdot u = \left(\binom{u}{a}, \binom{u}{b}, \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} \right)$$

where the binomial coefficients are computed modulo 2. Thus the states of the minimal automaton of L_1 encode the possible values modulo 2 of these two linear combinations of binomial coefficients. Now, one can recover (21) and (22) by observing that, if

$$(0,0,0)\cdot u = (z_1,z_2,z_3) = \left(\binom{u}{a}, \binom{u}{b}, \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} \right)$$

then

$$\begin{aligned}(0,0,0)\cdot ua = (z_1,z_2,z_3)\cdot a &= \left(\binom{ua}{a}, \binom{ua}{b}, \binom{ua}{aa} + \binom{ua}{ab} + \binom{ua}{bb} \right) \\ &= \left(\binom{u}{a} + 1, \binom{u}{b}, \binom{u}{aa} + \binom{u}{a} + \binom{u}{ab} + \binom{u}{bb} \right) \\ &= (z_1 + 1, z_2, z_1 + z_3)\end{aligned}$$

and

$$\begin{aligned}(0,0,0)\cdot ub = (z_1,z_2,z_3)\cdot b &= \left(\binom{ub}{a}, \binom{ub}{b}, \binom{ub}{aa} + \binom{ub}{ab} + \binom{ub}{bb} \right) \\ &= \left(\binom{u}{a}, \binom{u}{b} + 1, \binom{u}{aa} + \binom{u}{ab} + \binom{u}{a} + \binom{u}{bb} + \binom{u}{b} \right) \\ &= (z_1, z_2 + 1, z_1 + z_2 + z_3)\end{aligned}$$

The syntactic monoid of L_3 is the group Q_8 presented by the group relations $a^4 = 1$, $b^2 = a^2$ and $a^3 b = ba$. Its syntactic image is $\{1\}$.

	1	2	3	4	5	6	7	8
$*\,1$	1	2	3	4	5	6	7	8
a	2	3	4	1	6	7	8	5
b	6	5	8	7	4	3	2	1
a^2	3	4	1	2	7	8	5	6

	1	2	3	4	5	6	7	8
ab	5	8	7	6	3	2	1	4
ba	7	6	5	8	1	4	3	2
a^3	4	1	2	3	8	5	6	7
$a^2 b$	8	7	6	5	2	1	4	3

The Cayley graph of this group is represented in Figure 4. As one can see, this is exactly the same automaton as in Figure 3, up to the following renaming of the states:

$$\begin{array}{llll}(0,0,0) \leftrightarrow 1 & (1,0,0) \leftrightarrow a & (0,0,1) \leftrightarrow a^2 & (1,0,1) \leftrightarrow a^3 \\ (0,1,0) \leftrightarrow b & (1,1,0) \leftrightarrow ba & (0,1,1) \leftrightarrow a^2 b & (1,1,1) \leftrightarrow ab\end{array} \tag{24}$$

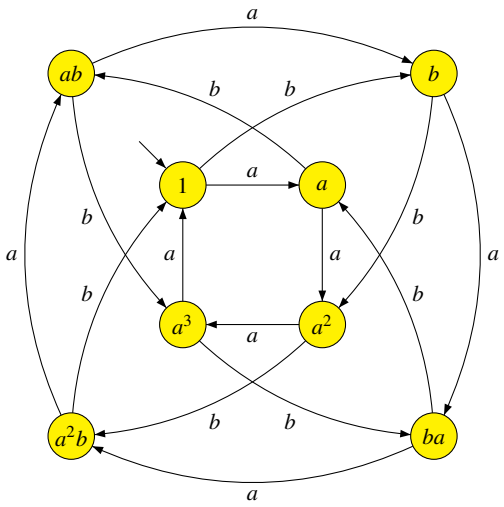


Figure 4 The Cayley graph of Q_8.

4.4 The varieties of languages $\mathscr{V}_{c,p}$

In this section, we revisit the congruences first introduced in [6, p. 240] and also studied in [15]. Let c be a nonnegative integer. For each alphabet A, let $\sim_{p,c}$ be the congruence on A^* defined by $u \sim_{p,c} v$ if and only if, for all words x such that $0 \leqslant |x| \leqslant c$,

$$\binom{u}{x} \equiv \binom{v}{x} \bmod p$$

This congruence has finite index and the languages which are saturated for this congruence form a Boolean algebra $\mathscr{V}_{c,p}(A^*)$, which is also the Boolean algebra generated by the languages $L(x,r,p)$ for $0 \leqslant r < p$ and $|x| \leqslant c$. x Let us first show that the class $\mathscr{V}_{c,p}$ is closed under inverses of morphisms. This relies on the following result.

Proposition 6. *Let $\varphi : A^* \to B^*$ be a morphism. Let u and v be two words of A^* such that $u \sim_{p,c} v$. Then $\varphi(u) \sim_{p,c} \varphi(v)$.*

Proof. If $u \sim_{p,c} v$, one has, for $0 \leqslant |s| \leqslant c$, $\binom{u}{s} \equiv \binom{v}{s} \bmod p$. Therefore by (5) we obtain for $|x| \leqslant c$,

$$\binom{\varphi(u)}{x} - \binom{\varphi(v)}{x} = \sum_{|s| \leqslant |x|} \left(\binom{u}{s} - \binom{v}{s} \right) \langle \gamma(s), x \rangle \equiv 0 \bmod p$$

Thus $\varphi(u) \sim_{p,c} \varphi(v)$.

We can now state:

Proposition 7. *Let $\varphi : A^* \to B^*$ be a morphism and L a language of $\mathscr{V}_{c,p}(B^*)$. Then $\varphi^{-1}(L)$ belongs to $\mathscr{V}_{c,p}(A^*)$.*

Proof. Let L be a language of $\mathscr{V}_{c,p}(B^*)$. Let $u \in \varphi^{-1}(L)$ and let v be a word such that $u \sim_{p,c} v$. Then $\varphi(u) \sim_{p,c} \varphi(v)$ by Proposition 6, and since $u \in L$ and L is saturated by $\sim_{p,c}$, we get $\varphi(v) \in L$, that is, $v \in \varphi^{-1}(L)$. This proves that $\varphi^{-1}(L)$ is saturated by $\sim_{p,c}$ and therefore $\varphi^{-1}(L)$ belongs to $\mathscr{V}_{c,p}(A^*)$.

Proposition 8. *For each c, the class $\mathscr{V}_{c,p}$ is a variety of languages.*

Proof. Proposition 7 shows that he class $\mathscr{V}_{c,p}$ is closed under inverses of morphisms. Furthermore, $\mathscr{V}_{c,p}(A^*)$ is by definition a Boolean algebra, generated by the languages of the form $L(x,r,p)$. We claim that it is closed under left quotient by a word u. Arguing on induction on the length of u, it suffices to consider the case where u is a letter a. Now, since left quotients commute with Boolean operations, it suffices to prove that any left quotient of the form $a^{-1}\big(L(x,r,p)\big)$ belongs to $\mathscr{V}_{c,p}(A^*)$. If x is the empty word, then $L(x,r,p)$ is either empty or equal to A^* and the result is trivial. Suppose that x is nonempty. Then, we get by (1):

$$
\begin{aligned}
a^{-1}\big(L(x,r,p)\big) &= \left\{ u \in A^* \mid \binom{au}{x} \equiv r \bmod p \right\} \\
&= \begin{cases} \{u \in A^* \mid \binom{u}{x} + \binom{u}{s} \equiv r \bmod p\} & \text{if } x = as \text{ for some } s \\ \{u \in A^* \mid \binom{u}{x} \equiv r \bmod p\} & \text{otherwise} \end{cases} \\
&= \begin{cases} \cup_{r_1+r_2 \equiv r \bmod p} \big(L(x,r_1,p) \cap L(s,r_2,p)\big) & \text{if } x = as \\ L(x,r,p) & \text{otherwise} \end{cases}
\end{aligned}
$$

which proves the claim. A dual argument proves that $\mathscr{V}_{c,p}(A^*)$ is closed under right quotient. Thus $\mathscr{V}_{c,p}$ is a variety of languages.

5 The formation generated by D_4 and by Q_8

We are now ready to prove our main result.

Theorem 2. *The groups D_4 and Q_8 generate the same formation and the associated formation of languages is the variety $\mathscr{V}_{2,2}$.*

Proof. Let $\mathbf{F}_1$ [$\mathbf{F}_2$] be the formation generated by D_4 [Q_8] and let $\mathscr{F}_1$ [$\mathscr{F}_2$] be the associated formation of languages. Let $\mathscr{V} = \mathscr{V}_{2,2}$ and let $\mathbf{V}$ be the associated group formation, which is actually a variety. For each alphabet A, $\mathscr{V}(A^*)$ is by definition the Boolean algebra generated by the languages $L(x,r,2)$ for $0 \leqslant r < 2$ and $|x| \leqslant 2$. Proposition 8 shows that $\mathscr{V}$ is a variety. We shall prove successively the following properties:

(1) D_4 and Q_8 belong to $\mathbf{V}$, and hence $\mathscr{F}_1$ and $\mathscr{F}_2$ are contained in $\mathscr{V}$,

(2) for each alphabet A, for $0 \leqslant r < 2$ and $|x| \leqslant 1$, the language $L(x,r,2)$ belongs to $\mathscr{F}_1(A^*)$ and to $\mathscr{F}_2(A^*)$,

(3) $\mathscr{V}$ is contained in $\mathscr{F}_1$ and hence $\mathscr{V} = \mathscr{F}_1$,

(4) $\mathscr{F}_1$ is contained in $\mathscr{F}_2$.

Step 1. The syntactic monoids of L_1 and L_2 are both equal to D_4 and that of L_3 is equal to Q_8. Formula (17) shows that L_1 belongs to $\mathscr{V}(\{a,b\}^*)$ and thus D_4 belongs to $\mathbf{V}$. Moreover, Formula (23) shows that L_3 can be written as

$$L(a,0,2) \cap L(b,0,2) \cap \Big(\bigcup_{i+j+k \equiv 0 \bmod 2} (L(ab,i,2) \cap L(aa,j,2) \cap L(bb,k,2)) \Big)$$

and thus L_3 belongs to $\mathscr{V}(\{a,b\}^*)$. It follows that Q_8 belongs to $\mathbf{V}$.

Step 2. If $x = 1$, the result is trivial. If $x = a$, where a is a letter, the syntactic monoid of $L(a,r,2)$ is the cyclic group C_2. Since C_2 is a quotient of both D_4 and Q_8, it belongs to $\mathbf{F}_1$ and to $\mathbf{F}_2$ and thus $L(a,r,2)$ belongs to $\mathscr{F}_1(A^*)$ and to $\mathscr{F}_2(A^*)$.

Step 3. Let A be an alphabet. It suffices to prove that, for $|x| \leqslant 2$ and $r = 0$ or $r = 1$, the language $L(x,r,2)$ belongs to $\mathscr{F}_1(A^*)$. Let $c(x)$ be the set of all letters occurring in x. In the minimal automaton of $L(x,r,2)$, every letter of $A \setminus c(x)$ acts as the identity on the set of states. It follows that the languages $L(x,r,2)$ and the language

$$\left\{ u \in c(x)^* \mid \binom{u}{x} \equiv r \bmod 2 \right\}$$

have the same syntactic monoid. Therefore, we may assume without loss of generality that $A = \{a,b\}$.

It already follows from (2) that for $|x| \leqslant 1$, $L(x,r,2)$ belongs to $\mathscr{F}_1(A^*)$. Suppose now that $x = ab$ with $a \neq b$. Then the minimal automaton of $L(ab,0,2)$ is obtained from the automaton of Figure 1 by taking $(0,0)$ and $(1,0)$ as final states. Indeed in this way the parameter $z_2 = \binom{u}{ab}$ will be equal to zero modulo 2. Thus the syntactic monoid of $L(ab,0,2)$ is D_4 and since D_4 belongs to $\mathbf{F}_1$, the language $L(ab,0,2)$ belongs to $\mathscr{F}_1(A^*)$ and so does its complement $L(ab,1,2)$.

Consider now the case $x = aa$. The automaton obtained from the automaton of Figure 2 by taking $(0,0)$ and $(1,0)$ as final states recognizes the language

$$K = \left\{ u \in \{a,b\}^* \mid \binom{u}{b} + \binom{u}{ba} + \binom{u}{aa} \equiv 0 \bmod 2 \right\}$$

The syntactic monoid of K is also D_4 and thus $K \in \mathscr{F}_1(A^*)$. Now since

$$\begin{aligned} L(aa,0,2) = {} & (K \cap L(b,0,2) \cap L(ba,0,2)) \cup (K \cap L(b,1,2) \cap L(ba,1,2)) \\ & \cup (K^c \cap L(b,0,2) \cap L(ba,1,2)) \cup (K^c \cap L(b,1,2) \cap L(ba,0,2)) \end{aligned}$$

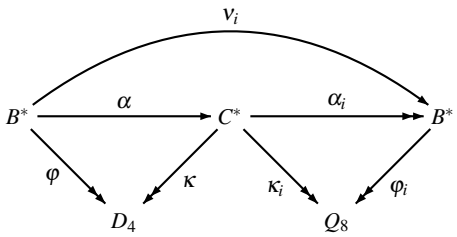
the language $L(aa,0,2)$ and its complement $L(aa,1,2)$ belong to $\mathscr{F}_1(\{a,b\}^*)$. Since the languages $L(bb,r,2)$ and $L(aa,r,2)$ have the same syntactic monoid, we also have $L(bb,r,2) \in \mathscr{F}_1(\{a,b\}^*)$ for $r = 0$ and $r = 1$.

Step 4. We will show that some language L having D_4 as syntactic monoid belongs to $\mathscr{F}_2$. By the Formation Theorem, this will show that D_4 belongs to F_2 and hence that $\mathscr{F}_1$ is contained in $\mathscr{F}_2$ as required. We choose for L the language of Example 3:

$$L = \varphi^{-1}(1) = \left\{ u \in \{a,b\}^* \mid \binom{u}{a} \equiv \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ab} \equiv 0 \bmod 2 \right\}$$

Let us now view D_4 as the group $\{1,a,b,a^2,ab,ba,a^3,a^2b\}$ presented by the group relations $b^2 = 1$, $a^4 = 1$ and $a^3b = ba$ and Q_8 as the group $\{1,a,b,a^2,ab,ba,a^3,a^2b\}$ presented by the group relations $a^4 = 1$, $b^2 = a^2$ and $a^3b = ba$.

Let $B = \{a,b\}$ and $C = \{a,b,c\}$. Consider the following diagram,



in which the morphisms are defined by

$$\begin{array}{llll} \varphi(a) = a & \varphi(b) = b & \alpha(a) = c & \alpha(b) = a \\ \varphi_1(a) = a & \varphi_1(b) = b & \varphi_2(a) = a & \varphi_2(b) = b \\ \nu_1(a) = a^2b & \nu_1(b) = a & \nu_2(a) = 1 & \nu_2(b) = a \end{array}$$

and

$$\begin{array}{lll} \alpha_1(a) = a & \alpha_1(b) = b & \alpha_1(c) = a^2b \\ \alpha_2(a) = a & \alpha_2(b) = b & \alpha_2(c) = 1 \\ \kappa_1(a) = a & \kappa_1(b) = b & \kappa_1(c) = a^2b \\ \kappa_2(a) = a & \kappa_2(b) = b & \kappa_2(c) = 1 \\ \kappa(a) = b & \kappa(b) = 1 & \kappa(c) = a \end{array}$$

Note that $\varphi_1 = \varphi_2$, but we keep two distinct names to preserve homogeneity of the notation. All these morphisms make the diagram commutative. Let

$$\begin{array}{ll} R_1 = \varphi_1^{-1}(1) = \varphi_2^{-1}(1) & R_b = \varphi_1^{-1}(b) = \varphi_2^{-1}(b) \\ R_{a^2} = \varphi_1^{-1}(a^2) = \varphi_2^{-1}(a^2) & R_{a^2b} = \varphi_1^{-1}(a^2b) = \varphi_2^{-1}(a^2b) \end{array}$$

Lemma 1. *The languages R_1, R_b, R_{a^2} and R_{a^2b} are all recognized by Q_8 and hence belong to $\mathscr{F}_2(B^*)$.*

Proof. Indeed, these languages are accepted by the automaton represented in Figure 4 by taking as final state 1, b, a^2 and a^2b respectively. Therefore, these four languages are recognized by Q_8 and belong to $\mathscr{F}_2(B^*)$.

Using the state renaming described in (24), one sees that R_1, R_b, R_{a^2} and R_{a^2b} are also accepted by the automaton represented in Figure 3 by taking as final state $(0,0,0)$, $(0,1,0)$, $(0,0,1)$ and $(0,1,1)$ respectively. Coming back to the interpretation of these states as linear combinations of binomial coefficients, as described in Example 4, one gets the following explicit descriptions:

$$R_1 = \left\{u \in B^* \mid \binom{u}{a} \equiv \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} \equiv 0 \bmod 2\right\}$$
$$R_b = \left\{u \in B^* \mid \binom{u}{a} \equiv \binom{u}{b} + 1 \equiv \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} \equiv 0 \bmod 2\right\}$$
$$R_{a^2} = \left\{u \in B^* \mid \binom{u}{a} \equiv \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} + 1 \equiv 0 \bmod 2\right\}$$
$$R_{a^2b} = \left\{u \in B^* \mid \binom{u}{a} \equiv \binom{u}{b} + 1 \equiv \binom{u}{aa} + \binom{u}{ab} + \binom{u}{bb} + 1 \equiv 0 \bmod 2\right\}$$

Let

$$R = \left(\alpha_1^{-1}(R_1) \cap \alpha_2^{-1}(R_1)\right) \cup \left(\alpha_1^{-1}(R_b) \cap \alpha_2^{-1}(R_b)\right) \cup \left(\alpha_1^{-1}(R_{a^2}) \cap \alpha_2^{-1}(R_{a^2})\right) \cup \left(\alpha_1^{-1}(R_{a^2b}) \cap \alpha_2^{-1}(R_{a^2b})\right)$$

A lengthy computation[5] shows that

$$R = \left\{u \in C^* \mid \binom{u}{a} \equiv \binom{u}{c} \equiv \binom{u}{ac} + \binom{u}{bc} + \binom{u}{cb} + \binom{u}{cc} \equiv 0 \bmod 2\right\}$$

Now, by (2), one gets $\binom{u}{bc} + \binom{u}{cb} = \binom{u}{b}\binom{u}{c}$ and $\binom{u}{ac} + \binom{u}{ca} = \binom{u}{a}\binom{u}{c}$. It follows that

$$R = \left\{u \in C^* \mid \binom{u}{a} \equiv \binom{u}{c} \equiv \binom{u}{ca} + \binom{u}{cc} \equiv 0 \bmod 2\right\}$$

The syntactic monoid of R is D_4 and is syntactic morphism is κ.

Lemma 2. *The language R belongs to $\mathscr{F}_2(C^*)$.*

Proof. For $i = 1,2$, the morphism $\varphi_i \circ \alpha_i$ is equal to κ_i and thus is surjective. By definition of a formation of languages, the languages $\alpha_i^{-1}(R_1)$, $\alpha_i^{-1}(R_b)$, $\alpha_i^{-1}(R_{a^2})$ and $\alpha_i^{-1}(R_{a^2b})$ belong to $\mathscr{F}_2(C^*)$. It follows that R belongs to $\mathscr{F}_2(C^*)$.

Lemma 3. *The language $\alpha^{-1}(R)$ belongs to $\mathscr{F}_2(B^*)$.*

Proof. The syntactic morphism of R is κ. Then since $\kappa \circ \alpha = \varphi$, $\kappa \circ \alpha$ is surjective and by definition of a formation of languages, $\alpha^{-1}(R)$ belongs to $\mathscr{F}_2(B^*)$.

The last step consists in computing $\alpha^{-1}(R)$.

[5] See the Appendix.

Lemma 4. *One has $\alpha^{-1}(R) = L$ and thus L belongs to $\mathscr{F}_2(\{a,b\}^*)$.*

Proof. Since $\nu_i = \alpha_i \circ \alpha$, one gets

$$\alpha^{-1}(R) = \big(\nu_1^{-1}(R_1) \cap \nu_2^{-1}(R_1)\big) \cup \big(\nu_1^{-1}(R_b) \cap \nu_2^{-1}(R_b)\big) \cup \big(\nu_1^{-1}(R_{a^2}) \cap \nu_2^{-1}(R_{a^2})\big) \cup \big(\nu_1^{-1}(R_{a^2b}) \cap \nu_2^{-1}(R_{a^2b})\big)$$

Another lengthy computation shows that

$$\begin{aligned}\alpha^{-1}(R) &= \Big(\nu_1^{-1}(R_1) \cap \nu_2^{-1}(R_1)\Big) \cup \Big(\nu_1^{-1}(R_{a^2}) \cap \nu_2^{-1}(R_{a^2})\Big) \\ &= \left\{u \in B^* \mid \binom{u}{a} \equiv \binom{u}{b} \equiv \binom{u}{aa} + \binom{u}{ba} \equiv 0 \bmod 2\right\}\end{aligned}$$

Finally, Proposition 1 shows that when $\binom{u}{a} \equiv \binom{u}{b} \equiv 0 \bmod 2$, then $\binom{u}{ab} \equiv \binom{u}{ba} \equiv 0 \bmod 2$. It follows that $\alpha^{-1}(R) = L$.

This concludes the proof of Theorem 2.

Important remark. It is tempting to prove directly that the languages $\nu_1^{-1}(R_1)$, $\nu_2^{-1}(R_1)$, etc. belong to $\mathscr{F}_2(\{a,b\}^*)$. However, the morphism $\varphi_2 \circ \nu_2$ is not surjective and one cannot conclude directly.

6 Conclusion

We used language theory to prove that D_4 and Q_8 generate the same formation and that this formation is a variety of groups. Our project for the future would be to show, also by language theoretic means, that any formation generated by a single nilpotent group is a variety.

Acknowledgements

We would like to thank Ramón Esteban-Romero and Adolfo Ballester-Bolinches for their useful comments and suggestions.

References

1. A. BALLESTER-BOLINCHES AND L. M. EZQUERRO, *Classes of finite groups*, *Mathematics and Its Applications (Springer)* vol. 584, Springer, Dordrecht, 2006.
2. A. BALLESTER-BOLINCHES, J.-É. PIN AND X. SOLER-ESCRIVÀ, Formations of finite monoids and formal languages: Eilenberg's variety theorem revisited, *Forum Math.* **26** (2014), 1737–1761.

3. A. BALLESTER-BOLINCHES, J.-É. PIN AND X. SOLER-ESCRIVÀ, Languages associated with saturated formations of groups, *Forum Math.* **27** (2015), 1471–1505.
4. O. CARTON, J.-E. PIN AND X. SOLER-ESCRIVÀ, Languages Recognized by Finite Supersoluble Groups, *Journal of Automata, Languages and Combinatorics* **14**,2 (2009), 149–161.
5. K. DOERK AND T. HAWKES, *Finite soluble groups*, *de Gruyter Expositions in Mathematics* vol. 4, Walter De Gruyter & Co., Berlin, 1992.
6. S. EILENBERG, *Automata, languages, and machines. Vol. B*, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976. Pure and Applied Mathematics, Vol. 59.
7. W. GASCHÜTZ AND U. LUBESEDER, Kennzeichnung gesättigter Formationen, *Math. Z.* **82** (1963), 198–199.
8. M. LOTHAIRE, *Combinatorics on words*, Cambridge University Press, Cambridge, 1997. With a foreword by Roger Lyndon and a preface by Dominique Perrin, corrected reprint of the 1983 original, with a new preface by Perrin.
9. P. M. NEUMANN, A note on formations of finite nilpotent groups, *Bull. London Math. Soc.* **2** (1970), 91.
10. C. REUTENAUER, *Free Lie algebras*, *London Mathematical Society Monographs. New Series* vol. 7, The Clarendon Press, Oxford University Press, New York, 1993. Oxford Science Publications.
11. L. A. SHEMETKOV, Product of formations of algebraic systems, *Algebra and Logic* **23**,6 (1984), 484–490.
12. L. A. SHEMETKOV AND A. N. SKIBA, *Formations of algebraic systems. (Formatsii algebraicheskikh sistem.)*, Современная Алгебра. *[Modern Algebra]*, Sovremennaya Algebra. Moskva: Nauka. 256 P. R. 3.00, Moscow, 1989. With an English summary.
13. A. N. SKIBA, Finite subformations of varieties of algebraic systems, in *Problems in algebra, No. 2 (Russian)*, pp. 7–20, 126, "Universitet·skoe", Minsk, 1986.
14. H. STRAUBING, Families of recognizable sets corresponding to certain varieties of finite monoids, *J. Pure Appl. Algebra* **15**,3 (1979), 305–318.
15. D. THÉRIEN, Subword counting and nilpotent groups, in *Combinatorics on words (Waterloo, Ont., 1982)*, pp. 297–305, Academic Press, Toronto, ON, 1983.
16. P. WEIL, An extension of the Schützenberger product, in *Lattices, semigroups, and universal algebra (Lisbon, 1988)*, pp. 315–321, Plenum, New York, 1990.
17. P. WEIL, Products of languages with counter, *Theoret. Comput. Sci.* **76** (1990), 251–260.
18. P. WEIL, Closure of varieties of languages under products with counter, *J. Comput. System Sci.* **45** (1992), 316–339.

Syntactic Structures of Regular Languages

O. Klíma[1], L. Polák[1]

[1] *Department of Mathematics and Statistics, Masaryk University, Brno, Czech Republic,*
{klima,polak}@math.muni.cz

Abstract

We introduce here the notion of syntactic lattice algebra which is an analogy of the syntactic monoid and of the syntactic semiring. We present a unified approach to get those three structures.

1 Introduction

The Eilenberg like theorems establish bijections between the class of all (in some sense generalized) varieties of regular languages and the class of all pseudovarieties of certain algebraic structures — see Eilenberg [4], Pin [7], Straubing [12], Polák [10]. The classical result concerns the varieties of regular languages and pseudovarieties of finite monoids. The goal is an algorithmic procedure for deciding the membership of a given language in various significant classes of regular languages. A basic source of that theory is the book by Pin [9].

The aim of the present contribution is to introduce modifications of the notion of the syntactic monoid which would be useful in other variants of Eilenberg type theorems. As well-known the syntactic monoid of a language L over the alphabet A can be viewed as the transformation monoid of the minimal complete deterministic automaton D_L of L. More precisely, we let words of A^* act on states of D_L and the composition of such transformations corresponds to multiplication in the syntactic monoid.

To get analogues of A^* with the multiplication, we consider structures with more operations, namely we use here the following three term algebras:

- F is the absolutely free algebra over the alphabet A with the operation symbol $\cdot$ and nullary symbol λ,
- to get F' we enrich the previous signature by binary $\wedge$ and nullary $\top$,
- to get F'' we enrich the last signature by binary $\vee$ and nullary $\bot$.

Now we let our terms act on the set 2^{A^*} of all languages over A in a natural way (the formal definitions are in Section 3).

We show here that identifying terms of F (F' and F'') giving the same transformations, we get exactly the free monoid A^* over A, (the free semiring $A^\square$ over A and the free, so-called, lattice algebra $A^\diamond$ over A, respectively). Let us stress that all our considerations concern three levels: level of monoids – the classical one (Pin [8, 9]), level of semirings (considered also in Polák [10, 11]), and that of lattice algebras – a new contribution.

When generating subalgebras in 2^{A^*} by a single regular language L using terms from F, F' and F'', and choosing the final states appropriately, we get the classical minimal complete deterministic finite automaton of L (here called the canonical finite automaton of L), the canonical meet automaton of L (see Section 6 of Polák [10]) and the canonical lattice automaton of L, respectively. Then transforming those automata accordingly, we get the corresponding syntactic structures.

Our constructions are also accompanied by examples. Moreover, a significant instance of a future Eilenberg type theorems is presented.

2 Specific Algebraic Structures

Usually, a semiring has two binary operations denoted by $+$ and $\cdot$, where the neutral element for $+$ is denoted by 0. Since we work with idempotent semirings, which can be naturally ordered, we use the symbol $\wedge$ instead of $+$, and the symbol $\top$ instead of 0 in the following basic definition. By an *idempotent semiring* we mean the structure $(S,\wedge,\cdot,\top,1)$ where $(S,\wedge,\top)$ is a commutative idempotent monoid, also called *semilattice*, with the neutral element $\top$, $(S,\cdot,1)$ is a monoid with the neutral element 1 and the zero element $\top$, and the operations $\wedge$ and $\cdot$ satisfy the usual distributivity laws

$$(\forall a,b,c\in S)\quad a\cdot(b\wedge c)=a\cdot b\wedge a\cdot c,\ \ (b\wedge c)\cdot a=b\cdot a\wedge c\cdot a.$$

The set S can be naturally ordered: for every $a,b\in S$ we have $a\leq b$ if and only if $a\wedge b=a$. Then $\top$ becomes the greatest element in $(S,\leq)$. This explains our choice of the symbol $\top$.

The elements of the free idempotent semiring $A^{\square}$ over the set A can be represented by finite subsets of A^*. This representation is one-to-one. Operations are the operation of union and the obvious multiplication, $\emptyset$ is the neutral element for $\wedge$, the zero for $\cdot$ and $\{\lambda\}$ is the neutral element for the multiplication. If we identify each word $u\in A^*$ with the element $\{u\}\in A^{\square}$, then we can see A^* as a subset of $A^{\square}$. For each $U=\{u_1,\ldots,u_k\}$, $k>0$, $u_1,\ldots,u_k\in A^*$ we can write $U=u_1\wedge\cdots\wedge u_k$.

Next structure we use is the free bounded distributive lattice $A^{\diamond}$ over A^*. The elements of $A^{\diamond}$ are of the form

$$\begin{aligned}&\{\{u_{1,1},\ldots,u_{1,r_1}\},\ldots,\{u_{k,1},\ldots,u_{k,r_k}\}\},\ \text{where } k,r_1,\ldots,r_k\geq 0,\\ &\quad u_{i,j}\in A^* \text{ for } i=1,\ldots,k,\ j=1,\ldots,r_i \text{ and the inner sets}\\ &\quad \{u_{i,1},\ldots,u_{i,r_i}\}\text{'s are incomparable with respect to } \subseteq .\end{aligned}\tag{1}$$

The interpretation of the element of the form (1) is

$$(u_{1,1}\wedge\cdots\wedge u_{1,r_1})\vee\cdots\vee(u_{k,1}\wedge\cdots\wedge u_{k,r_k}).\tag{2}$$

Particularly, each element of the form $\{\{u_1,\dots,u_k\}\} \in A^\diamond$ is identified with $u_1 \wedge \dots \wedge u_k$, which is equal to $\{u_1,\dots,u_k\}$ in $A^\Box$. Thus we can see $A^\Box$ as a subset of $A^\diamond$ under the identification $U \mapsto \{U\}$. Defining the operations $\wedge$ and $\vee$ on the set $A^\diamond$, one uses the form (2) for the element of the form (1). For both operations, one uses the distributivity laws first and then the superfluous $(u_{j,1} \wedge \dots \wedge u_{j,r_j})$'s are omitted to get the (unique) element of the form (1). Notice that $\{\emptyset\}$ is the greatest element in $A^\diamond$ and $\emptyset$ is the smallest one. For more details concerning the free bounded distributive lattice see Appendix.

The structure $A^\diamond$ is equipped also with a multiplication, namely extend the multiplication from A^* to $A^\diamond$ using

$$\begin{aligned} \mathscr{U}\cdot(\mathscr{V}\wedge\mathscr{W}) &= \mathscr{U}\cdot\mathscr{V}\wedge\mathscr{U}\cdot\mathscr{W},\ (\mathscr{U}\wedge\mathscr{V})\cdot w = \mathscr{U}\cdot w\wedge\mathscr{V}\cdot w, \\ \mathscr{U}\cdot(\mathscr{V}\vee\mathscr{W}) &= \mathscr{U}\cdot\mathscr{V}\vee\mathscr{U}\cdot\mathscr{W},\ (\mathscr{U}\vee\mathscr{V})\cdot w = \mathscr{U}\cdot w\vee\mathscr{V}\cdot w, \end{aligned} \tag{3}$$

for $\mathscr{U},\mathscr{V},\mathscr{W} \in A^\diamond$, $w \in A^*$.

We consider various kinds of automata. All of them are deterministic and complete and could have infinite number of states. When using the term *semiautomata*, no initial nor final states are specified.

Having an equivalence relation ρ on a set G and an element $a \in G$, we denote by $a\rho$ the class of G/ρ containing a.

3 Transformation structures

Let A be a finite non-empty set. We consider the actions of term algebras mentioned above on languages over the alphabet A.

For $u \in A^*$ and $L \subseteq A^*$, we write $u^{-1}L = \{\, v \in A^* \mid uv \in L \,\}$. We speak about a *left quotient* of L.

Monoids. Let F be the absolutely free algebra (that is, the algebra of all terms) over a set A with respect to the binary operational symbol $\cdot$ and nullary operational symbol λ.

We define inductively the actions of elements of F on subsets of A^* :

$$L\circ\lambda = L,\ L\circ a = a^{-1}L \text{ for } a\in A,\ L\circ(u\cdot v) = (L\circ u)\circ v \text{ for } u,v\in F\,. \tag{4}$$

This leads to a natural identification of certain pairs of elements of F, namely: for $u,v \in F$, we put $u\ \rho^*\ v$ if and only if $(\forall\, L \subseteq A^*)\ L\circ u = L\circ v$.

Proposition 1. *The relation ρ^* is a congruence relation on F and F/ρ^* is isomorphic to the free monoid A^* over A via the extension of the mapping $a\rho^* \mapsto a$, $a \in A$.*

Proof. Let $u,v,w \in F$. If $u\ \rho^*\ v$ then, for each $L \subseteq A^*$, we have $L\circ u = L\circ v$. Therefore $L\circ(u\cdot w) = (L\circ u)\circ w = (L\circ v)\circ w = L\circ(v\cdot w)$, which gives $u\cdot w\ \rho^*\ v\cdot w$. Similarly, $L\circ(w\cdot u) = (L\circ w)\circ u = (L\circ w)\circ v = L\circ(w\cdot v)$, which gives $w\cdot u\ \rho^*\ w\cdot v$. Thus ρ^* is a congruence relation on F.

Now we prove that, for each $u,v,w \in F$, we have $(u\cdot v)\cdot w\ \rho^*\ u\cdot(v\cdot w)$, $\lambda\cdot u\ \rho^*\ u$ and $u\cdot\lambda\ \rho^*\ u$. Indeed, choosing $L\subseteq A^*$, it holds $L\circ((u\cdot v)\cdot w)=(L\circ(u\cdot v))\circ w=((L\circ u)\circ v)\circ w=(L\circ u)\circ(v\cdot w)=L\circ(u\cdot(v\cdot w))$. Furthermore, $L\circ(\lambda\cdot u)=(L\circ\lambda)\circ u=L\circ u$, and $L\circ(u\cdot\lambda)=(L\circ u)\circ\lambda=L\circ u$.

Thus we can omit brackets in elements of F and λ acts as a neutral element. Therefore every element of F/ρ^* can be represented by a word from A^*. It remains to show that different words u and v represent different elements of F/ρ^*. Indeed, for $u\neq v$, we have $\lambda\in\{u\}\circ u$ but $\lambda\notin\{u\}\circ v$.

Semirings. Let F' be the absolutely free algebra over A with respect to the operational symbols $\cdot,\lambda$, binary symbol $\wedge$ and nullary symbol $\top$. We define inductively the actions of elements of F' on 2^{A^*}: we use the formulas from (4) for $u,v\in F'$ and

$$L\circ\top=A^*,\ L\circ(u\wedge v)=(L\circ u)\cap(L\circ v)\text{ for }u,v\in F'. \tag{5}$$

Again, it leads to certain identification of pairs of elements of F', namely: for $u,v\in F'$, we put $u\ \rho^\square\ v$ if and only if $(\forall\, L\subseteq A^*)\ L\circ u=L\circ v$.

Proposition 2. *The relation $\rho^\square$ is a congruence relation on F' and $F'/\rho^\square$ is isomorphic to the free idempotent semiring $A^\square$ over A via the extension of the mapping $a\rho^\square\mapsto a$, $a\in A$.*

Proof. Let $u,v,w\in F'$. If $u\ \rho^\square\ v$ then, for each $L\subseteq A^*$, we have $L\circ u=L\circ v$. We get $u\cdot w\ \rho^\square\ v\cdot w$ and $w\cdot u\ \rho^\square\ w\cdot v$ as in the case of Proposition 1.

Furthermore, $L\circ(u\wedge w)=(L\circ u)\cap(L\circ w)=(L\circ v)\cap(L\circ w)=L\circ(v\wedge w)$, which gives $u\wedge w\ \rho^\square\ v\wedge w$. In the same way we can prove that $w\wedge u\ \rho^\square\ w\wedge v$. Thus $\rho^\square$ is a congruence relation on F'.

Now we show that $(F'/\rho^\square,\wedge,\top\rho^\square)$ is a commutative idempotent monoid with the neutral element $\top\rho^\square$. The commutativity and associativity of $\wedge$ is clear as well as the fact that $\top\rho^\square$ is a neutral element for the operation $\wedge$. To show the idempotency of $\wedge$ notice that, for each $L\subseteq A^*$ and $u\in F'$, we have $L\circ(u\wedge u)=(L\circ u)\cap(L\circ u)=L\circ u$.

The proof of the associativity of $\cdot$ on $F'/\rho^\square$ and the fact that $\lambda\rho^\square$ is a neutral element for the operation $\cdot$ is similar to that for monoids. The fact that $\top\rho^\square$ is a zero element for $\cdot$ is clear.

Finally, we prove the distributivity laws. Let $L\subseteq A^*$, $u,v,w\in F'$. Then $L\circ(u\cdot(v\wedge w))=(L\circ u)\circ(v\wedge w)=(L\circ u)\circ v\cap(L\circ u)\circ w=L\circ u\cdot v\cap L\circ u\cdot w=L\circ(u\cdot v\wedge u\cdot w)$. Similarly, $L\circ((u\wedge v)\cdot w)=(L\circ(u\wedge v))\circ w=(L\circ u\cap L\circ v)\circ w=(L\circ u)\circ w\cap(L\circ v)\circ w=(L\circ u\cdot w)\cap(L\circ v\cdot w)=L\circ(u\cdot w\wedge v\cdot w)$.

We have proved that $F'/\rho^\square$ with the appropriate operations is an idempotent semiring. Therefore every element of $F'/\rho^\square$ can be represented by $u_1\wedge\cdots\wedge u_k$ with $k\geq 0$ and $u_1,\ldots,u_k\in A^*$. To get the unique representation of such element we use the idempotency and commutativity law and represent the element in $F'/\rho^\square$ by the set $\{u_1,\ldots,u_k\}$. Having such two different sets $\{u_1,\ldots,u_k\}$ and $\{v_1,\ldots,v_\ell\}$, $\ell\geq 0$, $v_1,\ldots,v_\ell\in A^*$, we show that they are not $\rho^\square$-related. Indeed, put $L=\{u_1,\ldots,u_k\}$. Then $\lambda\in L\circ\{u_1,\ldots,u_k\}=u_1^{-1}L\cap\cdots\cap u_k^{-1}L$ and $\lambda\in L\circ\{v_1,\ldots,v_\ell\}$ would give $\{v_1,\ldots,v_\ell\}\subsetneqq\{u_1,\ldots,u_k\}$. Take $L=\{v_1,\ldots,v_\ell\}$ in this case.

Lattice algebras. Let F'' be the absolutely free algebra over A with respect to the operational symbols $\cdot, \lambda, \wedge, \top$, binary $\vee$ and nullary $\bot$. We use (4), (5) with $u, v \in F''$ and

$$L \circ \bot = \emptyset,\ L \circ (u \vee v) = (L \circ u) \cup (L \circ v) \text{ for } u, v \in F'' \ . \tag{6}$$

Again, it leads to certain identification of pairs of elements of F'', namely: for $u, v \in F''$, we put $u\ \rho^\diamond\ v$ if and only if $(\ \forall L \subseteq A^*)\ L \circ u = L \circ v$.

Proposition 3. *The relation $\rho^\diamond$ is a congruence relation on F'' and $F''/\rho^\diamond$ is isomorphic to the free bounded distributive lattice $A^\diamond$ over A^* equipped with multiplication satisfying (3), via the extension of the mapping $a\rho^\diamond \mapsto a,\ a \in A$.*

Proof. Let $u, v, w \in F''$. If $u\ \rho^\diamond\ v$ then, for each $L \subseteq A^*$, we have $L \circ u = L \circ v$. We get $u \cdot w\ \rho^\diamond\ v \cdot w$, $w \cdot u\ \rho^\diamond\ w \cdot v$, $u \wedge w\ \rho^\diamond\ v \wedge w$, $w \wedge u\ \rho^\diamond\ w \wedge v$ as in the case of Proposition 2. Furthermore, $L \circ (u \vee w) = (L \circ u) \cup (L \circ w) = (L \circ v) \cup (L \circ w) = L \circ (v \vee w)$, which gives $u \vee w\ \rho^\diamond\ v \vee w$. In the same way we can prove that $w \vee u\ \rho^\diamond\ w \vee v$. Thus $\rho^\diamond$ is a congruence relation on F''.

Now we state the properties of operations $\wedge, \vee, \cdot, \top, \bot$ and λ on $F''/\rho^\diamond$. Proofs of all statements are straightforward and therefore omitted. The operation $\wedge$ is commutative, associative and idempotent, $\top$ is the neutral element and $\bot$ is the zero. The operation $\vee$ is commutative, associative and idempotent, $\bot$ is the neutral element and $\top$ is the zero. The operations $\wedge$ and $\vee$ are connected by the distributivity laws. The operation $\cdot$ is associative, λ is the neutral element, $\top$ and $\bot$ are right zeros, and $\top \cdot a\ \rho^\diamond\ \top$, $\bot \cdot a\ \rho^\diamond \bot$ for all $a \in A$. Finally, the distributivity $u \cdot (v \vee w) = u \cdot v \vee u \cdot w$ holds for arbitrary $u, v, w \in F''$ and the distributivity $(u \vee v) \cdot w = u \cdot w \vee v \cdot w$ for $u, v \in F''$ and $w \in A^*$. Similarly for the operation $\wedge$.

We have proved that every element of $F''/\rho^\diamond$ can be represented as

$$(u_{1,1} \wedge \cdots \wedge u_{1,r_1}) \vee \cdots \vee (u_{k,1} \wedge \cdots \wedge u_{k,r_k}),$$

where $k, r_1, \ldots, r_k \geq 0$ and $u_{i,j} \in A^*$ for all $i = 1, \ldots, k,\ j = 1, \ldots, r_i$. (Here $k = 0$ corresponds to the element $\bot$ and $k = 1,\ r_1 = 0$ corresponds to the element $\top$.) Using the idempotency and commutativity of $\wedge$ and $\vee$ we can write such element even as $\{\{u_{1,1}, \ldots, u_{1,r_1}\}, \ldots, \{u_{k,1}, \ldots, u_{k,r_k}\}\}$. To get canonical forms remove the richer one from each pair of comparable inner sets.

Let $\mathscr{U} = \{U_1, \ldots, U_k\}$ and $\mathscr{V} = \{V_1, \ldots, V_\ell\}$ be different canonical forms. We show that $\mathscr{U}$ and $\mathscr{V}$ represent elements of F'' which are not $\rho^\diamond$-related. If $U_i \notin \mathscr{V}$, take $L = U_i$. Then $\lambda \in L \circ \mathscr{U}$ and $\lambda \in L \circ \mathscr{V}$ would give that $V_j \subseteq U_i$ for some $V_j \in \mathscr{V}$ and we can take $L = V_j$. Therefore $F''/\rho^\diamond$ is isomorphic to $A^\diamond$.

Example 1. The distributivity (3) is not true for $w \in A^\diamond$ in general. Indeed, let $a, b \in A$ be different and let $L = \{aa, bb\}$. Then $\lambda \in L \circ (a \cdot (a \vee b) \wedge b \cdot (a \vee b))$ but $L \circ ((a \wedge b) \cdot (a \vee b)) = \emptyset$.

4 Canonical Automata

In each level, we consider the canonical finite automaton of a given regular language. To show examples of three types of automata, we consider the language $L = a^+b^+$ over the alphabet $A = \{a,b\}$.

Monoids. We considered the structure $(2^{A^*},A,\circ)$ defined by (4). It is called here the *canonical semiautomaton* on A. Given a regular language L over A, we can generate a subsemiautomaton by L in $(2^{A^*},A,\circ)$ called the *canonical finite semiautomaton* of L; namely

$$\mathscr{D}_L = (\,\{\, L\circ u \mid u \in A^*\,\},A,\circ\,).$$

It is really finite due to Proposition 4. Notice that $L\circ u = u^{-1}L$ for all $u \in A^*$. Taking L as the unique initial state and $T = \{\,L\circ u \mid \lambda \in L\circ u\,\}$ as the set of all final states, we get the *canonical finite automaton* of L.

Proposition 4 ([13]). *Given a regular language L over the alphabet A, the automaton $\mathsf{D}_L = (\,\{u^{-1}L \mid u \in A^*\},A,\circ,L,T\,)$ is finite and accepts L.*

For the sake of the completeness we prove this result in Appendix.

Example 2. In the canonical finite automaton D_L of the language $L = a^+b^+$, we have four states $L = a^+b^+$, $K = a^{-1}L = a^*b^+$, $b^{-1}L = \emptyset$ and $b^{-1}K = b^*$. There is just one state containing the empty word, namely the state b^*. Thus $T = \{b^*\}$. The automaton is depicted on Figure 1.

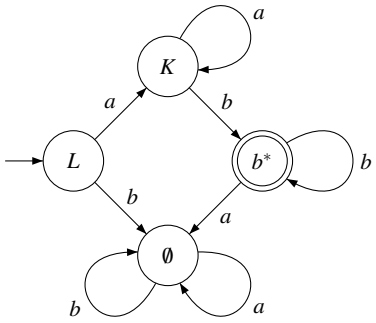


Fig. 1 The canonical finite automaton of the language $L = a^+b^+$.

Semirings. The structure $(2^{A^*},A,\circ,\cap)$ forms the *canonical meet semiautomaton* on A. Moreover, given a regular language L over A, we can generate by L in $(2^{A^*},A,\circ,\cap)$ the *canonical finite meet semiautomaton* of L; namely

$$\mathscr{M}_L = (\,\{L\circ U \mid U \in A^{\square}\,\},A,\circ,\cap\,).$$

Taking L as the unique initial state and all states containing λ as the set of all final states, we get the *canonical finite meet automaton* M_L of L.

Example 3. To construct the canonical finite meet automaton M_L of the language $L = a^+b^+$ we need to consider all possible intersections of states from $\mathscr{D}_L$. There are two new states: the intersection $K \cap b^* = b^+$ and the intersection of the empty system $\bigcap_\emptyset = A^*$. The canonical finite meet automaton is depicted on Figure 2.

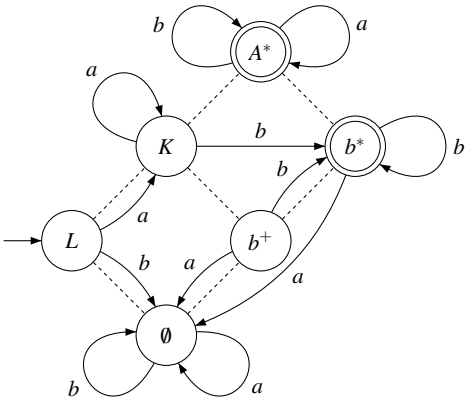


Fig. 2 The canonical finite meet automaton of the language $L = a^+b^+$.

Dashed lines indicate the inclusion relation on the set of all states. The inclusion relation completely describes a semilattice structure of the meet automaton M_L.

Lattice algebras. The structure $(2^{A^*}, A, \circ, \cap, \cup)$ forms the *canonical lattice semiautomaton* on A. Moreover, given a regular language L over A, we can generate by L in $(2^{A^*}, A, \circ, \cap, \cup)$ the *canonical finite lattice semiautomaton* of L; namely

$$\mathscr{L}_L = (\{L \circ \mathscr{U} \mid \mathscr{U} \in A^\diamond\}, A, \circ, \cap, \cup).$$

This structure is already mentioned in Klíma [5]. Taking L as the unique initial state and all states containing λ as the set of all final states, we get the *canonical finite lattice automaton* L_L of L.

Example 4. We consider the canonical finite lattice automaton L_L of the language $L = a^+b^+$, which is depicted on Figure 3. There is only one new state, namely $K^\lambda = K \cup b^* = K \cup \{\lambda\}$ in addition to the canonical finite meet automaton M_L. Now, the inclusion relation describes a lattice structure of L_L.

5 Syntactic structures

The basic tool of the algebraic language theory is the concept of the syntactic monoid of a regular language. It is a certain finite quotient of the free monoid on the

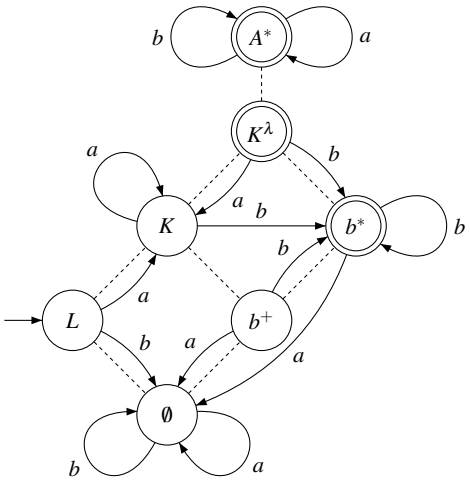


Fig. 3 The canonical finite lattice automaton of the language $L = a^+b^+$.

corresponding alphabet. We recall here its definition and its construction. Then we consider modifications for the remaining two levels.

Monoids. Given a regular language L over the alphabet A, we define the *syntactic congruence* $\sim_L^*$ of L on A^* as follows: for $u, v \in A^*$, put $u \sim_L^* v$ if and only if

$$(\forall\, p, q \in A^*)\ (\ puq \in L \Longleftrightarrow pvq \in L\ )\,.$$

The following is a folklore result.

Proposition 5. *The relation $\sim_L^*$ is a congruence relation on A^*. Moreover, for $u, v \in A^*$, we have that $u \sim_L^* v$ if and only if*

$$(\forall\ p \in A^*)\ (p^{-1}L) \circ u = (p^{-1}L) \circ v\,.$$

Therefore, the structure $A^*/\sim_L^*$ is isomorphic to the transformation monoid of the canonical finite semiautomaton of L.

We present here a proof since it is a suitable preparation for similar results in the next levels.

Proof. Clearly, the relation $\sim_L^*$ is reflexive, symmetric and transitive. Furthermore, for $u, v, w \in A^*$, if $u \sim_L^* v$ then $uw \sim_L^* vw$ and $wu \sim_L^* wv$. Clearly, the fact $u \sim_L^* v$ is equivalent to $(\forall\ p, q \in A^*)(\ q \in (pu)^{-1}L \Longleftrightarrow q \in (pv)^{-1}L\ )$, which is $(\forall\ p \in A^*)\,(pu)^{-1}L = (pv)^{-1}L$, that is $(\forall\ p \in A^*)\,(p^{-1}L) \circ u = (p^{-1}L) \circ v$.

The structure $A^*/\sim_L^*$ is called the *syntactic monoid* of L.

Semirings. Given a regular language L over the alphabet A, we define the *syntactic (semiring) congruence* $\sim_L^{\Box}$ of L on $A^{\Box}$ as follows: For $U = \{u_1, \ldots, u_k\}, V = \{v_1, \ldots, v_\ell\} \in A^{\Box}$, we put $U \sim_L^{\Box} V$ if and only if

$$(\forall\, p,q \in A^*)\ (\ pu_1q \in L, \ldots, pu_kq \in L \Longleftrightarrow pv_1q \in L, \ldots, pv_\ell q \in L\ ).$$

Proposition 6 ([11]). *The relation $\sim_L^{\Box}$ is a congruence relation on $A^{\Box}$. Moreover, for $U,V \in A^{\Box}$, we have that $U \sim_L^{\Box} V$ if and only if*

$$(\forall\, p \in A^*)\ (p^{-1}L) \circ U = (p^{-1}L) \circ V\,.$$

Proof. To show that the relation $\sim_L^{\Box}$ is a congruence relation on $A^{\Box}$ is easy and similar to the case of monoids. Clearly, the fact $U \sim_L^{\Box} V$ is equivalent to

$$(\forall\, p,q \in A^*)\ q \in (pu_1)^{-1}L \cap \cdots \cap (pu_k)^{-1}L \Longleftrightarrow q \in (pv_1)^{-1}L \cap \cdots \cap (pv_\ell)^{-1}L.$$

The last formula can be written as

$$(\forall\, p \in A^*)\, (pu_1)^{-1}L \cap \cdots \cap (pu_k)^{-1}L = (pv_1)^{-1}L \cap \cdots \cap (pv_\ell)^{-1}L,$$

which is $(\forall\, p \in A^*)\ p^{-1}L \circ U = p^{-1}L \circ V$.

Note that one can show (see [10]) that the structure $A^{\Box}/\sim_L^{\Box}$ is isomorphic to the transformation semiring of the whole canonical finite meet semiautomata M_L of L. The structure $A^{\Box}/\sim_L^{\Box}$ is called the *syntactic semiring* of L.

Numerous examples of syntactic semirings can be found e.g. in [10]. In [11] it is described how one can compute the syntactic semiring algorithmically from the syntactic monoid. For the handmade computations we can use Proposition 6.

Example 5. Consider again the language $L = a^+b^+$. We can choose the words λ, a, b, ab, and ba to represent five different transformations. There are no others, because both a and b are idempotent elements of both syntactic monoid and syntactic semiring and ba is a zero element. Moreover, ba is the smallest element in the syntactic semiring, because ba transforms all states, with exception of A^*, to the state $\emptyset$. So, if we want to compute all elements of the syntactic semiring, it is enough to consider only intersections of the elements λ, a, b and ab. The crucial observation is that both $\lambda \wedge ab$ and $a \wedge b$ give the same transformation as well as the intersection of any triple of elements. Hence in the syntactic semiring there are, besides the element $\top$ and elements λ, a, b, ab, and ba, just five elements given by intersections $\lambda \wedge a$, $\lambda \wedge b$, $\lambda \wedge ab$, $a \wedge ab$ and $b \wedge ab$. In Table 1 we present how all these elements transform the canonical finite meet automaton. The semilattice part of the syntactic semiring is fully described by Figure 4. Notice that for the computation of the syntactic semiring we do not need to know all the information from Table 1. For example, if a term $U \in F'$ acts on the state b^+, then the image is the intersection of images of the states K and b^*. Moreover, the images of states $\emptyset$ and A^* are clear. Thus we need to work only with first three columns.

	L	K	b^*	b^+	A^*	$\emptyset$
λ	L	K	b^*	b^+	A^*	$\emptyset$
a	K	K	$\emptyset$	$\emptyset$	A^*	$\emptyset$
b	$\emptyset$	b^*	b^*	b^*	A^*	$\emptyset$
ab	b^*	b^*	$\emptyset$	$\emptyset$	A^*	$\emptyset$
ba	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	A^*	$\emptyset$
$\top$	A^*	A^*	A^*	A^*	A^*	A^*
$\lambda \wedge a$	L	K	$\emptyset$	$\emptyset$	A^*	$\emptyset$
$\lambda \wedge b$	$\emptyset$	b^+	b^*	b^+	A^*	$\emptyset$
$\lambda \wedge ab$	$\emptyset$	b^+	$\emptyset$	$\emptyset$	A^*	$\emptyset$
$a \wedge ab$	b^+	b^+	$\emptyset$	$\emptyset$	A^*	$\emptyset$
$b \wedge ab$	$\emptyset$	b^*	$\emptyset$	$\emptyset$	A^*	$\emptyset$

Table 1 The transformations of $\mathscr{M}_L$ for the language $L = a^+b^+$.

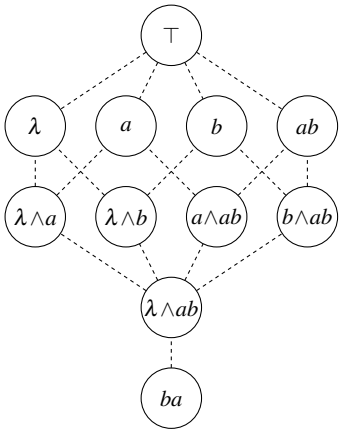


Fig. 4 The semilattice order of the syntactic semiring of the language $L = a^+b^+$.

Lattice algebras. Given a regular language L over the alphabet A, we define the so-called *syntactic (lattice) congruence* $\sim_L^\diamond$ of L on $A^\diamond$ as follows: for $\mathscr{U} = \{U_1, \ldots, U_k\}, \mathscr{V} = \{V_1, \ldots, V_\ell\} \in A^\diamond$ we put $\mathscr{U} \sim_L^\diamond \mathscr{V}$ if and only if, for every $p, q \in A^*$, the condition

$$pU_1q \subseteq L \text{ or } \ldots \text{ or } pU_kq \subseteq L$$

is equivalent to

$$pV_1q \subseteq L \text{ or } \ldots \text{ or } pV_\ell q \subseteq L.$$

Proposition 7. *The relation $\sim_L^\diamond$ is a congruence relation on $A^\diamond$. Moreover, for $\mathscr{U}, \mathscr{V} \in A^\diamond$, it holds that $\mathscr{U} \sim_L^\diamond \mathscr{V}$ if and only if*

$$(\forall\, p \in A^*)\ p^{-1}L \circ \mathscr{U} = p^{-1}L \circ \mathscr{V}.$$

Proof. To show that the relation $\sim_L^\diamond$ is a congruence relation on $A^\diamond$ is easy and similar to the case of monoids.

Let $\mathscr{U}, \mathscr{V} \in A^\diamond$ are of the form

$$\mathscr{U} = \{U_1, \ldots, U_k\}, \text{ where } U_1 = \{u_{1,1}, \ldots, u_{1,r_1}\}, \ldots, U_k = \{u_{k,1}, \ldots, u_{k,r_k}\},$$

$$\mathscr{V} = \{V_1, \ldots, V_\ell\}, \text{ where } V_1 = \{v_{1,1}, \ldots, v_{1,s_1}\}, \ldots, V_\ell = \{v_{\ell,1}, \ldots, v_{\ell,s_\ell}\}.$$

Clearly, $\mathscr{U} \sim_L^\diamond \mathscr{V}$ is equivalent to $(\forall\, p, q \in A^*)$

$$q \in ((pu_{1,1})^{-1}L \cap \cdots \cap (pu_{1,r_1})^{-1}L) \cup \cdots \cup ((pu_{k,1})^{-1}L \cap \cdots \cap (pu_{k,r_k})^{-1}L)$$

$$\iff q \in (pv_{1,1})^{-1}L \cap \cdots \cap (pv_{1,s_1})^{-1}L \cup \cdots \cup (pv_{\ell,1})^{-1}L \cap \cdots \cap (pv_{\ell,s_\ell})^{-1}L,$$

which is

$$(\forall\, p \in A^*)\ ((pu_{1,1})^{-1}L \cap \cdots \cap (pu_{1,r_1})^{-1}L) \cup \cdots \cup ((pu_{k,1})^{-1}L \cap \cdots \cap (pu_{k,r_k})^{-1}L)$$

$$= ((pv_{1,1})^{-1}L \cap \cdots \cap (pv_{1,s_1})^{-1}L) \cup \cdots \cup ((pv_{\ell,1})^{-1}L \cap \cdots \cap (pv_{\ell,s_k})^{-1}L),$$

that is $(\forall\, p \in A^*)\ p^{-1}L \circ \mathscr{U} = p^{-1}L \circ \mathscr{V}$.

The structure $A^\diamond / \sim_L^\diamond$ is called the *syntactic lattice algebra* of L.

Note that in this third level it is not true that the structure $A^\diamond / \sim_L^\diamond$ is isomorphic to the transformation lattice algebra of the whole canonical lattice semiautomaton $\mathscr{L}_L$ of L as mentioned in the next example.

Example 6. Now we present the syntactic lattice algebra of the language $L = a^+b^+$.

First of all, we could mentioned an interesting fact: the terms $\lambda \wedge ab$ and $a \wedge b$ transform $\mathscr{L}_L$ in a different way, namely $K^\lambda \circ (\lambda \wedge ab) = b^*$ and $K^\lambda \circ (a \wedge b) = b^+$. However these two terms $\lambda \wedge ab$ and $a \wedge b$ give the same element in the syntactic semiring of L, because they transform the states from $\mathscr{D}_L$ in the same way. In other words, $\lambda \wedge ab\, \rho_L^\diamond\, a \wedge b$. This example just recalls the observation from Proposition 7, that we need to check the images of the three states L, K and b^* only.

We can start from the syntactic semiring of L, since the syntactic lattice algebra can be viewed as an extension of the syntactic semiring by adding joins. Thus we need to compute joins of all elements described in Table 1. This can be done by a brute force algorithm, which gives Table 2.

To see that the computation is complete, we have to add some basic observations. At first, one can check the following equalities $\lambda = (\lambda \wedge a) \vee (\lambda \wedge b)$, $a = (\lambda \wedge a) \vee (a \wedge ab)$ and $b = (\lambda \wedge b) \vee (b \wedge ab)$. Therefore we can remove elements λ, a and b from the generating set. Since the elements $\top$, $\lambda \wedge ab$ and ba are comparable with the others, we do not obtain new elements adding these element into the joins. Thus, we need to compute the joins for five elements $\lambda \wedge a$, $\lambda \wedge b$, $a \wedge ab$ $b \wedge ab$ and ab.

We observe that $a \wedge ab$ transforms the state L to b^+, and that no other image of L under applications $\lambda \wedge a$, $\lambda \wedge b$, $b \wedge ab$ contains the word b. This means that $a \wedge ab$ can not be covered by a join of elements $\lambda \wedge a$, $\lambda \wedge b$, $b \wedge ab$. In the similar way, $b \wedge ab$ transforms K to b^* which contains λ, and therefore $b \wedge ab$ can not be covered

	L	K	b^*
λ	L	K	b^*
a	K	K	$\emptyset$
b	$\emptyset$	b^*	b^*
ab	b^*	b^*	$\emptyset$
ba	$\emptyset$	$\emptyset$	$\emptyset$
$\top$	A^*	A^*	A^*
$\lambda \wedge a$	L	K	$\emptyset$
$\lambda \wedge b$	$\emptyset$	b^+	b^*
$\lambda \wedge ab$	$\emptyset$	b^+	$\emptyset$
$a \wedge ab$	b^+	b^+	$\emptyset$
$b \wedge ab$	$\emptyset$	b^*	$\emptyset$

	L	K	b^*
$(\lambda \wedge a) \vee (b \wedge ab)$	L	K^λ	$\emptyset$
$(\lambda \wedge a) \vee b$	L	K^λ	b^*
$(\lambda \wedge a) \vee ab$	K^λ	K^λ	$\emptyset$
$(\lambda \wedge b) \vee (a \wedge ab)$	b^+	b^+	b^*
$(\lambda \wedge b) \vee a$	K	K^λ	b^*
$(\lambda \wedge b) \vee ab$	b^*	b^*	b^*
$(a \wedge ab) \vee (b \wedge ab)$	b^+	b^*	$\emptyset$
$(a \wedge ab) \vee \lambda$	K	K	b^*
$(a \wedge ab) \vee b$	b^+	b^*	b^*
$(b \wedge ab) \vee a$	K	K^λ	$\emptyset$
$\lambda \vee ab$	K^λ	K^λ	b^*

Table 2 The transformations of $\mathscr{L}_L$ for the language $L = a^+b^+$.

by a join of elements $\lambda \wedge a$, $\lambda \wedge b$, $a \wedge ab$. To see that both $\lambda \wedge a$ and $\lambda \wedge b$ can not be covered by a join of the others elements from the following ones $\lambda \wedge a$, $\lambda \wedge b$, $a \wedge ab$ $b \wedge ab$, ab, we just mention that $K \circ (\lambda \wedge a) = K$ contains ab and $b^* \circ (\lambda \wedge b) = b^*$ contains λ.

From the observations from the previous paragraph we can state that joins of elements $\lambda \wedge a, \lambda \wedge b, a \wedge ab, b \wedge ab$ are pairwise different elements of the syntactic lattice algebra of L. So we obtain 15 elements in this way. If we add the element ab into some of these joins, then we can remove from this join both $a \wedge ab$ and $b \wedge ab$ if they occur. So, we obtain additionally four elements ab, $ab \vee (\lambda \wedge a)$, $ab \vee (\lambda \wedge b)$, $ab \vee (\lambda \wedge a) \vee (\lambda \wedge b) = ab \vee \lambda$.

Altogether, the syntactic lattice of L consists of 22 elements (see Figure 5): $\perp =$ ba, $\lambda \wedge ab$, 15 elements described above as joins of elements $\lambda \wedge a, \lambda \wedge b, a \wedge ab, b \wedge ab$, and finally the elements ab, $ab \vee (\lambda \wedge a)$, $ab \vee (\lambda \wedge b)$, $ab \vee \lambda$, $\top$.

6 General algebras

The Eilenberg like theorems establish bijections between certain varieties of regular languages and pseudovarieties of certain algebraic systems. Not every finite monoid is isomorphic to a syntactic one, we have to generate the appropriate pseudovariety. Similarly in remaining levels.

Monoids. Here one considers varieties of languages and pseudovarieties of finite monoids. The Eilenberg theorem can be find in e.g. [8].

Semirings. Here one considers the so-called conjunctive varieties of languages and pseudovarieties of finite semirings. For more details see e.g. [10].

Lattice algebras. The following new definition of a notion of lattice algebras is a part of an effort of formulation of Eilenberg like theorem using the notion of

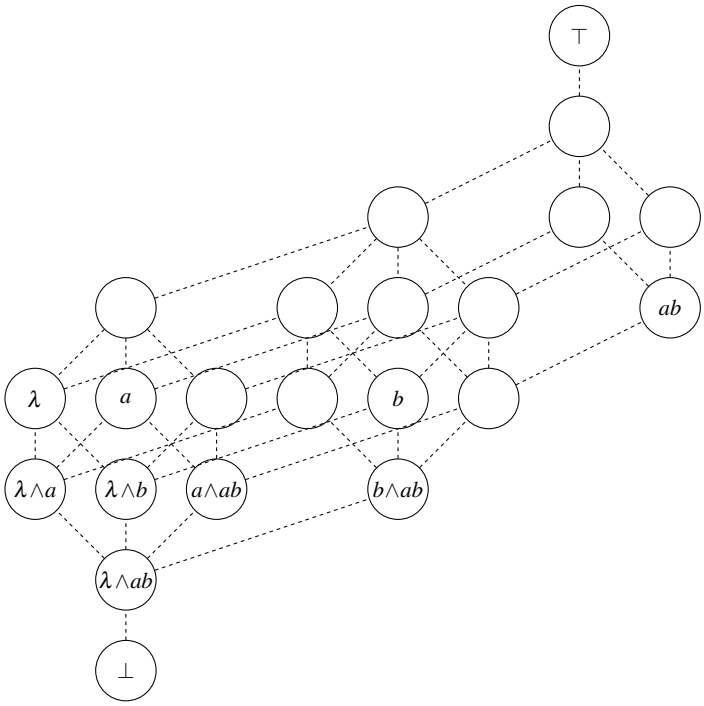


Fig. 5 The order of the syntactic lattice algebra of the language $L = a^+b^+$.

syntactic lattice algebra. Such a theorem is not formulated or even proved in this paper. Nevertheless, we try here to characterize the finite factors of $A^\diamond$.

A *lattice algebra* is 8-tuple $(K,\wedge,\vee,\cdot,P,\bot,\top,1)$ where $(K,\wedge,\vee)$ is a bounded distributive lattice with the bottom element $\bot$ and the top element $\top$, $(K,\cdot,1)$ is a monoid with right zero elements $\bot$ and $\top$, P is a finite subset of K such that the lattice $(K,\wedge,\vee)$ is generated by the set of all products of elements from P and $\top\cdot p=\top$ and $\bot\cdot p=\bot$ hold for $p\in P$, and finally such that the distributivities

$$q\cdot(r\wedge s)=q\cdot r\wedge q\cdot s,\;\; q\cdot(r\vee s)=q\cdot r\vee q\cdot s,$$

$$(q\wedge r)\cdot p=q\cdot p\wedge r\cdot p,\;\; (q\vee r)\cdot p=q\cdot p\vee r\cdot p$$

hold for all $q,r,s\in K$ and $p\in P$.

Notice that, considering $A^\diamond$, take P equal to the image of A, $\top$ the image of $\{\emptyset\}$ and $\bot$ the image of $\emptyset$.

7 Characterizing Reversible Languages

We consider the class of all reversible languages (see Golovkins and Pin [2]). We present them using the Ambainis and Freivalds condition (see [1]).

Proposition 8 ([1, 2]). *Let L be a regular language over an alphabet A. Then L is recognized by a reversible automaton if and only if the following condition for the canonical automaton of L holds:*

$$(\forall x,y \in A^*, f,g \in Q)\ f \neq g, f \circ x = g = g \circ x \implies g \circ y = g\,. \tag{7}$$

Note that a condition from the previous statement is usually formulated in a different way, namely that the canonical automaton of L does not contain the following configuration, with $f \neq g \neq h$.

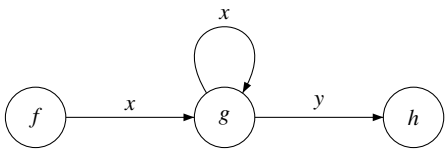


Fig. 6 The forbidden configuration for reversible language.

In [2] the Ambainis-Freivalds condition (7) for the language L was translated to a certain algebraic condition concerning the syntactic monoid of L together with the image of L in the syntactic homomorphism. They also mention that this class is not closed under binary intersections nor unions. Therefore it is not an instance of any known Eilenberg correspondence.

Here we show an equivalent condition which is, in some sense, an identity for the canonical lattice algebra of the considered language. We need the following classical notion from the semigroup theory. Each element s in a finite semigroup has a unique idempotent element among its powers, which is denoted by s^ω. So we use this notation for lattice algebra, where this operation $(\_)^\omega$ is related to the operation of multiplication. Moreover, in a fixed finite semigroup S, one can find natural number m such that $s^\omega = s^m$ for every element $s \in S$.

We are not going to define here the notion of an identity for (finite) lattice algebras in a full generality. Nevertheless, we use the concrete condition

$$x^\omega y \vee (x^\omega z \wedge t) = x^\omega y \vee (x^\omega t \wedge z)\ . \tag{8}$$

It is *valid* in the syntactic lattice algebra $\mathscr{L}_L$ of the language L if we get the same element of (8) on left and right sides after substituting $p \sim_L^\diamond$, $u \sim_L^\diamond$, $v \sim_L^\diamond$ and $w \sim_L^\diamond$ ($p,u,v,w \in A^*$), for x,y,z and t, respectively.

Proposition 9. *The canonical automaton of a regular language L satisfies (7) if and only if the canonical lattice algebra of L satisfies condition (8).*

Proof. To simplify notation we write simply u instead of $u\sim_L^\diamond$, for any $u\in A^*$. This simplification does not lead to a confusion, because for a state of K of the canonical semiautomaton of a language L, by $K\circ(U\sim_L^\diamond)$ is meant $K\circ U$.

Let L be a regular language with the canonical automaton satisfying the condition (7). Let $p,u,v,w\in A^*$ be arbitrary words and denote $\mathscr{U}=p^\omega u\vee(p^\omega v\wedge w), \mathscr{V}=p^\omega u\vee(p^\omega w\wedge v)$, both from $A^\diamond/\sim_L^\diamond$. Furthermore, let $s\in A^*$ be an arbitrary word and consider the state $K=s^{-1}L$ in the canonical automaton of L. We need to show that $K\circ\mathscr{U}=K\circ\mathscr{V}$. At first, assume that $K\circ p^\omega=K$. Then $K\circ\mathscr{U}=K\circ u\cup(K\circ v\cap K\circ w)$ which is equal to $K\circ\mathscr{V}$. Assume now that $K\circ p^\omega\neq K$, particularly $K\circ p\neq K$. From the definition of p^ω we know that $(K\circ p^\omega)\circ p^\omega=K\circ p^\omega$. Since the canonical semiautomaton $\mathscr{D}_L$ satisfies (7), we get that $(K\circ p^\omega)\circ y=K\circ p^\omega$ for every $y\in A^*$. Therefore, $K\circ\mathscr{U}=K\circ p^\omega\cup(K\circ p^\omega\cap K\circ w)=K\circ p^\omega$ and similarly we obtain $K\circ\mathscr{V}=K\circ p^\omega\cup(K\circ p^\omega\cap K\circ v)=K\circ p^\omega$.

To prove the opposite implication, we consider a regular language L which has the forbidden configuration in its canonical semiautomaton $\mathscr{D}_L$ and then we show that its canonical lattice algebra does not satisfy (8). Let f,g,h be states in $\mathscr{D}_L$ and $x,y\in A^*$ be words such that $f\neq g\neq h$, $f\circ x=g=g\circ x$ and $h=g\circ y$. Recall that $f,g,h\subseteq A^*$, because there are left quotient of L. Since $f\neq g$, there is a word $s\in A^*$ such that $s\in f$, $s\notin g$ or $s\notin f$, $s\in g$. Note that the condition $s\in f$ is equivalent to $\lambda\in s^{-1}f$, i.e. $\lambda\in f\circ s$. Similarly, since $g\neq h$, there is a word r such that $r\in g$, $r\notin h$ or $r\notin g$, $r\in h$. Thus there are four cases to be discussed. In all these cases, the word $p=x$ is already fixed by the forbidden configuration. For this p, we have $f\circ p^\omega=g$.

Case I) If $s\in f$, $s\notin g$, $r\in g$, $r\notin h$ then we put $u=s$, $v=r$ and $w=s$ and consequently we denote $\mathscr{U}=p^\omega u\vee(p^\omega v\wedge w)$, $\mathscr{V}=p^\omega u\vee(p^\omega w\wedge v)$. Now we see that

$$\begin{aligned}&\lambda\notin g\circ s=(f\circ p^\omega)\circ s=f\circ(p^\omega u)=f\circ(p^\omega w),\\&\lambda\in g\circ r=(f\circ p^\omega)\circ r=f\circ(p^\omega v)\text{ and}\\&\lambda\in f\circ s=f\circ w.\end{aligned}$$

Therefore, $\lambda\in f\circ p^\omega u\cup(f\circ p^\omega v\cap f\circ w)=f\circ\mathscr{U}$ and $\lambda\notin f\circ p^\omega u\cup(f\circ p^\omega w\cap f\circ v)=f\circ\mathscr{V}$. Hence $f\circ\mathscr{U}\neq f\circ\mathscr{V}$ and $\mathscr{U},\mathscr{V}$ are different elements in the canonical lattice algebra of L.

Case II) If $s\in f$, $s\notin g$, $r\notin g$, $r\in h$ then we put $u=s$, $v=yr$ and $w=s$ and again $\mathscr{U}=p^\omega u\vee(p^\omega v\wedge w)$, $\mathscr{V}=p^\omega u\vee(p^\omega w\wedge v)$. Now we have

$$\begin{aligned}&\lambda\notin g\circ s=(f\circ p^\omega)\circ s=f\circ(p^\omega u)=f\circ(p^\omega w),\\&\lambda\in h\circ r=(f\circ p^\omega y)\circ r=f\circ(p^\omega v)\text{ and}\\&\lambda\in f\circ s=f\circ w.\end{aligned}$$

Therefore, $\lambda\in f\circ p^\omega u\cup(f\circ p^\omega v\cap f\circ w)=f\circ\mathscr{U}$ and $\lambda\notin f\circ p^\omega u\cup(f\circ p^\omega w\cap f\circ v)=f\circ\mathscr{V}$. This means that $\mathscr{U}\neq\mathscr{V}$ in the canonical lattice algebra of L.

Case III) If $s\notin f$, $s\in g$, $r\in g$, $r\notin h$ then we put $u=yr$, $v=s$, $w=pr$, $\mathscr{U}=p^\omega u\vee(p^\omega v\wedge w)$ and $\mathscr{V}=p^\omega u\vee(p^\omega w\wedge v)$. Now we have

$$\lambda \notin h \circ r = (g \circ y) \circ r = g \circ u = (f \circ p^{\omega}) \circ u = f \circ (p^{\omega} u),$$
$$\lambda \in g \circ s = g \circ v = (f \circ p^{\omega}) \circ v = f \circ (p^{\omega} v),$$
$$\lambda \in g \circ r = (f \circ p) \circ r = f \circ w \text{ and}$$
$$\lambda \notin f \circ s = f \circ v.$$

Hence, $\lambda \in f \circ p^{\omega} u \cup (f \circ p^{\omega} v \cap f \circ w) = f \circ \mathscr{U}$ and $\lambda \notin f \circ p^{\omega} u \cup (f \circ p^{\omega} w \cap f \circ v) = f \circ \mathscr{V}$.

Case IV) If $s \notin f$, $s \in g$, $r \notin g$, $r \in h$ then we put $u = r$, $v = s$ and $w = ps$ and consequently $\mathscr{U} = p^{\omega} u \vee (p^{\omega} v \wedge w)$, $\mathscr{V} = p^{\omega} u \vee (p^{\omega} w \wedge v)$. Now we see that

$$\lambda \notin g \circ r = g \circ u = (f \circ p^{\omega}) \circ u = f \circ (p^{\omega} u),$$
$$\lambda \in g \circ s = g \circ v = (f \circ p^{\omega}) \circ v = f \circ (p^{\omega} v),$$
$$\lambda \in g \circ s = (f \circ p) \circ s = f \circ w \text{ and}$$
$$\lambda \notin f \circ s = f \circ v.$$

And we can finish this case in the same manner as the previous ones.

8 Appendix

8.1 The Free Bounded Distributive Lattice

The free (bounded) distributive lattice over a finite set is well-known – see i.e. Grätzer [3]. Here we describe how to get a free distributive lattice over an arbitrary set.

Proposition 10. *Let X be an arbitrary set. The set $\overline{X}$ of all elements*

$$\{\{u_{1,1}, \ldots, u_{1,r_1}\}, \ldots, \{u_{k,1}, \ldots, u_{k,r_k}\}\}, \tag{9}$$

$$k, r_1, \ldots, r_k \geq 0,\ u_{i,j} \in X \text{ for } i = 1, \ldots, k,\ j = 1, \ldots, r_i,$$

with the inner sets incomparable is the carrier of a free bounded distributive lattice over X. The interpretation of the above element is

$$(u_{1,1} \wedge \cdots \wedge u_{1,r_1}) \vee \cdots \vee (u_{k,1} \wedge \cdots \wedge u_{1,r_k}). \tag{10}$$

This tells us how to apply the lattice operations. Notice that $\{\emptyset\}$ is the greatest element and $\emptyset$ is the smallest one.

Proof. We present here the elements of the free object by canonical forms.

Taking a term over X with respect to operational symbols $\wedge$ and $\vee$ and using the laws of associativity, idempotency, commutativity and distributivity, we can get a term of the form (10). If two inner set are comparable, we can omit the richer one.

We show that two different elements of $\overline{X}$ differ by an appropriate evaluation of variables in the two element lattice $\{0,1\}$, $0<1$. So let $\mathscr{U}$ and $\mathscr{V}$ be two different canonical forms. We assume that

$$\mathscr{U}=\{U_1,\ldots,U_k\},\ U_1=\{u_{1,1},\ldots,u_{1,r_1}\},\ldots,U_k=\{u_{k,1},\ldots,u_{k,r_k}\},$$

$$\mathscr{V}=\{V_1,\ldots,V_\ell\},\ V_1=\{v_{1,1},\ldots,v_{1,s_1}\},\ldots,V_\ell=\{v_{\ell,1},\ldots,v_{\ell,s_\ell}\},$$

There exists $i\in\{1,\ldots,k\}$ such that $U_i\notin\mathscr{V}$ (the case of the $V_i\notin\mathscr{U}$ can be treated similarly). Send $u_{i,1},\ldots,u_{i,r_i}$ to 1 and all other variables to 0. Then $\mathscr{U}$ goes to 1 and $\mathscr{V}$ goes to 1 only if there is a proper subset $V_{i'}$ of U_i. In this case, send all elements of $V_{i'}$ to 1 and all other variables to 0. Then $\mathscr{V}$ goes to 1 and it is impossible that there is some $U_{i''}\in\mathscr{U}$ such that $U_{i''}\subseteq V_{i'}$. Thus $\mathscr{U}$ goes to 0.

Thus the elements of $\overline{X}$ can be treated as the canonical forms.

8.2 Proof of Proposition 4

Proposition 4 ([13]). *Given a regular language L over the alphabet A, the automaton $\mathsf{D}_L=(\{u^{-1}L\mid u\in A^*\},A,\circ,L,T)$ is finite and accepts L.*

Proof. Take a finite automaton $(Q,A,\cdot,i,T')$ accepting L. For $q\in Q$, put $L_q=\{v\in A^*\mid q\cdot v\in T'\}$. In particular $L_i=L$. Notice that, for each $q\in Q$ and $a\in A$, we have $L_{q\cdot a}=a^{-1}L_q$. Furthermore, in D_L we have that, for each $u\in A^*$, it holds $u^{-1}L=L_{i\cdot u}$ and therefore there are only finitely many $(u^{-1}L)$'s.

Now the automaton D_L accepts a word u if and only if $\lambda\in L\circ u=u^{-1}L$, that is $u\in L$.

References

1. A. Ambainis and R. Freivalds, *1-Way Quantum Finite Automata: Strengths, Weaknesses and Generalizations*, In Proc. FOCS 1998, pp. 332-341 (1998).
2. M. Golovkins and J.-É. Pin, *Varieties generated by certain models of reversible finite automata*, Chicago Journal of Theoretical Computer Science **2**, (2010).
3. G. Grätzer, *General Lattice Theory*, Second edition, Birkhäuser (2003).
4. S. Eilenberg, *Automata, Languages and Machines*, Vol. B, Academic Press (1976).
5. O. Klíma, *On varieties of automata enriched with an algebraic structure (Extended abstract)*, In Proc. AFL 2014, EPTCS 151, pp 49-54 (2014), arXiv:1405.5272.
6. O. Klíma and L. Polák, *Hierarchies of piecewise testable languages*, In Proc. Developments Language Theory 2008, LNCS 5257, pp 479-490 (2008).
7. J.-É. Pin, *A variety theorem without complementation*, Russian Mathem. (Iz. VUZ) **39**, pp 74-83 (1995).
8. J.-É. Pin, *Syntactic Semigroups*, Chapter 10 in Handbook of Formal Languages, G. Rozenberg and A. Salomaa eds, Springer (1997).
9. J.-É. Pin, *Varieties of Formal Languages*, North Oxford Academic, Plenum (1986).
10. L. Polák, *Syntactic semiring of a language*, In Proc. Mathematical Foundations of Computer Science 2001, LNCS 2136, pp 611–620 (2001).

11. L. Polák, *Syntactic semiring and universal automaton*, In Proc. Developments Language Theory, Szeged 2003, LNCS 2710, pp 411-422 (2003).
12. Straubing, H.: *On logical descriptions of regular languages*, In Proc. LATIN 2002, Springer Lecture Notes in Computer Science, Vol. 2286, pp 528-538 (2002).
13. Yu, S.: *Regular languages*, Chapter 2 in Handbook of Formal Languages. G. Rozenberg and A. Salomaa eds, Springer (1997).

Improving witnesses for state complexity of catenation combined with boolean operations

P. Caron, J.-G. Luque, B. Patrou

Université de Rouen, France, {Pascal.Caron, Jean-Gabriel.Luque, Bruno.Ptrrou}@univ-rouen.fr

Abstract

We propose a common 3-letters witness for state complexity of catenation combined with union and catenation combined with intersection proving two conjectures of Brzozowski. We use some combinatorial tools to prove that this witness does not fit for the combination of the catenation with the symmetric difference.

1 Introduction

Cui *et al.* [3] compute the state complexity for the combination of catenation with union and intersection operations. They build a witness over a 3-letters alphabet for the $\cup$-case and over a 4-letters alphabet for the $\cap$-case.

Furthermore, Brzozowski [1] proposes a family of automata which should be good candidates for witnesses for many combinations of operations. In his work, several conjectures are stated. In particular, he proposes a common 4-letters witness for the previous combinations. An equivalent work for the combination of catenation with symmetric difference is done in [2] where a 4-letters Brzozowski witness is built.

In this paper, we give a Brozowski witness with only 3 letters which improves conjecture 18 and 19 of [1]. It is still common for both combinations and also improves the result of Cui *et al.* [3].

We also prove, using some combinatorial objects, this witness does not suit for the symmetric difference operation.

Section and give some definitions and notations about automata and combinatoric. In section , we show that every state of our witness is accessible. Section , is devoted to separability of pair of states for both combinations. In Section 1, we explain using some combinatorial tools why our witness fails for the combination of the catenation and the symmetric difference.

2 Preliminaries

For any integer $i \in \mathbb{Z}$, any $p \in \mathbb{N} \setminus \{0\}$, we set $ip = \min\{j \mid j \geq 0 \wedge j \equiv i(p)\}$. Let Σ denotes a finite alphabet. A word w over Σ is a finite sequence of symbols of Σ. The length of w, denoted by $|w|$ is the number of occurrences of symbols of Σ in w. For

$a \in \Sigma$, we denote by $|w|_a$ the number of a in w. The set of all finite words over Σ is denoted by Σ^*. The empty word is denoted by ε. A language is a subset of Σ^*. The set of subsets of a finite set A is denoted by 2^A and $|A|$ denotes the cardinality of A. The symbol $\circ$ denotes any binary boolean operation on languages. In the following, by abuse of notation, we often write q for any singleton $\{q\}$.

A finite automaton (FA) is a 5-tuple $A = (\Sigma, Q, I, F, \cdot)$ where Σ is the input alphabet, Q is a finite set of states, $I \subset Q$ is the set of initial states, $F \subset Q$ is the set of final states and $\cdot$ is the transition function from $Q \times \Sigma$ to 2^Q. An FA is deterministic (DFA) if $|I| = 1$ and for all $q \in Q$, for all $a \in \Sigma$, $|q \cdot a| \leq 1$. The transition function is extended to any word by $q \cdot aw = \bigcup_{q' \in q \cdot a} q' \cdot w$ and $q \cdot \varepsilon = q$ for any symbol a of Σ and any word w of Σ^*. For convenience, we sometimes use the notation $q \xrightarrow{w} q'$ to denote that $q' \in q \cdot w$.

The dual operation is defined by $w \cdot q = \{q' \mid q \in q' \cdot w\}$. We extend the dot notation to any set of states S by $S \cdot w = \bigcup_{s \in S} s \cdot w$ and $w \cdot S = \bigcup_{s \in S} w \cdot s$. A word $w \in \Sigma^*$ is recognized by an FA A if $I \cdot w \cap F \neq \emptyset$.

We assume that all FAs are complete which means that for all $q \in Q$, for all $a \in \Sigma$, $|q \cdot a| \geq 1$. A state q is accessible in an FA if there exists a word $w \in \Sigma^*$ such that $q \in I \cdot w$. The language recognized by an FA A is the set $L(A)$ of words recognized by A. For convenience we will often identify A with $L(A)$. Two automata are said to be equivalent if they recognize the same language.

Let $D = (\Sigma, Q_D, i_D, F_D, \cdot)$ be a DFA. Two states q_1, q_2 of D are equivalent if for any word w of Σ^*, $q_1 \cdot w \in F_D$ if and only if $q_2 \cdot w \in F_D$. Such an equivalence is denoted by $q_1 \sim q_2$. A DFA is minimal if there does not exist any equivalent complete DFA with less states and it is well known that for any DFA, there exists a unique minimal equivalent one [6]. Such a minimal DFA can be obtained from D by computing the accessible part of the automaton $D/\sim = (\Sigma, Q_D/\sim, [i_D], F_D/\sim, \cdot)$ where for any $q \in Q_D$, $[q]$ is the $\sim$-class of the state q and satisfies the property $[q] \cdot a = [q \cdot a]$, for any $a \in \Sigma$. In a minimal DFA, any two distinct states are pairwise inequivalent.

The state complexity of a regular language L denoted by (L) is the number of states of its minimal DFA. Let $\mathscr{L}_n$ be the set of languages of state complexity n. The state complexity of a unary operation $\otimes$ is the function $_\otimes$ associating with an integer n the maximum of the state complexities of $(\otimes L)$ for $L \in \mathscr{L}_n$. A language $L \in \mathscr{L}_n$ is a witness (for $\otimes$) if $(\otimes L) =_\otimes (n)$. This can be generalized, and the state complexity of a k-ary operation $\otimes$ is the k-ary function which associates with any tuple $(n_1, \ldots, n_k)$ the integer $\max\{(\otimes(L_1, \ldots, L_k)) | L_i \in \mathscr{L}_{\setminus}, \forall\rangle \in [\infty, \|]\}$. Then, a witness is a tuple $(L_1, \ldots, L_k) \in (\mathscr{L}_{n_1} \times \cdots \times \mathscr{L}_{n_k})$ such that $(\otimes(L_1, \ldots, L_k)) =_\otimes (n_1, \ldots, n_k)$. An important research area consists in finding witnesses for any $(n_1, \ldots, n_k) \in \mathbb{N}^k$.

The state complexity of an operation defined as a composition of more elementary ones is upper-bounded by the composition of the corresponding elementary state complexities.

For example, consider the ternary operation defined for any three languages L_1, L_2, L_3 by $L_1 \cdot (L_2 \cup L_3)$ and let h be its state complexity. Denote f, g the respective state complexity of catenation and union. Then for any three integers n_1, n_2, n_3,

it holds $h(n_1,n_2,n_3) \leq f(n_1,g(n_2,n_3))$ [5]. In fact, applying the union on a witness does not produce a good candidate for a witness for catenation.

In [1], Brzozowski defines a family of languages that turns to be universal witnesses for several operations. The automata denoting these languages are called *Brzozowski automata*. We need some background to define these automata. We follow the terminology of [4]. Let $Q = \{0,\ldots,n-1\}$ be a set. A *transformation* of the set Q is a mapping of Q into itself. If t is a transformation and i an element of Q, we denote by it the image of i under t. A transformation of Q can be represented by $t = [i_0,i_1,\ldots i_{n-1}]$ which means that $i_k = kt$ for each $0 \leq k \leq n-1$ and $i_k \in Q$. A *permutation* is a bijective transformation on Q. The *identity* permutation of Q is denoted by 1. A *cycle* of length $\ell \leq n$ is a permutation c, denoted by $(i_0,i_1,\ldots i_{\ell-1})$, on a subset $I = \{i_0,\ldots,i_{\ell-1}\}$ of Q where $i_k c = i_{k+1}$ for $0 \leq k < \ell-1$ and $i_{\ell-1}c = i_0$. A *transposition* $t = (i,j)$ is a permutation on Q where $it = j$ and $jt = i$ and for every elements $k \in Q \setminus \{i,j\}$, $kt = k$. A *contraction* $t = \begin{pmatrix} i \\ j \end{pmatrix}$ is a transformation where $it = j$ and for every elements $k \in Q \setminus \{i\}$, $kt = k$. Then, a Brzozowski automaton is a complete DFA $(\Sigma, Q = \{0,\ldots,n-1\}, 0, F = \{n-1\}, \cdot)$, where any letter of Σ induces one of the transformation among transposition, cycle over Q, contraction and identity.

3 Tools

In the following of this paper, let $A = (\Sigma, Q_A, i_A, F_A, \cdot_A)$, $B = (\Sigma, Q_B, i_B, F_B, \cdot_B)$ and $C = (\Sigma, Q_C, i_C, F_C, \cdot_C)$ be any three DFAs with $|Q_A| = m$, $|Q_B| = n$ and $|Q_C| = p$ for any three integers m, n, p. We recall two classical constructions allowing to compute a DFA for the catenation and for any binary boolean operation over two rational languages.

We define the DFA $A \cdot B = (\Sigma, Q, i, F, \cdot)$ as follows :

- $Q = \{(p,S) \mid p \in Q_A, S \subset Q_B\}$
- $i = \begin{cases} (i_A,\emptyset) & \text{if } i_A \notin F_A \\ (i_A,\{i_B\}) & \text{otherwise} \end{cases}$
- $F = \{(p,S) \mid S \cap F_B \neq \emptyset\}$
- $(p,S)\cdot a = \begin{cases} (p\cdot a, S\cdot a) & \text{if } p\cdot a \notin F_A \\ (p\cdot a, S\cdot a \cup \{i_B\}) & \text{otherwise} \end{cases}$

We define the DFA $A \circ B = (\Sigma, Q, i, F, \cdot)$ as follows :

- $Q = \{(p,q) \mid p \in Q_A, q \in Q_B\}$
- $i = (i_A, i_B)$
- $F = \{(p,q) \mid p \in F_A\} \circ \{(p,q) \mid q \in F_B\}$
- $(p,q)\cdot a = (p\cdot a, q\cdot a)$

It is easy to verify the following lemma:

Lemma 1. *$L(A \cdot B) = L(A) \cdot L(B)$ and $L(A \circ B) = L(A) \circ L(B)$.*

Combining these two constructions, we obtain the DFA $A \cdot (B \circ C)$ whose states are of the form (i,T) where $i \in Q_A$ and $T \subset Q_B \times Q_C$.

For any state (i,T), assuming $|Q_B| = n$ and $|Q_C| = p$, the set T can be seen as a tableau with n rows and p columns where any cell (j,k) is marked if and only if the couple of states (j,k) is in T (see Figure 5). In the following, for simplicity, when the dimensions are fixed, we assimilate a tableau with the set of its marked cells.

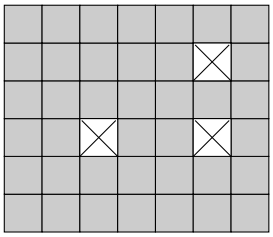


Fig. 1 The tableau corresponding to $T = \{(3,2),(1,5),(3,5)\}$ with $n = 6$ and $p = 7$.

Since the state complexity of catenation is $_\bullet(m,m') = (m-1)2^{m'} + 2^{m'-1}$ and the state complexity of a binary boolean operation $\circ$ is bounded by $_\circ(n,p) = np$ (see [7]), from Claim , their composition allows to bound the state complexity of $A \cdot (B \circ C)$ by $(m-1)2^{np} + 2^{np-1}$. This bound is reached when $\circ = \cap$ [3].

The state complexity for the combination of catenation with union $(A \cdot (B \cup C))$ has been studied in [3] but it can be reinterpreted using the tableaux defined previously. Let (i,T) and (i,T') be two distinct states such that the couples (x,x') and (y,y') are in T' and $T = T' \cup \{(x,y')\}$. Then the two states (i,T) and (i,T') are equivalent. Indeed, to separate these states, one has to find a word w such that (1) $T' \cdot w$ is equal to a set of couples which members are both non-final and (2) $(x,y') \cdot w$ leads to a couple of states at least one of the two is final. The fact that $x \cdot w$ or $y' \cdot w$ is final contradicts (1). So (i,T) and (i,T') are equivalent.

Such equivalent states have tableaux with specific patterns. Indeed, the tableaux for T and T' contain the pattern of Figure 6(a) and Figure 6(b) respectively. None of them can be distinguished from the pattern of Figure 6(c). So the number of equivalent states is the number of indistinguishable tableaux represented by the patterns of Figure 6. The number of tableaux not containing patterns of Figure 6(a) or Figure 6(b) is $(2^n - 1)(2^p - 1) + 1$.

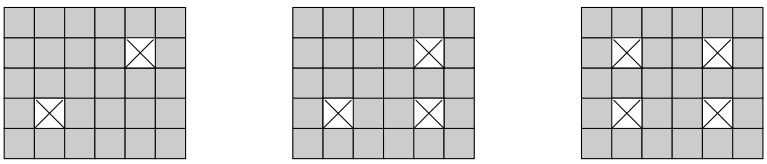


Fig. 2 Three indistinguishable tableaux (a), (b), (c), for the union operator.

Indeed, one has to choose among n rows and p columns (at least one of each) and mark every cell at the intersection of the chosen rows and columns ($(2^n-1)(2^p-1)$) plus one configuration with no cell marked. We also have to count the same tableaux but with the cell $(0,0)$ marked ($2^{n-1}2^{p-1}$ tableaux). Combined with the state complexity of catenation, these observations lead to the state complexity $(m-1)((2^n-1)(2^p-1)+1)+2^{n-1}2^{p-1} = (m-1)(2^{n+p}-2^n-2^p+2)+2^{n+p-2}$ of $A\cdot(B\cup C)$.

As there exist DFAs A, B and C such that there are no indistinguishable tableaux for $A\cdot(B\cap C)$, the state complexity of catenation combined with intersection coincides with the bound.

As for the union, some particular states are necessarily equivalent for $A\cdot(B\oplus C)$. Let (i,T) and (i,T') be two distinct states such that the couples (x,x'), (x,y') and (y,y') are in T' and $T = T'\cup\{(y,x')\}$. Then the two states (i,T) and (i,T') are equivalent. Indeed, if a word w separates (i,T) and (i,T'), then w sends y in F_B or x' in F_C but not both, sending (i,T) to a final state of $A\cdot(B\oplus C)$. This cannot be achieved without sending (i,T') to a final state of $A\cdot(B\oplus C)$, thus contradicting the separation by w.

Such equivalent states imply indistinguishable tableaux as described below. Four distinct marked cells s_1, s_2, s_3 and s_4 define a *rectangle* if there exist four integers x, x', y and y' such that $\{s_1,s_2,s_3,s_4\} = \{x,y\}\times\{x',y'\}$. Three distinct marked cells s_1, s_2 and s_3 form a *right triangle* if there exists an unmarked cell s_4 such that s_1, s_2, s_3 and s_4 form a rectangle (See Figure 3 and Figure 4). A tableau T is *saturated* if it does not contain any right triangle. For each tableau T, we define $\mathrm{Sat}(T)$ as the smallest saturated tableau containing T. Notice that $\mathrm{Sat}(T)$ is the intersection of all saturated tableaux containing T. Its existence is ensured since the tableau with each cell marked is saturated. Its unicity is due to the fact that the intersection of two saturated tableau containing T is still a saturated tableau containing T.

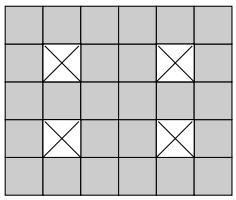


Fig. 3 A rectangle.

Let $\alpha_{n,p}$ be the number of saturated tableaux with n lines and p rows. Furthermore, if i is final in A, then T contains $(0,0)$ and consequently so does $\mathrm{Sat}(T)$. Let $\alpha'_{n,p}$ be the number of such tableaux where the cell $(0,0)$ is marked. The values $\alpha_{n,p}$ and $\alpha'_{n,p}$ are precised in [2] where it is proved that

Theorem 1. $\mathrm{sc}(A\cdot(B\oplus C)) = (m-1)\alpha_{n,p}+\alpha'_{n,p}$

Combining these two constructions, we obtain the DFA $A\cdot(B\circ C)$ whose states are of the form (i,T) where $i\in Q_A$ and $T\subset Q_B\times Q_C$.

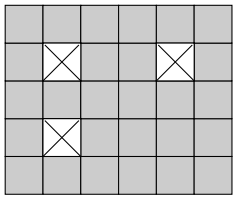


Fig. 4 A right triangle.

For any state (i,T), assuming $|Q_B| = n$ and $|Q_C| = p$, the set T can be seen as a tableau with n rows and p columns where any cell (j,k) is marked if and only if the couple of states (j,k) is in T (see Figure 5). In the following, for simplicity, when the dimensions are fixed, we assimilate a tableau with the set of its marked cells.

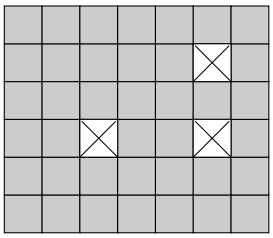


Fig. 5 The tableau corresponding to $T = \{(3,2),(1,5),(3,5)\}$ with $n = 6$ and $p = 7$.

Since the state complexity of catenation is $_\bullet(m,m') = (m-1)2^{m'} + 2^{m'-1}$ and the state complexity of a binary boolean operation $\circ$ is bounded by $_\circ(n,p) = np$ (see [7]), from Claim , their composition allows to bound the state complexity of $A \cdot (B \circ C)$ by $(m-1)2^{np} + 2^{np-1}$. This bound is reached when $\circ = \cap$ [3].

The state complexity for the combination of catenation with union $(A \cdot (B \cup C))$ has been studied in [3] but it can be reinterpreted using the tableaux defined previously. Let (i,T) and (i,T') be two distinct states such that the couples (x,x') and (y,y') are in T' and $T = T' \cup \{(x,y')\}$. Then the two states (i,T) and (i,T') are equivalent. Indeed, to separate these states, one has to find a word w such that (1) $T' \cdot w$ is equal to a set of couples which members are both non-final and (2) $(x,y') \cdot w$ leads to a couple of states at least one of the two is final. The fact that $x \cdot w$ or $y' \cdot w$ is final contradicts (1). So (i,T) and (i,T') are equivalent.

Such equivalent states have tableaux with specific patterns. Indeed, the tableaux for T and T' contain the pattern of Figure 6(a) and Figure 6(b) respectively. None of them can be distinguished from the pattern of Figure 6(c). So the number of equivalent states is the number of indistinguishable tableaux represented by the patterns of Figure 6. The number of tableaux not containing patterns of Figure 6(a) or Figure 6(b) is $(2^n-1)(2^p-1)+1$.

Indeed, one has to choose among n rows and p columns (at least one of each) and mark every cell at the intersection of the chosen rows and columns $((2^n-1)(2^p-$

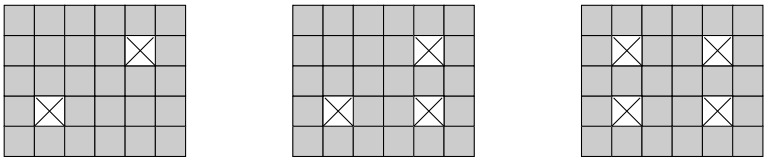


Fig. 6 Three indistinguishable tableaux (a), (b), (c), for the union operator.

1)) plus one configuration with no cell marked. We also have to count the same tableaux but with the cell $(0,0)$ marked ($2^{n-1}2^{p-1}$ tableaux). Combined with the state complexity of catenation, these observations lead to the state complexity $(m-1)((2^n-1)(2^p-1)+1)+2^{n-1}2^{p-1} = (m-1)(2^{n+p}-2^n-2^p+2)+2^{n+p-2}$ of $A\cdot(B\cup C)$.

As there exist DFAs A, B and C such that there are no indistinguishable tableaux for $A\cdot(B\cap C)$, the state complexity of catenation combined with intersection coincides with the bound.

As for the union, some particular states are necessarily equivalent for $A\cdot(B\oplus C)$. Let (i,T) and (i,T') be two distinct states such that the couples (x,x'), (x,y') and (y,y') are in T' and $T = T'\cup\{(y,x')\}$. Then the two states (i,T) and (i,T') are equivalent. Indeed, if a word w separates (i,T) and (i,T'), then w sends y in F_B or x' in F_C but not both, sending (i,T) to a final state of $A\cdot(B\oplus C)$. This cannot be achieved without sending (i,T') to a final state of $A\cdot(B\oplus C)$, thus contradicting the separation by w.

Such equivalent states imply indistinguishable tableaux as described below. Four distinct marked cells s_1, s_2, s_3 and s_4 define a *rectangle* if there exist four integers x, x', y and y' such that $\{s_1,s_2,s_3,s_4\} = \{x,y\}\times\{x',y'\}$. Three distinct marked cells s_1, s_2 and s_3 form a *right triangle* if there exists an unmarked cell s_4 such that s_1, s_2, s_3 and s_4 form a rectangle (See Figure 3 and Figure 4). A tableau T is *saturated* if it does not contain any right triangle. For each tableau T, we define $\mathrm{Sat}(T)$ as the smallest saturated tableau containing T. Notice that $\mathrm{Sat}(T)$ is the intersection of all saturated tableaux containing T. Its existence is ensured since the tableau with each cell marked is saturated. Its unicity is due to the fact that the intersection of two saturated tableau containing T is still a saturated tableau containing T.

Let $\alpha_{n,p}$ be the number of saturated tableaux with n lines and p rows. Furthermore, if i is final in A, then T contains $(0,0)$ and consequently so does $\mathrm{Sat}(T)$. Let $\alpha'_{n,p}$ be the number of such tableaux where the cell $(0,0)$ is marked. The values $\alpha_{n,p}$ and $\alpha'_{n,p}$ are precised in [2] where it is proved that

Theorem 2. $\mathrm{sc}(A\cdot(B\oplus C)) = (m-1)\alpha_{n,p}+\alpha'_{n,p}$

4 Accessibility

We propose as a common family of witnesses for $A \cdot (B \cup C)$ and $A \cdot (B \cap C)$ the family of triples $W_{m,n,p}$ presented in Figure 9.

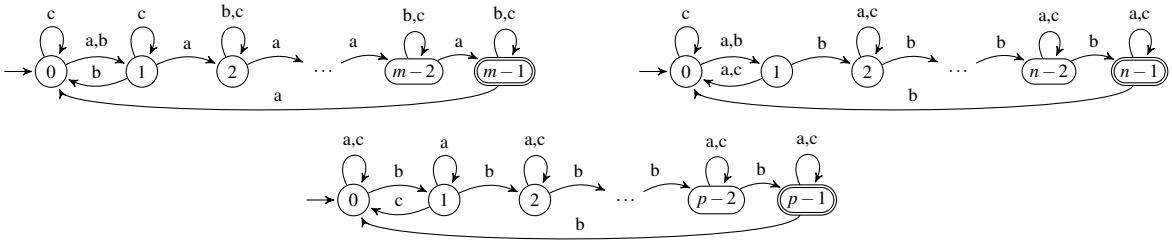


Fig. 9 3-letters witness for $A \cdot (B \cup C)$ and $A \cdot (B \cap C)$

According to the constructions described in Section , we define, for each $m,n,p \geq 3$ the automaton $A \cdot (B \circ C)$ whose accessible states are indexed by pairs (i,T) where $0 \leq i \leq m-1$ and $T \subset [0,n-1] \times [0,p-1]$. The transitions are described as follows: for each pair (i,T) and each symbol σ,

$$(i,T) \cdot \sigma = \begin{cases} (i \cdot \sigma, \{(j \cdot \sigma, k \cdot \sigma) : (j,k) \in T\}) & \text{if } i \cdot \sigma \neq m-1 \\ (i \cdot \sigma, \{(j \cdot \sigma, k \cdot \sigma) : (j,k) \in T\} \cup \{(0,0)\}) & \text{if } i \cdot \sigma = m-1 \end{cases}$$

It is easy to see that only the states (i,T) satisfying $i = m-1 \Rightarrow (0,0) \in T$ are accessible. We set

$$\begin{aligned} Acc_{m,n,p}^{A \cdot (B \circ C)} = & \{(i,T) : 0 \leq i < m-1, T \subset Q_n \times Q_p\} \\ & \cup \{(m-1, \{(0,0)\} \cup T) : T \subset Q_n \times Q_p\} \end{aligned}$$

Notice that the set $Acc_{m,n,p}^{A \cdot (B \circ C)}$ does not depend on $\circ$.

Lemma 2. *For any state $s = (i, \{(j,k)\})$ with $i \neq m-1$, there exists a word $w \in \{a,b\}^*$ such that $(m-1, \{(0,0)\}) \cdot w = s$.*

Proof. We have to consider three cases:

1. If $j > 2$ then we have

$$\begin{aligned} (m-1, \{(0,0)\}) \xrightarrow{a} (0, \{(1,0)\}) & \xrightarrow{(ab)^{k-j+1}p} (0, \{(1, k-j+1)\} \\ & \xrightarrow{b^{j-2}} (j2, \{(j-1, k-1)\}). \end{aligned}$$

   Since $j-1 > 1$, one obtains

$$\begin{aligned} (j2, \{(j-1,k-1)\}) & \xrightarrow{a^{[j+1]}2} (1, \{(j-1,k-1)\}) \\ & \xrightarrow{b} (0, \{(j,k)\} \xrightarrow{a^i} (i, \{(j,k)\}). \end{aligned}$$

   In conclusion, the word

$$w = a(ab)^{k-j+1p}b^{j-2}a^{j+12}ba^i \in \{a,b\}^*$$

is such that $(m-1,\{(0,0)\}).w = (i,\{(j,k)\})$. This proves the lemma.

2. If $j = 2$ then we have

$$(m-1,\{(0,0)\}) \xrightarrow{a} (0,\{(1,0)\}) \xrightarrow{(ab)^{k-1p}} (0,\{(1,k-1)\} \xrightarrow{b^{np-2}} (np2,\{(n-1,k-3)\}).$$

Hence,

$$(np2,\{(n-1,k-3)\}) \xrightarrow{a^{np+12}} (1,\{(n-1,k-3)\}) \xrightarrow{b^3} (0,\{(2,k)\} \xrightarrow{a^i} (i,\{(2,k)\}).$$

In conclusion, the word

$$w = a(ab)^{k-1p}b^{np-2}a^{np+12}b^3a^i \in \{a,b\}^*$$

is such that $(m-1,\{(0,0)\}) \cdot w = (i,\{(j,k)\})$. This proves the lemma.

3. If $j < 2$ then set $\gamma_i = i$ if $i > 1$ and $\gamma_i = i+j+12$ if $i \leq 1$. One has

$$(\gamma_i,\{(n-1,k-j-1)\}) \xrightarrow{b^{j+1}} (i,\{(j,k)\}).$$

Hence, as $\gamma_i \neq m-1$, there exists a word v such that $(m-1,\{(0,0)\}) \xrightarrow{v} (\gamma_i, n-1, k-j-1)$ due to one of the previous cases. So one obtains $(m-1,\{(0,0)\}) \cdot vb^{j+1} = (i,\{(j,k)\})$ as expected.

Proposition 1. *For any boolean operation $\circ$, all the states of $Acc_{m,n,p}^{A\cdot(B\circ C)}$ are accessible.*

Proof. We prove by induction on $|T|$ that each state (i,T) is accessible. First, observe that all the states $(i,\emptyset)$ are reachable from $(0,\emptyset)$ reading a^i. Now consider a state (i,T) with $T \neq \emptyset$.

1. Suppose that $i = m-1$ then the states $(m-1,T)$ is reachable by a from $(m-2, a\cdot(T\setminus\{(0,0)\})$ which is accessible by induction.
2. Suppose now $i < m-1$ and let $(j,k) \in T$. Since $i \neq m-1$, by Lemma 3, there exists a word $w \in \{a,b\}^*$ such that $(m-1,\{(0,0)\}) \cdot w = (i,\{(j,k)\})$. Observe that, from the definition of the automata, the letters a and b encode permutations of the states (no contraction is involved). It follows that $w \cdot T$ has the same number of elements as T and so the state (i,T) is accessible by w from $(m-1, w\cdot T)$ which is accessible from (1).

According to the constructions described in Section , we define, for each $m,n,p \geq 3$ the automaton $A\cdot(B\circ C)$ whose accessible states are indexed by pairs (i,T) where $0 \leq i \leq m-1$ and $T \subset [0,n-1]\times[0,p-1]$. The transitions are described as follows: for each pair (i,T) and each symbol σ,

$$(i,T)\cdot\sigma = \begin{cases} (i\cdot\sigma,\{(j\cdot\sigma,k\cdot\sigma) : (j,k) \in T\}) & \text{if } i\cdot\sigma \neq m-1 \\ (i\cdot\sigma,\{(j\cdot\sigma,k\cdot\sigma) : (j,k) \in T\}\cup\{(0,0)\}) & \text{if } i\cdot\sigma = m-1 \end{cases}$$

It is easy to see that only the states (i,T) satisfying $i = m-1 \Rightarrow (0,0) \in T$ are accessible. We set

$$\begin{aligned} Acc_{m,n,p}^{A\cdot(B\circ C)} = &\{(i,T) : 0 \leq i < m-1, T \subset Q_n \times Q_p\} \\ &\cup \{(m-1, \{(0,0)\} \cup T) : T \subset Q_n \times Q_p\} \end{aligned}$$

Notice that the set $Acc_{m,n,p}^{A\cdot(B\circ C)}$ does not depend on $\circ$.

Lemma 3. *For any state $s = (i, \{(j,k)\})$ with $i \neq m-1$, there exists a word $w \in \{a,b\}^*$ such that $(m-1, \{(0,0)\}) \cdot w = s$.*

Proof. We have to consider three cases:

1. If $j > 2$ then we have
$$\begin{aligned} (m-1, \{(0,0)\}) \xrightarrow{a} (0, \{(1,0)\}) &\xrightarrow{(ab)^{k-j+1p}} (0, \{(1, k-j+1)\} \\ &\xrightarrow{b^{j-2}} (j2, \{(j-1, k-1)\}). \end{aligned}$$
Since $j-1 > 1$, one obtains
$$\begin{aligned} (j2, \{(j-1,k-1)\}) &\xrightarrow{a^{[j+1]_2}} (1, \{(j-1,k-1)\}) \\ &\xrightarrow{b} (0, \{(j,k)\} \xrightarrow{a^i} (i, \{(j,k)\}). \end{aligned}$$
In conclusion, the word
$$w = a(ab)^{k-j+1p} b^{j-2} a^{j+12} b a^i \in \{a,b\}^*$$
is such that $(m-1, \{(0,0)\}).w = (i, \{(j,k)\})$. This proves the lemma.
2. If $j = 2$ then we have
$$\begin{aligned} (m-1, \{(0,0)\}) \xrightarrow{a} (0, \{(1,0)\}) &\xrightarrow{(ab)^{k-1p}} (0, \{(1, k-1)\} \\ &\xrightarrow{b^{np-2}} (np2, \{(n-1, k-3)\}). \end{aligned}$$
Hence,
$$\begin{aligned} (np2, \{(n-1,k-3)\}) &\xrightarrow{a^{np+12}} (1, \{(n-1,k-3)\}) \\ &\xrightarrow{b^3} (0, \{(2,k)\} \xrightarrow{a^i} (i, \{(2,k)\}). \end{aligned}$$
In conclusion, the word
$$w = a(ab)^{k-1p} b^{np-2} a^{np+12} b^3 a^i \in \{a,b\}^*$$
is such that $(m-1, \{(0,0)\}) \cdot w = (i, \{(j,k)\})$. This proves the lemma.
3. If $j < 2$ then set $\gamma_i = i$ if $i > 1$ and $\gamma_i = i + j + 12$ if $i \leq 1$. One has
$$(\gamma_i, \{(n-1, k-j-1)\}) \xrightarrow{b^{j+1}} (i, \{(j,k)\}).$$
Hence, as $\gamma_i \neq m-1$, there exists a word v such that $(m-1, \{(0,0)\}) \xrightarrow{v} (\gamma_i, n-1, k-j-1)$ due to one of the previous cases. So one obtains $(m-1, \{(0,0)\}) \cdot vb^{j+1} = (i, \{(j,k)\})$ as expected.

Proposition 2. *For any boolean operation $\circ$, all the states of $Acc_{m,n,p}^{A\cdot(B\circ C)}$ are accessible.*

Proof. We prove by induction on $|T|$ that each state (i,T) is accessible. First, observe that all the states $(i,\emptyset)$ are reachable from $(0,\emptyset)$ reading a^i. Now consider a state (i,T) with $T \neq \emptyset$.

1. Suppose that $i = m-1$ then the states $(m-1,T)$ is reachable by a from $(m-2, a\cdot(T\setminus\{(0,0)\})$ which is accessible by induction.
2. Suppose now $i < m-1$ and let $(j,k) \in T$. Since $i \neq m-1$, by Lemma 3, there exists a word $w \in \{a,b\}^*$ such that $(m-1,\{(0,0)\})\cdot w = (i,\{(j,k)\})$. Observe that, from the definition of the automata, the letters a and b encode permutations of the states (no contraction is involved). It follows that $w\cdot T$ has the same number of elements as T and so the state (i,T) is accessible by w from $(m-1, w\cdot T)$ which is accessible from (1).

5 Pairwise non-equivalence

In this section, we restrict the $\circ$ notation to $\cup$ and $\cap$ only. Notice that the accessibility proved in the previous section is valid for any boolean operation.

Let us first notice that the action of the letter c is not used to prove the accessibility of the states. Nevertheless, this letter is needed to separate the states. The following lemma highlights a property of the action of c which is central in the study of the separability. Its proof is straightforward from the definition of $A\cdot(B\circ C)$.

Lemma 4. *Let (i,T) be a state in $\mathrm{Acc}_{m,n,p}^{A\cdot(B\circ C)}$. We have $(i,T)\cdot c = (i,T')$ with*

$$T' \subset (Q_n \setminus \{1\}) \times (Q_p \setminus \{1\})$$

Let us consider first the case where $\circ = \cap$. Notice that a state (i,T) of $\mathrm{Acc}_{m,n,p}^{A\cdot(B\cap C)}$ is final if and only if $(n-1,p-1) \in T$.

Proposition 3. *The states belonging to $\mathrm{Acc}_{m,n,p}^{A\cdot(B\cap C)}$ are pairwise nonequivalent.*

Proof. Let $s = (i,T)$ and $s' = (i',T')$ be two distinct states. Without loss of generality we assume $i' \leq i$ (otherwise we permute the role of the states) and we construct a word $w_{s,s'}$ sending one of the state on a final state and the other on a non final state. We consider several cases as follows

1. Suppose $i' < i < m-1$. We have

   $$s \xrightarrow{a^{m-i-2}c} (m-2,T_2) \text{ and } s' \xrightarrow{a^{m-i-2}c} (m-2+i'-i,T_2')$$

   with, from Lemma 4, $(1,0) \notin T_2 \cup T_2'$.
   Hence,

   $$(m-2,T_2) \xrightarrow{a} (m-1,T_3) \xrightarrow{b^{np-1}} (m-1,T_4)$$

   with $(0,0) \in T_3$ and $(n-1,p-1) \in T_4$, and

$$(m-2+i'-i,T_2') \xrightarrow{a} (m-1+i'-i,T_3') \xrightarrow{b^{np-1}} (i_4',T_4')$$

with $(0,0) \notin T_3'$ because both $m-1+i'-i \neq m-1$ and $(1,0) \notin T_2'$. Furthermore as $m-1$ is never reached from $m-1+i'-i$ reading b^{np-1}, we have $(n-1,p-1) \notin T_4'$. Setting $w_{s,s'} = a^{m-i-2}cab^{np-1}$, we have $s \cdot w_{s,s'} = (m-1,T_4)$ which is final and $s' \cdot w_{s,s'} = (i_4',T_4')$ which is not final. So, s and s' are not equivalent.

2. If $i' < i = m-1$ then reading a sends s to a state $s_1 = (0,T_1)$ and s' to a state $s_1' = (i'+1,T_1')$. If $i'+1 \neq m-1$,then we set $w_{s,s'} = aw_{s_1',s_1}$ where w_{s_1',s_1} is computed from the previous case. If $i'+1 = m-1$ then we read another a and this sends s_1 to a state $s_2 = (1,T_2)$ and s_1' to a state $s_2' = (0,T_2')$. As A has at least 3 states, $m-1 \neq 1$. So $w_{s,s'} = a^2 w_{s_2,s_2'}$ where $w_{s_2,s_2'}$ is the word computed in the previous case.
3. If $i = i'$ then $T \neq T'$. Without loss of generality we assume that there exists $(j,k) \in T \setminus T'$. Let us recall the Kronecker delta $\delta_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$ We have :

$$s \xrightarrow{b^{\delta_{0,k}}} (i_1,T_1) \text{ and } s' \xrightarrow{b^{\delta_{0,k}}} (i_1,T_1')$$

with $i_1 \in \{0,1,i\}$ and $(j_1,k_1) \in T_1 \setminus T_1'$ where $j_1 = j + \delta_{0,k}$ and $k_1 = k + \delta_{0,k}$. Let us notice that $k_1 \neq 0$. Then, we have

$$(i_1,T_1) \xrightarrow{a^{m-i_1+1}} (1,T_2) \xrightarrow{(ba)^{n-j_2}} (1,T_3) \xrightarrow{(ba)^{-n-k_3p}} (1,T_4) \xrightarrow{b^{n-1}} (\varepsilon,T_5)$$

and

$$(i_1,T_1')' \xrightarrow{a^{m-i_1+1}} (1,T_2') \xrightarrow{(ba)^{n-j_2}} (1,T_3') \xrightarrow{(ba)^{-n-k_3p}} (1,T_4') \xrightarrow{b^{n-1}} (\varepsilon,T_5')$$

with $(j_2,k_2) = (j_1,k_1) \cdot a^{m-i_1+1} \in T_2 \setminus T_2'$, $(0,k_3) = (j_2,k_2) \cdot (ba)^{n-j_2} \in T_3 \setminus T_3'$, $(0,p-n) = (0,k_3) \cdot (ba)^{-n-k_3p} \in T_4 \setminus T_4'$, $(n-1,p-1) = (0,p-n) \cdot b^{n-1} \in T_5 \setminus T_5'$, and $\varepsilon \in \{0,1\}$. So, the state (ε,T_5) is final while (ε,T_5') is not.
Setting $w_{s,s'} = b^{\delta_{0,k}} a^{m-i_1+1} (ba)^{n-j_2+-n-k_3p} b^{n-1}$ we obtain that $s.w_{s,s'}$ is final while $s'.w_{s,s'}$ is not final. In other words, s and s' are nonequivalent.

Now, we consider the case where $\circ = \cup$. The final states of $A \cdot (B \cup C)$ are the pairs (i,T) such that

$$T \cap (\{n-1\} \times Q_p \cup Q_n \times \{p-1\}) \neq \emptyset.$$

We say that a set T is *saturated* if $(j,k),(j',k') \in T$ implies $(j,k') \in T$.

$$(T) = \{j : (j,k) \in T\} \times \{k : (j,k) \in T\}. \tag{1}$$

Lemma 5. *In* $\mathrm{Acc}_{m,n,p}^{A \cdot (B \cup C)}$*, any state* (i,T) *is equivalent to* $(i,(T))$.

Proof. Suppose that there exists a word w such that $(i,(T)) \cdot w$ is final and $(i,T) \cdot w$ is not final. Then, there are two couples (j,k) and (j',k') in T with $(j,k') \in (T) \setminus T$

and

$$(j,k')\cdot w \in (\{n-1\}\times Q_p)\cup(Q_n\times\{p-1\}).$$

This means that either $j\cdot w = n-1$ or $k'\cdot w = p-1$. But since $(j,k)\cdot w, (j',k')\cdot w \in T\cdot w$ we have

$$T\cdot w \cap (\{n-1\}\times Q_p \cup Q_n\times\{p-1\}) \neq \emptyset$$

and this is not possible because $(i,T)\cdot w = (i\cdot w, T\cdot w)$ which is not final.

Corollary 1. *Let (i,T) and (i,T') be two states of $\mathrm{Acc}_{m,n,p}^{A\cdot(B\cup C)}$ such that $(T)=(T')$. Then (i,T) and (i,T') are equivalent.*

From now, we will only consider the set of saturated states defined as follows :

$$Sat = \{(i,(T)) \in \mathrm{Acc}_{m,n,p}^{A\cdot(B\cup C)}\}$$

We define $\mathrm{split}(T) = (\{j : (j,k)\in T\}, \{k : (j,k)\in T\})$. For any $s=(i,T)\in Sat$ we define $L(s)=S_1$ and $R(s)=S_2$ if $\mathrm{split}(T)=(S_1,S_2)$. With this notation, a state s is final if and only if $n-1\in L(s)$ or $p-1\in R(s)$. Notice that Lemma 4 can be restated as follows.

Lemma 6. *Let $s\in \mathrm{Acc}_{m,n,p}^{A\cdot(B\cup C)}$. Then $s\cdot c = s'$ with $1\notin L(s')$ and $1\notin R(s')$.*

Now we have defined all the material we need to prove the pairwise non-equivalence of the states of *Sat*.

Proposition 4. *The states belonging to Sat are pairwise non-equivalent.*

Proof. Let $s=(i,T)$ and $s'=(i',T')$ be two distinct states of *Sat* . Without loss of generality we assume $i'\leq i$. First suppose $i'<i$. We have to consider the following cases.

- If $i'<i<m-1$ then we set

$$s \xrightarrow{a^{m-i+1}(ba)^{i-i'-1}} s_1=(1,T_1) \xrightarrow{a^{m-3}c} s_2=(m-2,T_2)$$

$$s' \xrightarrow{a^{m-i+1}(ba)^{i-i'-1}} s'_1=(0,T'_1) \xrightarrow{a^{m-3}c} s'_2=(m-3,T'_2)$$

Using Lemma 6 we observe that $1\notin L(s'_2)$ and $1\notin R(s'_2)$. Setting

$$s_2 \xrightarrow{a} s_3=(m-1,T_3) \xrightarrow{aba} s_4=(2,T_4) \xrightarrow{b^{np-3}} s_5=(2,T_5),$$

We observe that $0\in L(s_3)$, $2\in L(s_4)$, and then $n-1\in L(s_5)$. In other words s_5 is a final state. In the other hand, we set

$$s'_2 \xrightarrow{a} s'_3=(m-2,T'_3) \xrightarrow{aba} s'_4=(0,T'_4) \xrightarrow{b^{np-3}} s'_5=(\varepsilon,T'_5),$$

with $\varepsilon\in\{0,1\}$. We observe that $0\notin L(s'_3)$ and $1\notin R(s'_3)$, $2\notin L(s'_4)$ and $2\notin R(s'_4)$, and finally $n-1\notin L(s'_5)$ and $p-1\notin R(s'_5)$. In other words s'_5 is not a final state.

Setting $w_{s,s'} = a^{m-i+1}(ba)^{i-i'-1}a^{m-3}caabab^{np-3}$, we obtain that $s \cdot w_{s,s'}$ is final but not $s' \cdot w_{s,s'}$. This proves that s and s' are not equivalent.

- If $i' < i = m-1$ then, by reading a or aa, we recover the case where $i' < i < m-1$.

Suppose now that $i = i'$ and $L(s) \neq L(s')$. Without loss of generality we consider $j \in L(s) \setminus L(s')$. We have two cases to consider:

- If $j > 1$ then we set

$$s \xrightarrow{a^{m-i}} s_1 = (0, T_1) \xrightarrow{(ab)^{n-1-j}} s_2 = (0, T_2)$$

and

$$s' \xrightarrow{a^{m-i}} s'_1 = (0, T'_1) \xrightarrow{(ab)^{n-1-j}} s'_2 = (0, T'_2)$$

We observe that $j \in L(s_1) \setminus L(s'_1)$ and $n-1 \in L(s_2) \setminus L(s'_2)$. We set

$$s_2 \xrightarrow{cb} s_3 = (1, T_3) \xrightarrow{(ba)^{-n-2p}} s_4 = (1, T_4) \xrightarrow{b^{n-1}} s_5 = (\varepsilon, T_5)$$

and

$$s'_2 \xrightarrow{cb} s'_3 = (1, T'_3) \xrightarrow{(ba)^{-n-2p}} s'_4 = (1, T'_4) \xrightarrow{b^{n-1}} s'_5 = (\varepsilon, T'_5)$$

with $\varepsilon \in \{0,1\}$. We have $0 \in L(s_3) \setminus L(s'_3)$ and $2 \notin R(s'_3)$. Hence, $0 \in L(s_4) \setminus L(s'_4)$ and $p-n \notin R(s'_4)$. Finally, $n-1 \in L(s_5) \setminus L(s'_5)$ and $p-1 \notin R(s'_5)$. In conclusion, if we set $w_{s,s'} = a^{m-i}(ab)^{n-1-j}cb(ba)^{-n-2p}b^{n-1}$ then the state $s_5 = s \cdot w_{s,s'}$ is final while $s'_5 = s' \cdot w_{s,s'}$ is not. Consequently, s and s' are not equivalent.
- If $j \leq 1$ then we act by b or b^2 in the aim to send s and s' respectively to $s_1 = (i_1, T_1)$ and $s'_1 = (i_1, T'_1)$ with $2 \in L(s_1) \setminus L(s'_1)$. So we find the result by applying the previous point.

Now we suppose $i = i'$ and $R(s) \neq R(s')$. Without loss of generality we assume that there exists $k \in R(s) \setminus R(s')$. We have to consider two cases:

- If $k > 1$ then we set

$$s \xrightarrow{a^{m-i}} s_1 = (0, T_1) \xrightarrow{c} s_2 = (0, T_2)$$

and

$$s' \xrightarrow{a^{m-i}} s'_1 = (0, T'_1) \xrightarrow{c} s'_2 = (0, T'_2)$$

We observe that $k \in R(s_1) \setminus R(s'_1)$, $1 \notin L(s'_2)$ and $k \in R(s_2) \setminus R(s'_2)$. We set

$$s_2 \xrightarrow{(ab)^{-n-k+1p}} s_3 = (0, T_3) \xrightarrow{(b)^{n-2}} s_4 = (\varepsilon, T_4)$$

and

$$s'_2 \xrightarrow{(ab)^{-n-k+1p}} s'_3 = (0, T'_3) \xrightarrow{(b)^{n-2}} s'_4 = (\varepsilon, T'_4)$$

with $\varepsilon \in \{0,1\}$. We have $1 \notin L(s'_3)$ and $p-n+1 \in R(s_3) \setminus R(s'_3)$. Finally, $n-1 \notin L(s'_4)$ and $p-1 \in R(s_4) \setminus R(s'_4)$. In conclusion, if we set $w_{s,s'} = a^{m-i}c(ab)^{-n-k+1p}b^{n-2}$

then the state $s_4 = s \cdot w_{s,s'}$ is final while $s'_4 = s' \cdot w_{s,s'}$ is not. Consequently, s and s' are not equivalent.

- If $k \leq 1$ then we act by b or b^2 in the aim to send s and s' respectively to $s_1 = (i_1, T_1)$ and $s'_1 = (i_1, T'_1)$ with $2 \in R(s_1) \setminus R(s'_1)$. So we find the result by applying the previous point.

The following theorem summarizes our main results.

Theorem 3. *For $A \cdot (B \cap C)$ as well as $A \cdot (B \cup C)$, the bound given by the state complexity of these two combinations is reached by the 3-letters witnesses family $W_{m,n,p}$.*

1 The symmetric difference case

Unfortunatly, the family $W_{m,n,p}$ fails for the combination of catenation with boolean xor operator. We prove it by studying the case $m = n = 3$ and $p = 4$ using tableaux described in Section .

A final state of the catenation combined with the xor has at least one marked cell on the last line or row but not both.

Let us show that the two final states represented by $t = (i,$  $)$ and $t' = (j,$  $)$ are not distinguishable. Indeed, Figure 10 denotes all accessible configurations starting from the tableaux of t and t'. Every couple of tableaux represent a couple of final states. In this figure, we suppose that $j \cdot w$ is not $m-1$. If we have $j \cdot w = m-1$, we have two cases to consider:

1. the cell $(0,0)$ is marked in t'. As accessing $m-1$ creates this state, both tableaux are unchanged;
2. the cell $(0,0)$ is not marked in t'. In this case, we have to notice that marking this state and saturating the obtained tableau gives the full tableau for t' and so the states are undistinguishable.

Conclusion

In this paper, we have improved witnesses for the state complexity of catenation combined with the union and the intersection. We give a common 3-letters witness for both combinations resolving two conjectures of Brzozowski. Moreover, these witnesses are Brzozowski automata. We also show, using combinatorial tools, why these witnesses fail for the combination of the catenation and the symmetric difference. Furthermore, after numerous unsuccessful attempts to find a 3-letters witness

for this combination with the Sage software, we conjecture that 4 letters are needed to obtain a witness. Such a 4-letters witness is provided in [2].

References

1. Janusz A. Brzozowski. In search of most complex regular languages. *Int. J. Found. Comput. Sci.*, 24(6):691–708, 2013.
2. Pascal Caron, Jean-Gabriel Luque, Ludovic Mignot, and Bruno Patrou. State complexity of catenation combined with a boolean operation: A unified approach. *Int. J. Found. Comput. Sci.*, 27(6):675–704, 2016.
3. Bo Cui, Yuan Gao, Lila Kari, and Sheng Yu. State complexity of two combined operations: Catenation-union and catenation-intersection. *Int. J. Found. Comput. Sci.*, 22(8):1797–1812, 2011.
4. O Ganyushkin and Volodymyr Mazorchuk. *Classical finite transformation semigroups: an introduction*. Algebra and Applications. Springer, Dordrecht, 2008.
5. Yuan Gao and Sheng Yu. State complexity approximation. In Jürgen Dassow, Giovanni Pighizzini, and Bianca Truthe, editors, *Proceedings Eleventh International Workshop on Descriptional Complexity of Formal Systems, DCFS 2009, Magdeburg, Germany, July 6-9, 2009.*, volume 3 of *EPTCS*, pages 121–130, 2009.
6. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.
7. Sheng Yu, Qingyu Zhuang, and Kai Salomaa. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125(2):315–328, 1994.

An improvement of the determinization of fuzzy finite automata via factorization of fuzzy states

S. Stanimirović[1], M. Ćirić[2], J. Ignjatović[3]

University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Srebia
[1] *stanimirovic.stefan@gmail.com*
[2] *miroslav.ciric@pmf.edu.rs*
[3] *jelena.ignjatovic@pmf.edu.rs*

Unlike nondeterministic finite automata (NFAs, for short), which can always be determinized, not all fuzzy finite automata (FFAs, for short) can be determinized. Therefore, the determinization problem is of special interest for FFAs, and it was well elaborated in the recent literature.

De Mendívil and Garitagoitia have developed in [3] the determinization method that produces the *complete* and *deterministic* FFA for automata which accept fuzzy languages of infinite range. They were motivated by the fact that the common procedure of conversion into an equivalent crisp-deterministic fuzzy finite automaton may fail for such fuzzy automata. Their method is based on the idea of *factorizations* of fuzzy sets, and therefore it is called the *determinization via factorization of fuzzy states*. The idea of factorizations was firstly introduced by Kirsten and Maürer [2] in the determinization algorithm for weighted finite automata (WFAs, for short) over semirings, in order to give a generalization of the well-known Mohri's determinization algorithm [4] for WFAs over tropical semirings.

In this paper we provide an improvement of the determinization via factorization of fuzzy states. Our method is based on the usage of the fuzzy relational calculus, namely, on the usage of the right invariant fuzzy quasi-orders. Similar approach has been taken in [1], where it has been shown that such approach combines the determinization and the state reduction methods into two-in-one algorithms that simultaneously perform the determinization and the state reduction. The aim of this paper is to show that such approach can also be employed in the determinization via factorization of fuzzy states.

The results of the paper are the following: For any fuzzy finite automaton $\mathscr{A}$ with the set of states A, a factorization D on A and a fuzzy relation φ on A, we construct the complete deterministic fuzzy automaton $\mathscr{A}_\varphi^D$ of $\mathscr{A}$. We show that $\mathscr{A}_\varphi^D$ and $\mathscr{A}$ recognize the same fuzzy language when φ is a reflexive weakly right invariant fuzzy relation on $\mathscr{A}$. We prove that, if φ and ϕ are right invariant fuzzy quasi-orders on A that satisfy $\varphi \leq \phi$, then $|\mathscr{A}_\phi^D| \leq |\mathscr{A}_\varphi^D|$. In other words, larger right invariant fuzzy quasi-orders determine smaller complete deterministic fuzzy automata. We provide an example where our algorithm results in a finite complete deterministic fuzzy automaton, while the method based on factorization of fuzzy states results in an infinite one.

References

1. Z. Jančić, I. Micić, J. Ignjatović, M. Ćirić, *Further improvement of determinization methods for fuzzy finite automata*, Fuzzy Sets and Systems **301**, pp. 79–102 (2016).
2. D. Kirsten, L. Mäurer, *On the determinization of weighted automata*, J. Automata, Lang. Combin. **10**, pp. 287–312 (2005).
3. J. R. G. de Mendívil, J. R. Garitagoitia, *Determinization of fuzzy automata via factorization of fuzzy states*, Inform. Sci. **283**, pp. 165–179 (2014).
4. M. Mohri, *Finite-state transducers in language and speech processing*, Comput. Linguist. **23** (2), pp. 269–311 (1997).

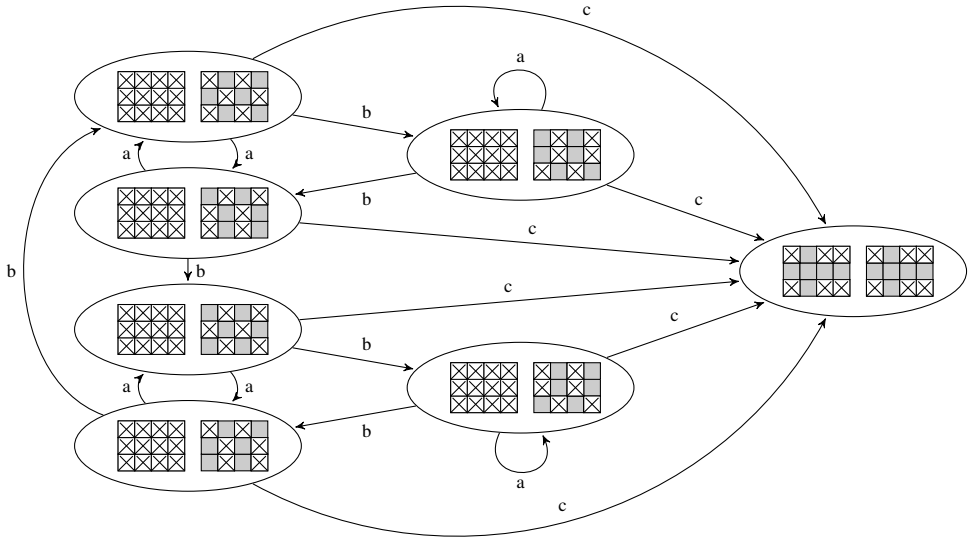


Fig. 10 Two undistinguishable tableaux

Track 2: Cryptography Coding Theory

Chairs: Stephane Ballet (France), Dimitrios Poulakis (Greece), Robert Rolland (France)

Invited Speaker: Claude Carlet

Boolean Functions With Constrained Inputs and the Cryptosystem FLIP

Claude Carlet[1]

[1] *LAGA (Universités Paris 8 et Paris 13, CNRS)*

Abstract

After a recall on the principle of Fully Homomorphic Encryption (FHE) and its use in the Cloud, we shall recall the FLIP encryption model (EUROCRYPT 2016) that minimizes the growth of the noise inherent in the FHE. In FLIP stream ciphering, a Boolean function (of a large number, at least 500, of variables, but very simple) filters a register containing the bits of the secret key swapped at each clock cycle by a pseudo-random permutation. The input to the filtering function has constant Hamming weight since it is the permuted key, and the resistance of FLIP to conventional attacks (the one that exploits the bias of the weight, that of Berlekamp-Massey, the fast correlation attack, and algebraic attacks) need to be re-evaluated in this new context (which did not make the article introducing FLIP). This opens a new research field for Boolean functions. We shall study the cryptographic criteria of balance, algebraic degree, nonlinearity and algebraic immunity under this type of constraint.

A topological approach to network coding

Cristina Martínez and Alberto Besana

[1] *University of Maynooth, Ireland, {cristina.martinezramirez*@nuim.ie}

Consider a network modelled as a digraph $G = (V, E)$ with V the set of vertices $V = \{1, \ldots, n\}$ and E the set of edges $m = |E|$. An automorphism is a permutation of the vertices of the network graph. The network automorphism group decomposition relates automorphism group structure to network topology. There is a natural representation given by a vector space V and a map of groups $\rho : S_n \to GL(V)$, where S_n is the group of permutations of n elements. Codes over finite fields are very much related with the study of the representation theory of the symmetric group over finite fields. Let F be a field. Given integers n, k and d with $1 \leq k \leq n$, an $[n,k,d]_F-$code is a subspace C of F^n of dimension k, such that every non-zero codeword $\alpha \in C$ satisfies $wt(\alpha) \leq d$, where the weight of $\alpha = (a_0, \ldots, a_{n-1}) \in F$ is the number of nonzero components a_i. We study cyclic codes which include generalized Reed-Solomon codes as subspaces subcodes [1]. Namely, consider the isomorphism $l_n : F^n \mapsto F[x]/(x^n - 1)$ between F^n and the ring $R = F[x]/(x^n - 1)$. A subspace C is a cyclic code over F iff C is an ideal of R. The variety of $[n,k,d]_F-$codes over F is parametrized by a Grassmannian $\mathcal{G}_{n,k}(F)$ of $k-$dimensional subspaces in the $F-$vector space F^n, and the set of RS codes constitute a closed set in the Zariski topology. We construct designs with prescribed groups where the blocks are the orbits by the action of the general linear group $GL(n+1, F)$ on the Grassmannian $\mathcal{G}_{n,k}(F)$. Moreover, we study the relation between t-designs of given parameters and codes of constant weight, (see [2], [3]).

References

1. Maria Bras-Amorós and Michael E. O'Sullivan, *From the Euclidean Algorithm for Solving a Key Equation for Dual Reed-Solomon Codes to the Berlekamp-Massey Algorithm*, AAECC (2009), LNCS 5527, pp 32-42, 2009.
2. B. D. Mac Arthur, R. J. Sánchez García, James N. Anderson, *Symmetry in complex networks*, Discrete Applied Mathematics 156 (2008), 525-531.
3. G. Ge, H. Wei, *Group divisible designs with block sizes from $K_{1(3)}$ and Kirkman frames of type $h^u m^1$*, Discrete Mathematics 329 (2014), 42-68.
4. M. Hattori, R. J. Mc Eliece, G. Solomon, *Subspaces subcodes of Reed-Solomon Codes* IEEE Transactions on Information Theory, Vol. 44, No. 5., 1998.

Pairing-Friendly Elliptic Curves Resistant to TNFS Attacks

G. Fotiadis[1], E. Konstantinou[1]

[1] *University of the Aegean, Karlovassi, Samos, Greece, {gfotiadis,ekonstantinou}@aegean.gr*

Abstract

The recent variants of the tower number field sieve method (TNFS) [4, 5] reduce the complexity of the discrete logarithm problem (DLP) in finite extensions of composite degree and this has a major impact on the selection of elliptic curve parameters for pairing-based applications. In this paper we update the criteria for selecting these parameters in order to: (1) surpass the TNFS attacks in finite extensions of composite embedding degree and (2) meet today's security requirements for both prime and composite embedding degrees.

For a prime q let E/F_q be an ordinary elliptic curve with Frobenius trace t. An *asymmetric pairing* is a bilinear, non-degenerate, efficiently computable map $e : G_1 \times G_2 \to G_T$, where $G_1, G_2 \subset E(F_q)$ and $G_T \subset F^*_{q^k}$. The order of all three groups satisfies $\#G_1 = \#G_2 = \#G_T = r$, for some large prime r and k is the *embedding degree*. An elliptic curve is suitable for pairing-based applications, if: (1) its order is $\#E(F_q) = hr$, for some integer $h > 0$, (2) the ρ-value: $\rho = \log q / \log r$ is close to 1, (3) r is large enough, so that the DLP in G_1, G_2 is hard, (4) k is large enough, so that the DLP in F_{q^k} (hence in G_T) is as hard as in G_1, G_2, (5) k is small enough for efficient operations in G_T and (6) the sizes of r and q^k provide at least an 128 bits security level in G_1, G_2 and G_T. Such elliptic curves are called *pairing-friendly*.

A survey of pairing-friendly constructions can be found in [3]. In general, the smallest ρ-values are achieved when representing q, t, r as *polynomial families* $q(x), t(x), r(x) \in Q[x]$. This idea was introduced in [2]. In this case, pairing-friendly parameters are generated by evaluating these polynomials at some integer x_0, such that $q(x_0)$ and $r(x_0)$ are both primes and the *CM-equation* $4q(x_0) - t(x_0)^2 = Dy^2$ is satisfied, for some square-free $D > 0$ and some integer y.

The complexity of the DLP in G_1, G_2 is $O(\sqrt{r})$, due to Pollard's rho method. The complexity of the DLP in the extension field F_{q^k} depends on its characteristic and the embedding degree k and it is measured asymptotically by the L-notation:

$$L_N[\ell, c] := \exp\left[(c + o(1))(\ln N)^{\ell}(\ln \ln N)^{1-\ell}\right], \quad \text{where} \quad N = q^k \tag{1}$$

for some real constants $\ell \in [0, 1]$ and $c > 0$. For an extension F_{q^k}, the NFS attack has complexity $L_N[1/3, 1.923]$ for prime k. In the case that k is composite, recent variants of the TNFS method [4, 5] reduce the DLP complexity to $L_N[1/3, 1.526]$.

The improvements of the TNFS method have a major effect on the construction of pairing-friendly curves with composite embedding degree. The most important consequence is that the extension field must be larger than before and thus the condition $\rho \approx 1$ may not be ideal for composite k any more. For example, the Barreto-Naehrig (BN) curves [1] for $k = 12$ were optimal for generating a 256 bit prime r and a 3072

bit extension field (i.e. $\rho \approx 1$). Such parameters correspond to an 128 bit security level. After the improvements of the TNFS method, an extension field of this size reaches a security level of 110 bits. In order to achieve an 128 bit security level, one should now choose q^{12} around 4608 bits.

Considering the impact of the TNFS variants, in this paper we revise the criteria for choosing polynomial families $(q(x), t(x), r(x))$. For composite embedding degrees we propose the use of families that are likely to produce a balanced security level between G_1, G_2 and G_T and produce pairing-friendly parameters that are resistant to TNFS attacks. Additionally, for prime values of k we recommend the use of polynomial families that achieve balanced security levels. However, some of these polynomial families were not considered before due to a larger ρ-value. All families we present provide a security level of 128, 256 and 512 bits with $\rho \leq 2$. We produce numerical examples of cryptographic value obtained by the recommended families where the asymptotic complexity of the DLP in the extensions F_{q^k} is measured by Equation 1.

References

1. P. S. L. M. Barreto and M. Naehrig, *Pairing-friendly elliptic curves of prime order*, International Workshop on Selected Areas in Cryptography–SAC'05, pp. 319-331 (2005).
2. F. Brezing and A. Weng, *Elliptic curves suitable for pairing based cryptography*, Designs, codes and cryptography **37**, 1, pp. 133-141 (2005).
3. D. Freeman, M. Scott and E. Teske, *A taxonomy of pairing-friendly elliptic curves*, Journal of Cryptology **23**, 2, pp. 224-280 (2010).
4. J. Jeong and T. Kim, *Extended tower number field sieve with application to finite fields of arbitrary composite extension degree*, IACR Cryptology ePrint Archive (2016).
5. T. Kim and R. Barbulescu, *Extended tower number field sieve: A new complexity for the medium prime case*, Advances in Cryptology–CRYPTO'16, pp. 543-571 (2016).

Collaborative Multi-Authority Key-Policy Attribute-Based Encryption for Shorter Keys and Parameters

R. Longo[1], C. Marcolla[2], M. Sala [3]

[1] *University of Trento, Italy, riccardolongomath@gmail.com*
[2] *University of Turin, Italy chiara.marcolla@gmail.com*
[3] *University of Trento, Italy, maxsalacodes@gmail.com*

Abstract

Architectures relying on a single central authority often offer a great efficiency but suffer of resiliency problems and are quite vulnerable to attacks. In our proposal, a Multiple-Authorities Key-Policy Attribute-Based Encryption scheme is constructed in which the authorities collaborate to achieve shorter keys and parameters, enhancing the efficiency of encryption and decryption, since the key creation requires the private secrets of all authorities.

The scheme that we propose in this paper evolves from the scheme presented in [1] exploiting the collaboration between authorities to improve the efficiency.

Basically our scheme proceeds as follows: the first step is the creation of the parameters. Namely, each authority sets up independently its *master key* and then they collaborate together with the other authorities to create:

- a common *public key* used by users to encrypt,
- the *authority parameters* that will be used to generate *secret keys* (used to decrypt).

Once the *public key* is published, a user, who we will call Alice, chooses a set of attributes that describe her message and encrypts it using this key. Let Bob be another user, so he has an *access policy*. Suppose that Bob wants to decrypt Alice's message (note that he can do so if and only if the message has the attributes prescribed by his policy). Bob requests a *secret key* for his policy to every authority. Independently, each authority checks the policy pertinence and generates a secret key. Once he has obtained all keys, he can merge them and obtain a single compact key. In this way Bob may store and use them as a single key.

We prove our system secure under a variation of the bilinear Diffie-Hellman assumption, providing also a lower bound on its complexity.

References

1. R. Longo, C. Marcolla, M. Sala, *Key-policy multi-authority attribute-based encryption*, Algebraic Informatics, pp. 152-164. Springer (2015).

Conditional Blind Signatures

A. Zacharakis[1], P. Grontas[2], A. Pagourtzis[3]

[1] *National Technical University of Athens, azacharakis@yandex.com*
[2] *National Technical University of Athens, pgrontas@gmail.com*
[3] *National Technical University of Athens, pagour@cs.ntua.gr*

Abstract

We propose a novel cryptographic primitive that we call *conditional blind signatures*. Our primitive allows a user to request blind signatures on messages of her choice. The signer has a secret Boolean input which determines if the supplied signature is valid or not. The user should not be able to distinguish between valid and invalid signatures. A designated verifier, however, can tell which signatures verify correctly, and is in fact the only entity who can learn the secret input associated with the signed message after the unblinding process. We instantiate our primitive as an extension of the Okamoto-Schnorr blind signature scheme. We analyze and prove the security properties of the new scheme and explore potential applications.

Hash Function Design for Cloud Storage Data Auditing

Nikolaos Doukas[1], Oleksandr P. Markovskyi[2], Nikolaos G. Bardis[1]

[1] *Department of Mathematics and Engineering Science, Hellenic Military Academy, Vari - 16673, Greece.*
[2] *Department of Computer Engineering, National Technical University of Ukraine, (Polytechnic Inst. of),Peremohy pr.*

Abstract

Cloud based storage is being widely used as a viable solution to the problem of data storage in contexts where financial and practical considerations prohibit the use of locally based hardware and software resources. User reservations and legal constraints however have given rise to questions about the verifiability of the integrity of the stored data, especially in the case of public cloud infrastructure. A new problem has hence arisen, that of auditing stored files in order to obtain Proof of Retrievability. Secure cloud storage systems are limited by the overheads they require in order to provide the required security levels. Combined use of cloud and local computational resources is necessary in order to enable the desirable user experiences. With increasing local processing capacities, the most significant relevance is encountered in the Big Data Processing paradigm. The volumes of data that need to be processed are overwhelming to such an extent that approaches which use unlimited amounts of power, for processing and storage are not feasible. This paper focuses on a recent study of hash function requirements for big data applications and an associated key â based hash function design technique that makes the real â time collection, summarization, analysis and decision making based on streaming data. A file auditing technique is proposed that uses fundamental big data mass processing operations in order to.

Keywords: Secure cloud storage, Big data, proof of Retrievability, data auditing, data summarization, hash functions, data mining

Introduction

Key based search is a fundamental component for various data processing routines, such as real â time collection, summarization, analysis and decision making. The progressive development of information systems and of information integration determines a dynamic increase of the volume of the key indices upon which searches are performed [1]. A particular field of application of key based searching, along with other operations based on keys is the Big Data Processing paradigm, especially in Data Mining applications. The term Data Mining in this case, implies a process by which a representative model is sought that adequately describes data sets whose size renders them unmanageable via the classical sequential or batch processing concepts of algorithmic design. The representative requirement that is

set for such models inherently dictates that the type of models that are sought are statistical models.

The process of mining into data in streaming form and seeking to develop the corresponding model, may be described as learning the data, i.e. following the changes in the characteristics of the data and continuously updating critical parameter estimates that lead to decision making. The problems of interest that fall in this type of applications may be in general classified in two categories:
Problems where a summary description of the data is required, e.g. when tracking the progress of a measurement or a statistical measure related to the data, like a mean or a higher order moment, or trying to obtain a representative subset of the data.
Problems where characteristic features need to be determined that allow the determination of data values that usually appear in groups or data values that are similar, in some sense. Depending on the context, this class of problems may also be described as seeking to determine outlying values in the data stream.

Summarization algorithms may in some cases produce a single measurement that determines the merit or the rank of a complex data value or may also track the progress of a series of measurements that enable the process of decision making. Characteristic features correspond to data values that present the extreme variations within a data set and hence may be described as outstanding or as outliers. Being able to determine characteristics features serves multiple purposed, such as describing complicated interconnections between measurement vectors observed in different situations by only a limited number of values.

A fundamental problem in all the above applications is the requirement for accelerating searches based on a key, as well as the selection of this key, so that it exhibits the required characteristics, given the nature of the data. The necessity for scalability in such systems and increase of the volumes of the data that are required to be processed, has as a consequence imposed stricter requirements for the efficiency of search procedure results. More specifically, a large proportion of big data processing systems, in which key search is actively used, operate in real time conditions, on hardware that eventually is shown to have limited capabilities. Based on these facts, the development of efficient key based search technologies is necessary in order for the system to be scalable and viable.

The concept of Proof of Retrievability (POR) was introduced in [2] in order to describe the process by which a user may obtain from a storage service provider, such as a cloud storage service, evidence that the data file(s) of interest may be retrieved in their entirety and with guaranteed integrity. The concept is similar to that to the cryptographic proof of knowledge, but specific to large and time varying files or bit â streams. Factors of significance include the numbers and size of the exchanges that need to take place between the user and the provider, the numbers of seek operations that the provider is required to make on the data and the amount of data the user, or owner, of the data needs to store locally. An important requirement is that the user should be able to update the data and insert new items, with minimal necessity for computationally intensive recalculations.

This paper is organized as follows. In the following section an analysis of existing search technologies is presented, that illustrates how computational complexity increases exponentially with data size. Following that, fundamental big data analysis operations whose implementation is based on suitable hash functions are outlined. The importance of suitably designed hash functions for achieving computational efficiency in big data processing contexts is hence explained. The concept of proof of recoverability protocol design for secure cloud storage is explained. A recently proposed class of hash functions is hence presented that provides suitable statistical separation properties while maintaining a linear rate of increase of the computational effort involved. Subsequently, the application of this class of functions in fundamental big data processing problems is outlined. A POR protocol design is hence presented that is consists of a sequence of these fundamental operations. Conclusions are hence drawn for the mathematical specifications required for hash function design for such protocols.

Analysis of Existing Search Technologies

An important factor for improving the efficiency of key based search, is the incorporation of the multilevel memory organization of modern computational systems into search algorithms. In current conditions, where the volume of indices is constantly increasing and the efficiency requirements upon the search are becoming ever more demanding, the applicability of binary trees and B trees is significantly reduced, given the dependence of the search time on the volume of the key index.

For most applications, hash addressing is considered to be the most efficient search technique. There exist practical applications of key â based search, where the key index is considered permanent or quasi-permanent (the computational load required for the key based search procedures exceeds by several orders of magnitude the computational load of the procedures for changing the search index).

During a hash search in permanent key indices, it is possible to determine a one-way hash transformation that eliminates collisions. In bibliography, this class of methods is referred to as perfect hash addressing [1]. The fundamental advantage of perfect hash addressing is the absence of collisions, i.e. the key search time is determined by the time required for a single memory access. This permits the search schemes to attain maximum search speed, independent of the volume of the search index [1]. The exercise of determining a hash transformation for perfect hash addressing exhibits exponential complexity. For the completion of this exercise, a series of methods have been proposed in bibliography [1], [3], [5], [6] and [7]. The disadvantages of these methods are that they do not take into account the multilevel organization of memories and that they do not allow changing of the keys during operation. The purpose of this research is the modification of the organization of hash searches so that it becomes oriented to the quasi-permanent nature of the key index. Additionally, developments are sought in the mathematical model of such hash searches for the optimization of its characteristics.

A hash search organization was recently proposed [8] that uses quasi â permanent keys in conjunction with perfect hash addressing and probing. A mathematical model of the hash search in multilevel memory has been developed that allows the optimization of the hash memory parameters during the design
Big Data Mining Operations based on Hash Functions

Data mining, as it has already been defined and described in a previous section is a topic that has attracted significant research interest in the context of mining when the amount of data is characterized as ”Big ”, i.e. despite the evolution of processors, multiprocessor information processing systems and the ample availability of storage facilities, it is still infeasible to plan the processing, in whatever form of all the data. It may be impossible to process the data in real â time, at the rate at which they arrive or it may be economically or physically impractical to store all the data for online processing. Such limitations affect the capability of data owners (users) to perform POR calculations on their own infrastructure. This section summarizes common required functionalities of big data processing systems [9], [10], [11], where all the above restrictions may be satisfied via the use of appropriate search techniques that are based on suitably designed hash functions.

Shingle Hashing

A large class of applications need to process observations that can be mapped to a group of objects appearing together. An example of such application is the processing of texts in the form of characters. Typically, the text will be segmented into groups of characters of equal length, called shingles [9]. Typically, since objects are not arithmetic values, an encoding is required for representing those objects that is not based on the frequency of appearance of the corresponding shingles, but rather on the size of the domain from which sample observed objects are drawn. The hashing operation can be hence seen as a classification operation with the number of categories (buckets) being significantly larger than the number of possible different samples. The pre-requisite is that the hash function is collision â free, i.e. it will not classify different shingles into the same bucket, despite of the existence of empty buckets. In the case of document processing, the use of this type of hashing is an enabling technique for processing and comparing documents that are of sizes that are impossible to store into memory in their totality simultaneously.
Min Hashing

As the numbers of the objects under consideration increases, it becomes increasingly infeasible to store even all the possible the compressed shingle â based representations, corresponding to the pre â processing of all possible objects, in memory for drawing global conclusions.

Object		S1	S2	S3	S4	S5
Shingle	a	0	0	0	1	0
	b	1	0	1	1	0
	c	0	0	0	1	1
	d	0	1	1	0	1
	e	1	1	0	0	1

Table 1. Representation of an object as a binary table

In order to overcome this problem, an alternative representation of an object needs to be defined. Consider a set of 5 possible shingles {a, b, c, d, e } and objects consisting of collections (sets) of these shingles. It is then possible to represent these objects using a binary matrix, such as the one shown in Table 1.
A value of 1 in a cell of the table shown in Table 1, means that the shingle corresponding to that row is part of the object of the corresponding column. The representation of Table 1 is therefore interpreted that object S1 = {b, e }, S2 = {d, e }, S3 = {b, d }, S4 = {a, b, c } and S5 = {c, d, e }. The set of all objects is hence represented, as far as the search for similarity is concerned as a binary matrix of size NxM, where N the number of possible shingles and M the total number of objects being processed. Furthermore, this table is, for large N and M, in general a sparse matrix and the corresponding savings in the required amount of memory and processing effort required may hence be achieved. It is trivial to deduce that two equal objects will exhibit equal columns using this representation
Consider a permutation of the rows of Table 1. The minhash signature of each object of this permutation is defined as the index of the first row of the permuted Table 1 where a value of 1 appears for this particular object. An example permutation is shown in Table 2.
The minhash signature for each object is the set of the minhashes of that object for all possible permutations of the table. It can be shown [9] *that the probability of two objects exhibiting the same minhash signature, is a similarity measure for the two objects that approximates quantitative similarity measures based on vector distances*.

Object		S1	S2	S3	S4	S5
Shingle	a	1	1	0	0	1
	b	0	1	1	0	1
	c	0	0	0	1	1
	d	0	0	0	1	0
	e	1	0	1	1	0

Table 2. Permutation of the rows of the binary table

For this permutation, the minhash for S1 is 1, for S2 is 1, for S3 is 2 for S4 is 3 and for S5 is 1.

Generate the binary shingle table TGenerate a table S of size KxM, setting all cells to ∞Generate K hash functions *h1 â hK* mapping K items K bucketsFor each value i from 0 to K - 1
Calculate *h1(i), h2(i), â, hK(i)* For each column j from 1 to M â 1

For each value i from 0 to K - 1

If $T(i, j) = 0$, do nothing

If $T(i, j) \neq 0$, $S(i, j) = min \{S(i, j), hi(r) \}$.

Algorithm 1. Minhash signature similarity calculation based on hash functions

In practice, the number of shingles will be large and it is therefore computationally expensive to examine all the permutations of the rows of the table. Additionally, it becomes computationally expensive, even to generate random permutations of the rows. This problem is overcome by defining a number K « N, of permutations to be considered. An algorithm for estimating the similarity if the minhash signatures of the objects may then be estimated as shown in Algorithm 1.The success of Algorithm 1 is critically dependent on the hash signatures not presenting collisions.
Locality Sensitive Hashing

The Technique of Section 3.2 becomes infeasible as the table T of Algorithm 1 becomes larger, with increasing numbers of objects. Even though the amount of information, or the number of bits, that need to be processed per object is small, the number of pairs that exist is infeasibly large. As a consequence, the number of comparisons that needs to be performed is correspondingly prohibitive for computation in some cases with reasonable resources, while in other cases the computations cannot be realistically implemented. For example, with 106 objects to be compared, the number of comparisons is of the order of 1012 and even with a hash signature of 1KB, it would take about 6 days to complete in a powerful modern personal computer [9].

The solution taken is a two pass approach; first the binary signature table is split into horizontal bands, i.e. into bands of groups of shingles. Subsequently, smaller hash functions are applied to hash all objects of a band to a small number of bins. The same process is applied to all the bands. Objects that hashed into the same bucket in multiple bands are considered to be candidate pairs and those are compared using the full minhashing procedure described in the previous section.

The hash functions used for the first pass of the Locality Sensitive Hashing, map many objects in the same buckets, i.e. present a large number of collisions. However these collisions need to satisfy similarity conditions. If d(x, y) is the distance between the subset of shingles examined for a specific band and pair of objects then: If $d(x, y) \leq d1$, the probability of x and y hashing to the same bucket is at least p1

If $d(x, y) \geq d1$, the probability of x and y hashing to the same bucket is at most p2

Suitable determination of the values of p1 and p2 attain a balance between the number of erroneous proclamations of candidate pairs and the number of misses, i.e. true candidate pairs are classified as irrelevant.

Stream Data Sampling

When a stream of data is necessary to be processed in real time and the number of unique elements is large, unknown and random, a common requirement is to be able to derive a representative sample of the incoming values. Assuming for example that the incoming tuples are {*object _identity, variable _value* }, where the object is one of the unique entities and the variable is some observation, a natural inclination would be to select a portion of the objects, such that storage and processing is economically or environmentally viable and keep only tuples originating from these objects. This could be performed by observing the following procedure:

For each incoming tuple

- Draw a random number in the range 1 â N, where N is the number of unique entities
- If the number drawn is less than M, where M is an integer such that M/N is equal to the portion of unique objects that are to be maintained, appropriately process the tuple

However, this approach requires that the numbers and identities of the unique objects are known. This assumption is severely restricting, since applications of interest usually deal with very large, unknown and rapidly varying numbers of unique objects (e.g. the unique users visiting an e-shop). Additionally, it may be shown that such an approach would fail when a tuple appear multiple times [9], an occasion that is of particular interest.

This problem may be addressed by use of suitable hash functions. The number of buckets is determined such that it is feasible to select the proportion of objects required. If for example a ratio of M/N of the overall tuples need to be maintained, then the number of buckets may be assigned to N. A suitable hash function is required that will map incoming tuples to the N buckets and only buckets in the range 1 â N will be taken into consideration. Again, given that the number of unique tuples is extremely large, and the number of buckets is comparatively very small, the hash function is one that presents large numbers of collisions. However it is important that all buckets are selected with equal probability. Given this implementation, it becomes relatively easy to resize or alter the sample size or the proportion by altering the parameters or switching the hash function in real â time and waiting for a significant number of samples to arrive and any transients to die out.

A Hash Search Model for Quasi-Permanent Indices

This section analyses design approaches for hash search models and presents a hash function design model [8] that produces quasi â permanent indices.

The purpose of the hash search model presented is to incorporate the analytic form of the dependencies between the characteristics of the hash-memory that determine its organization into the model and enable it to solve problems of optimization of

the architecture of hash memory during the design phase. At the basis of the model that will be presented, lies the concept of the determination of a hash transformation *H(X)* the ensures the mapping of a given set Ω from *m* keys in *s* pages of hash memory in such a way that the entire set of keys that are classified in each of the pages do not exceed the value ($\alpha+ \delta$) ·w, where α is the load factor of the hash memory, δ the allowed variability of the load of a hash memory page load and $\alpha+ \delta \leq 1$. The load factor α of the hash memory is defined by the relation between the set of *m* stored records to the maximum feasible record count $M = sw$ that is determined by the size of the memory:

(1)

As a record, one may consider the information taken as the key associated to a particular data item. The reference address of the position where the data is stored may be found in the record instead of the data. The determination of the hash transformation *H(X)* that satisfies the above condition may be done by trial and error. As the test mechanism for the hash transformations, it is proposed that the prototype, block â based cryptographic algorithms (DES or Rijndael) be used, that incorporate the one-way cryptographic encoding using the key *K*, of the data *D* in the codeword *C: C = HK*(*D*) [4].

The key for the search data *X* in this case is used as input data to the cipher block whence the key *Đ* of the cipher block assumes the role of synchronization code and actually appears together with the number of the hash transformation. The resulting code *C* of the cipher block is divided in two parts: an *h*-bit packet that serves as a hash address *AK*(*X*) of the page and the remaining bits that become the hash Sign *SK*(*Đ*) of the key *X* of the search [7], [8]. Consequently the choice of the Hash transformation *HK*(*Đ*) is attained via the procedure of changing the key *Đ* of the cipher block.

The analysis of the mathematical model given in [8] demonstrates that, the compromise involved in the hash search, exists in the selection of the number of the pages among which exchanges take place between the main and the cache memory. The analysis leads to the conclusion that the search speed essentially depends on the time for transferring the arranged hash page addresses from the main hash memory to the cache memory, which in turn depends on the size of the pages. Consequently, from the point of view of attaining high speed hash searches, the page size needs to be reduced. At the same time, reducing the time required for selecting the hash transformation requires according to (5) an increase of the page size. A resolution of the above compromise may be found by the defined frequency of the key index reconstruction; the more frequently new keys are assigned, the smaller the required time for selecting the hash transformation and the larger the page size δ may be. The resolution of these contradicting requirements may be attained either by increasing

the size of the pages or by reducing the proportion of the hash memory that is occupied. Within this context:

A hash search model was developed that corresponds to the nearly constant key index case. The model takes into account the multilevel memory organization of modern computational systems.
The basis of the model that was developed, proposed the organization of hash searches in nearly constant key indices. It was shown that the search time is defined by access to no more than low level memory pages.
In the case of shingle hashing, the proposed method provides satisfactory results since it can be guaranteed by consideration of the encryption algorithm that each value will produce a unique value, which is the fundamental requirement for this operation
In the case of min hashing, it may be similarly guaranteed that the proposed method will permute the rows of the signature table and this permutation will have randomness characteristics.
In the case of locality sensitive hashing, the hash function is required to maintain similarity criteria, i.e. small variations in the input need to leave the output unchanged. This requirement is fundamentally contrary to the operation of the encryption algorithm. The design of a suitable substitute hash function is being investigated.
In the case of stream data sampling, it is required that hashing different tuple key identities is mapped with uniform probability to a small number of buckets. This is consistent with the operation of the encryption algorithm and is hence satisfied by the proposed method.
POR protocol design

The problem of efficient Proof of Recoverability [2], [13] protocol design is equivalent to storing data securely in storage facilities that are not under the direct control of the data owner (user). There exists therefore one more entity that participates in such transaction, namely the storage provider (provider). Without loss of generality, the data may be considered to be contained in a single, suitably encoded file. The user entrusts the data to the provider for storage. At any instant the user may request to audit the data. The user performs audits by verifying the integrity of the data and the fact that the file can be recovered in its entirety by suitably interacting with the provider and giving them challenges to which they return with responses. The series of exchanges taking place is governed by a suitably designed POR protocol. This auditing is required so that the user can confirm that the provider is suitably preserving and maintaining the data, e.g. by protecting against equipment failure, accidental erasure, malicious actions etc. and does not attempt to modify it. An additional danger is that the provider dumps the data in order to cut costs and in the hope that the user will not need to access data that may be old, obsolete or simply backups that are very rarely used.

Due to the data size considerations explained at the beginning of this article, the fundamental requirement from a POR protocol is that the audit is performed without the user or the provider having to process all the data. The computational

load necessary for the POR exchanges calculations should be balanced so that the largest proportion of this load is performed by the provider, who is assumed to possess significantly more powerful infrastructure. POR schemes may be classified as publicly verifiable, if anyone can verify integrity or privately verifiable if only the user is capable of performing the audit, by means of some secretly kept information (key).

A brute force approach to solving the problem would be to download the entire data and perform any calculations necessary. As it has been extensively analyzed however, such an audit procedure would in general be impractical, since the user may not be assumed to possess neither the space, not the time and processing power to perform this action. Ideally, a POR protocol is required to be [14]:

- Efficient: measured in computational complexity, local storage and communication burden. Ideally the overall measure of complexity should be linear to the level of security measure
- Publicly verifiable: this way a trusted third party auditor needs to be assigned the task, a public one is preferred. Auditing service providers could hence be established, similarly to digital signature authorities.
- Publicly retrievable: Any third party should be able to recover the data. It should therefore be shown that an attacker cannot overcome the POR security, even if they had access to all data. Separate access control functionalities should protect the data from unauthorized access.
- Stateless: While the data is fixed, the user should only be required to store only private key information but not any state information.
- Privacy preserving [15]: The auditors should not be able to derive content based on the auditing data.

A stricter design process would require from the POR to be able to properly detect anomalies even if a malicious provider manages to answer correctly a portion of the challenges [16].

The proposed scheme

In this section, an outline will be given of how a POR design problem may be addressed via a sequence of the operations described in Section 3. The user equipment treats the data as a monolithic, block based file system, but outsources it as a single, serially accessible file of size M. All data is suitably encoded so that all data have uniform appearance, e.g. plain or uuencoded 32 â bit or 64 â bit words. The high level perceived organization of the file is the sectored structure produced by the locality sensitive hashing. The structure is predefined by the user. A maximum data size S is defined, with S » M. The user maintains a local data organization table with rows and columns as described in Section 3.1. The numbers of rows and columns is a design parameter defined by the user.

For each data item (user file) a record is kept locally by the user with filename, address and size. An identifier is constructed from this record, based on which the item is hashed and stored onto the appropriate cell (row and column) of the file structure.

For each data item, a number of challenge â response results are pre â calculated. Each response is stored in different files that are decided based on hashing the response text and the challenge number. The number of challenges and the number of times each one is applied are design parameters decided by the user. A map of the locations of the first challenge for each data item is stored locally by the user.
Data blocks may optionally be padded with pseudo random data, generated by suitable hash functions from the payload data of the actual data item, so that all data items appear to be of equal size. Empty data blocks may be similarly padded according to the same lightweight cryptography principle.
Auditing involves the comparison of a randomly selected subset of data. Columns are permuted based on the principle of min-hashing described in Sections 3.1 and 3.2. The data items of first resulting row or a randomly selected row are then used as the object of the challenges.
For the provider, an audit consists of the retrieval of a series of words from the data. The user performs a number of simple hash calculation operations and value comparisons.
For the initial data commitment to the provider, all calculations may be performed on powerful provider equipment. The user submits a series of calculations to the provider and all results are returned back to the user. The user then embeds the responses into the appropriate files.
Data insertions and substitutions involve substituting the data items containing the payload data and the challenge response data. The new response data are stored in different data items from old ones, as the data and hence the hashes have been modified.

The proposed scheme will be optimized and implemented in order to draw final conclusions about the parametrizations necessary.

Conclusions

In this paper, the use of hash functions that are suitable for achieving computational efficiency when extracting information from extremely large sets of data was used for data file auditing. The auditing is necessary for obtaining POR when storing data in cloud resources. A technique that was recently proposed in order to achieve quasi â permanent keys was employed to design appropriate hash functions. Requirements for the design of efficient POR protocols were explained. Hash functions were employed in order to implement a series of fundamental data manipulation primitives. These primitives were combined in order to construct an implementable PRO protocol.

References

Berman F., Bock M.E., Dittert E., O‘Donnel M.J., Plank D. Collections of function of perfect hashing // SIAM Journal Computers. â 1986, - Vol. 15, â2, - P. 604-618.
Juels, Ari, and Burton S. Kaliski Jr. "PORs: Proofs of retrievability for large files." Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007.
Czech Z.J., Havas G., Majevski B.S.. An Optimal algorithm for generating minimal perfect hash functions.//Information processing letters. 1997, - 43(5). p. 257-264.

Jagannathan R. Optimal partial-match hashing design // ORSA Journal of Computing. â 1991, - Vol.3, â2, - P.86-91.
Ningping Sun, Ryozo Nakamura, Nonbing Zhu, Akiro Tada, Wenling Sun. An analysis of average search cost of external hashing with separate chain.// Processing of 7-th WSEAS International Conference on Circuits, Systems, Communications and Computers (CSCC-2003).- 2003,-P. 315-324.
Ramakrishna M.V., Bannai Y. Direct perfect hashing function of external files. // Journal of Database Administration. â 1991, - Vol.2, â1, - P.19-28.
Polymenopoulos A., Bardis E.G., Bardis N.G., Markovskaja N.A., "Perfect Hashing Using Linear Boolean Functions", WSEAS Press â Problem in Applied thematics and Computational Intelligence, ISBN: 960-8052-30-0, 2001, pp. 5-11.
Bardis E.G., Bardis N.G., Markovskyy A.P., Spyropoulos A.K., "High Storage Utilization of Hash Memory by Reducing of Information Redundancy for Hashing". Submitting as a special issue of IMACS/IEEE CSCC'99 International MultiConference, "Software and Hardware Engineering for the 21th Century", ISBN: 960-8052-06-8, pp. 272-276, 1999
Bardis, Nikolaos G., Nikolaos Doukas, and Oleksandr P. Markovskyi. "Hash addressing of the quasi-permanent key arrays in multilevel memory." Journal of Applied Mathematics and Bioinformatics 3.4, pp 91 â 105, 2013.
Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2014.
Kharchenko V., Illiashenko O. Concepts of Green IT Engineering: Taxonomy, Principles and Implementation. Inbook: Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, V. Kharchenko, Y. Kondratenko, J.Kacprzyk (Eds.), Vol. 74. Berlin, Heidelberg: Springer InternationalPublishing, 3–20 (2017) , DOI: 10.1007/978-3-319-44162-7 _1
Kondratenko Y.P., Korobko O.V.,Kozlov O.V.: PLC-Based Systems for Data Acquisition and Supervisory Control of Environment-Friendly Energy-Saving Technologies. In book: Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.), Vol. 74.Berlin, Heidel-berg: Springer International Publishing, 247–267 (2017),DOI: 10.1007/978-3-319-44162-7 _13
Chen, Fei, et al. "Secure cloud storage meets with secure network coding." IEEE Transactions on Computers 65.6 (2016): 1936-1948.
Shacham, Hovav, and Brent Waters. "Compact proofs of retrievability." International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2008.
Wang, Cong, et al. "Privacy-preserving public auditing for secure cloud storage." IEEE transactions on computers 62.2 (2013): 362-375.
Wang, Cong, et al. "Privacy-preserving public auditing for secure cloud storage." *IEEE transactions on computers* 62.2 (2013): 362-375.

Method for Accelerated Zero-Knowledge Identification of Remote Users based on Standard Block Ciphers

Nikolaos G. Bardis[1], Oleksandr P. Markovskyi[2], Nikolaos Doukas[1]

[1] *Department of Mathematics and Engineering Science, Hellenic Military Academy, Vari - 16673, Greece. nbardis@sse.gr, ndoukas@sse.gr*
[2] *Department of Computer Engineering, National Technical University of Ukraine, (Polytechnic Inst. of Kiev),Peremohy pr., Kiev 252056, KPI 2003, Ukraine. markovskyy@i.ua*

Abstract

Identification of remote users that implements the cryptographic concept of zero knowledge strict identification. In contrast to existing methods that implementing this principle and are based on modular arithmetic, the proposed solution is based on the use of standard block ciphers. This solution may significantly accelerate the process of strict user identification. Existing zero â knowledge user identification techniques are revised. The proposed procedure for user registration and for the execution of a round of identification. Theoretical and experimental evaluation of the effectiveness have demonstrated that the proposed method achieves an acceleration of the identification process by two orders of magnitude, compared to existing schemes.

Index Terms - multi user systems, remote users identification, methods of strong identification, zero-knowledge identification, block ciphers.

1 Introduction

The efficiency of information security algorithms is defined based on two factors: the level of security and the amount of computational resources required for the implementation of the security functions. At the basis of all information security algorithms lies an analytically insoluble mathematical problem. In practice, such problems are one way transformations of the form $Y = F(X)$. For such one way transformations, the forward transformation F(X) is defined such that there is no analytic way of deriving the inverse transformation () of the known function $F(X)$ such that $X = \Omega(Y)$. The only way to accomplish such a task, i.e. the determination of a value X for a given value Y such that $Y = F(X)$, is to tabulate all possible values of X. The largest part of the problems that do not have an analytic solution and form the basis of cryptographic algorithms, originate from the Boolean algebra and Number Theory. Especially for number theory, problems are related to discrete logarithm calculations and are the basis of most asymmetric cryptographic algorithms (public key encryption algorithms). Well known algorithms belonging in this class are RSA or El-Gamal, as well as digital signature algorithms like DSS [1]. The basis of another broad class of cryptographic algorithms, lies the difficult to solve Boolean

algebra problem of finding the roots of a system of non-linear Boolean functions. In this class belong all symmetric encryption algorithms such as DES, IDEA, Rijndael, as well as a significant proportion of hash RIPEMD-160 [2].

The basic advantage of algorithms based on number theory problems that cannot be solved analytically, is the existence of multiple keys. This can be illustrated in the cases of RSA, El-Gamal, EEC where the keys for the forward and backward transformation are different. Hence, when there exist multiple keys, some of them may be used for decryption and some of them may be used for encryption. This enables more efficient information security schemes to be created on this basis, that permit a more efficient organization of information resources, compared to algorithms based on a single key.

This observation was the motivating force for the creation of the public key encryption algorithm RSA in 1978, the basis of which was the one-way transformation , that is connected to âa new era in information security technologyâ [2]. From a theoretical point of view, the existence of multiple keys in cryptographic transformations is necessary so as to create multiple solutions, i.e. there exist at least two keys $X1$ and $X2$, such that The basic drawback of algorithms based on the principle of the difficult to solve number theory problems, is the low speed of implementation that arises from the high computational complexity of the operations of modular exponentiation in numbers whose size is of the order of several thousand bits.

This drawback is eliminated with information security algorithms that are based on difficult to solve Boolean algebra problems. The recursive computation of a system of Boolean equations may be organized efficiently enough, via software in general purpose processors or on special computational hardware. As far as evaluation is concerned, the speed of an implementation of a Boolean function based algorithm and that of a modulo arithmetic based algorithm, differ by about a factor of 2-3 orders of magnitude [3]. However, algorithms based on Boolean transformations have fewer functional possibilities that do not permit the creation of efficient information security protocols, similarly to those based on analytically impossible to solve number theory problems. More specifically, the use of Boolean transformation does not permit the implementation of asymmetric cryptography, a very significant principle of modern cryptography, the encoding of digital signatures and the identification based on the basis of a âzero knowledgeâ scheme. One of the factors that contribute to the limitation of the functional capabilities of Boolean transformations, is the uniqueness of Boolean transformations [3].

One of the most promising research areas of current information security technologies is the extension of the functional capabilities of Boolean algebra based algorithms [2]. On the basis of such algorithms, the design of efficient information security protocols will be enabled. Such efficient information security protocols require significantly less computational resources and may hence be executed at speeds that are several orders of magnitude faster than the corresponding ones for

modular arithmetic based problems.

Towards the realization of this direction of modern information security technology, an important goal is the development of a methodology for the design of one way Boolean transformations that possess the non-single-value property [3]. Such algorithms may be efficiently used in protocols for the âzero knowledgeâ based identification of remote subscribers in multi user systems [4].

An important problem regarding multi-subscriber distributed systems, such as the networks used for multimedia content delivery, is user authentication and user rights management. Due to the nature of such networks, it is imperative that the security protocols are robust but minimize the storage needs and the computational complexity. Analyses of the efficiency of such protocols and ways of improving it, have been presented in [5], [6], [7] and [8]. Previous work on the front of remote subscriber identification was focused on improving efficiency and increasing the level of security by reducing the need for storage and communications [9]. The aim of this research is the development of a method for designing Boolean functions that have the necessary one way properties and present non-single-value during the inverse transformation.

In its general form, the idea of the use of Boolean transformations for the implementation of identification based on the principle of zero knowledge consists of the following steps. Subscriber A produces, via the pre-defined procedure a Boolean transformation algorithm, $F_A(X)$ such that there exists a finite set Î©A of m incoming vectors $\Omega_A = X_A1, X_A2, , X_Am$, for which the transformation $F_A(X)$ assumes the same values $U_A : Q, G \in \Omega_A, Q : F_A(Q) = F_A(G) = U$. The set Ω_A practically contains a list of session passwords of subscriber A, that are known only to subscriber A. For this reason it may be said that, the set Ω_A is considered to be the private key of subscriber A. The function $FA(X)$ and the password U_A, are considered to be the public key of subscriber A and during the registration of the subscriber, it is communicated to the system. During access to the system, the subscriber chooses a sequential password G that belongs to the set Ω_A and sends it to the system. The system calculates $F_A(G)$ and compares it to U_A. If $F_A(G) = U_A$, then the subscriber A is considered to have been successfully identified. The proposed identification scheme base on the zero knowledge principle is designed for identification in m sessions. After that, a new registration is required. Taking into account the non-reversibility of $F_A(X)$, the system may not in any other way acquire the passwords that compose the set Ω_A.

2 Analysis of the problem of the efficiency of identification based on the concept of zero knowledge

According to current trends in technological development and its applications, there exist increased possibilities for unauthorized access to sensitive information resources of the integrated systems, possibilities that are enabled by interventions to the user identification procedures.

It is a well understood fact that the increased use of wireless data transmission technologies makes it feasible for illegitimate users to mount attacks during the stage of user identification. Specifically in the case of wireless communications actions like the sniffing of passwords for access of legal subscriber, as well as his replacement after the session of identification are facilitated. A robust defense mechanism against imitation of legitimate users is the periodical repetition of the user identification procedures during the interaction of the system with a subscriber. For this reason the process of identification should be such that it enables fast implementations.

Additional ways for illegitimate interventions during the identification process are the side-channel directed interactions with the system simultaneously with legitimate users, the use of viruses or via the actions of irresponsible personnel. For the broad class of commercial multi-subscriber systems the elimination of the possibility of impersonation of user access by imitation of access codes is important.

On the basis of the circumstances indicated, the current means for subscriber identification must satisfy the following requirements [1]:

1. The identifying information message (password) must change with each access to the system and the passwords used must be statistically independent;
2. The length of password should be such that it completely excludes the possibility of a brute force attack;
3. The information, which is stored in the system must not be sufficient for the reproduction of subscriber passwords;
4. Identification procedures must be carried out sufficiently rapidly

In literature [1] identification methods, which satisfy the first three of the given requirements are classified as "strict", in contrast the remaining schemes that are classified as "weak". In the class of the weak schemes belong, for example, the procedure of identification which is used in the UNIX [1] operating system. This procedure involves the storage in the system of only the hash value of the passwords of users, that, with the use of the one way hash functions, excludes the possibility of the reproduction of password of the system; however, passwords themselves do not change, which makes it sufficient simple to intercept them.

The class of strict procedures is principally composed by methods of identification that are based the concept "zero knowledge".

The most commonly known of these methods is the FFSIS (Feige Fiat Shamir Identification Scheme) [4]. The basic computing operation of this method is the modular squaring of numbers, with a length of 2048-4096 bits. Based on applications of using this scheme in practice, the main disadvantage of FFSIS are:

- The necessity for a large number of data exchanges during the user identification process, which noticeably loads the communication channels used.
- The large computational complexity of the operation of modular squaring performed on numbers whose bit capacity is much larger than the processor capacity.

Other identification schemes that implement the concept of "zero knowledge" using modular arithmetic, such as Guillou-Quisquater [10], the Schnorr method [11] require a significantly smaller volume of transfers, but the procedures they provide are more computationally complex, thus, in place of squaring, they use operations of modular exponentiation.

Thus, the main shortcoming of existing implementations of the progressive concept of "zero knowledge" in identifying users is a long time, which is very critical for modern integrated systems of collective access with millions of users.

A promising direction of radical acceleration of the implementation of strict identification is the transition to a different mathematical basis [12]. The aim of this research is to improve the efficiency of identifying remote users by reducing the execution time of the corresponding computational procedures while maintaining a high level of security. The purpose of this research is the development of a modified schemes for zero knowledge user identification, which involves significantly smaller computational complexity and increases the speed of identification with software and hardware implementations.

3 Method for accelerated zero-knowledge identification

The acceleration of zero â knowledge identification may be pursued via the use of existing known methods, but also via the use of algebraic cryptographic transformations. It is known that the foundation of any cryptographic mechanism are non â reversible cryptographic transformations. In modern applied cryptography, one â way transformations from number theory and non â linear Boolean algebraic transformations are used.

The one â way transformations are the foundation for the class of algorithms known as modern public key cryptography and manifested by well-known algorithms, such as RSA and DSA. The non â reversible Boolean transformations are the mathematical foundation for all the symmetric cryptographic algorithms, including algorithms that are in most countries certified as standard such as Rijndael, all hash algorithms, such as SHA-1, RIPEMD -160, as well as stream cryptographic algorithms.

It is known that the fundamental advantage of using non â reversible transformations from number theory is the wide and advanced functionality. On the other hand, the principle advantage of the use of non â reversible Boolean transformations, is the significantly larger speed of execution of the cryptographic calculations. As a reference therefore, it may be stated [12] that for approximately the same level of security, symmetric cryptography provides faster execution of three orders of magnitude more compared to public key cryptography.

For the acceleration of user authentication according to the zero â knowledge principle, an implementation method was developed that uses non â reversible Boolean functional transformations. For the implementation of these transformations it is proposed that standardized block ciphers be used and more specifically Rijndael block ciphers.

An important advantage of the use of standard block ciphers is that they have undergone extensive and in â depth testing and therefore extensive experience from their use has been acquired. This fact guarantees a high level of security and simplifies the evaluation of the cryptographic security of the proposed method.

In this architecture, the Block Ciphers (BC) represent the encryption â decryption algorithm for the fixed length data block D, using a single key K. Later in this section the process of the execution of the encryption defined as C=F(D, K), as well as the complimentary process of the decryption D=R(C,K) are outlined.

The architecture of the cryptographic transformation associated with the proposed authentication method is illustrated in Figure 1.

The proposed method includes a user registration procedure and a user authentication cycle. During the registration, the following sequence of actions is executed.

1. The system sends to the user the authentication code U.
2. The user defines the number n that represents the number of authentication cycles.
3. The user produces the random session password *pn* at the end of the nth authentication cycle. The index j assumes the value $n-1$: $j = n-1$.
4. The user calculates $q_j = F(j|U, p_j)$, where $j|U$ is the of the number of the authentication cycle and the authentication code.

5. The user calculates $p_{j-1} = F(p_j, q_j)$.
6. The index j is decremented: $j = j - 1$. If $j > 0$, then return to step 4.
7. Send the code p_0 to the system.

The session passwords $p_0, p_1, , p_n$ are stored in user memory. The execution of the jth cycle of user authentication consists of the following sequence of actions:

1. The user sends the jth session password p_j to the system.
2. The system calculates $d = F(j|U, p_j)$.
3. The system calculates $\zeta = F(p_j, d)$. If $\zeta = p_{j-1}$, then the user authentication was successful and access to system resources is permitted.

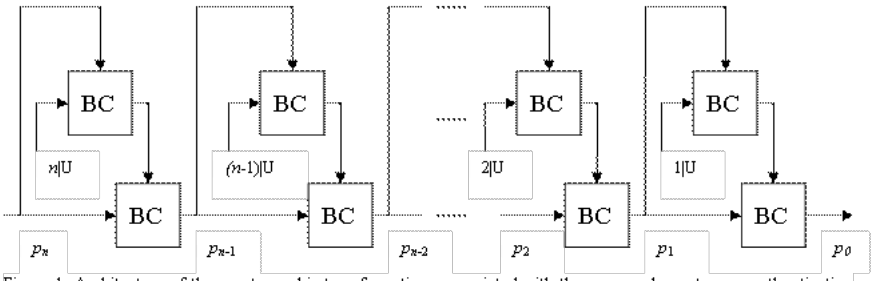


Figure 1: Architecture of the cryptographic transformations associated with the proposed remote user authentication method.

4 Effectiveness Evaluation

The principal effectiveness evaluation for the proposed scheme are the security level attained, regarding attempts illegitimate access to system resources, and the amount of processing power used for the calculations required for authentication.

For the intruder, the achievement of unauthorized access to the system is equivalent to acquiring the subsequent session password p_j of the user, based on the preceding sequence $p + j - 1, p_{j-2}, , p_0$.

In reality, the session password p_j is associated with the previous available password p_{j-1}, available to the intruder and is the functional equation $p_{j-1} = F(p_j, d) = F(p_j, F(\alpha, p_j)), where \alpha = j|U$.

If the intruder does not know the code U, then for breaching the security and obtaining illegitimate access, then they must find λ, μ and $p_{j-1} = F(\lambda,\mu)$, such that λ and μ are functionally connected with the equation $\mu = F(\alpha, \lambda)$ where the code α

is not known. This exercise may not practically be solved, since it requires the exhaustive testing of two components: Î±*andtheblockcipher*λ. The volume of tests for using the block cipher of the Rijndael algorithm with key length 256 is $2256 + L(U)$, where $L(U)$ is the length of the code U. Consequently, for the illegal calculation of a session password, the required amount of computational resources is double of the amount of computational resources required for breaking the block cipher.

The system is also not practically capable of creating the subsequent session password p_j of the user, since it only has available codes U and $p_{j-1}, p_{j-2},, p_0$. Given that the system does not know the value of $\alpha = j|U$, then for recovering the session password a p_j must be selected, such that it satisfies $p_{j-1} = F(p_j, F(\alpha, p_j))$. The volume of computational resources required for this task is equivalent to breaking the block cipher, a task beyond practical attainability. Another tactic for obtaining the session password of the user for the system, is to test all possible values for the initial session password pn of the user. Obviously, the solution to this problem requires n times more computational resources than the exercise of testing in order to recover the key of the block cipher.

Standard block ciphers have been exhibited to display resilience to different types of cryptanalysis. It has also been shown that the volume of computational resources required to break a block cipher via brute force, is beyond the scope of practical applicability. Consequently, the above consideration confirm the high security level attained by the proposed remote user authentication method, in terms of the resilience against possible attacks.

The principle advantage of the proposed method for remote user authentication in the context of the strict zero knowledge concept is to increase the speed of completion of the required calculation procedures. From the authentication procedure described above, it may be deduced that in each cycle of the user and the system the block cipher is calculated twice. Considering the use of Rijndael block ciphers of data packet size 256, the block cipher executes 14 rounds.

The data matrix and the key contain 4 rows and 8 columns. In each round, 32 substitution operations using tables are carried out, along with four 64 â bit shift operations, four logical addition of 64 â bit words to the key and 40 logical linear column transformation operations. In general, the execution of a round requires about 80 logical operations. Therefore the execution of the 14 rounds requires 1120 operations.

The FFSIS authentication operation involves 20 cycles for the exchange of data between the user and the system. In each cycle of the system, modular exponentiation is carried out, for sizes equal to the square of the bit size m of the number. In most modern protocols, $m = 2048$. In practice the number of size m bits is divided in d sections whose size is equal to that of the processor. For $m = 2048$ and for processor size of 64 bits, the section d is equal to 32. The modular exponentiation

to the square requires d^2 multiplications in the processor, plus some addition operations. The modular reduction operation involves, on average, m subtractions of m â bit numbers, each of which consists of d subtraction operations at the processor. Hence the time required for the execution of the modular squaring may be calculated as $d^2 \times (t_m + t_a) + m \times d \times t_a$, where t_m is the time required for multiplication at the microprocessor and ta the time for an addition or a subtraction operation by the processor.

Considering that the multiplication operation in modern processors is performed with about 10 times more computational effort than a logical operation and that an addition requires about 2 times the effort of a logical operation, it may be stated that for the evaluation, the execution of a modular squaring requires time equal to $32^2 \times 12 + 2048 \times 32 \times 2 = 143 \times 10^3$.

Comparing these estimates it is derived that the application of the two block ciphers is performed by about 64 times faster than the modular exponentiation of 2048 bits. Given that the FFSIS technology requires about 20 executions of the modular squaring, the estimated reduction in computational effort achieved by the use of the proposed method is 1280.

Experimental studies have demonstrated that the proposed method provides a real acceleration of the remote user authentication procedure by approximately three orders of magnitude.

5 Conclusions

As a result of the research presented in this paper, a method for the fast authentication of remote users was proposed that is conformant to the cryptographically strict concept of zero â knowledge. A particular characteristic of the proposed method is that focuses on standard block ciphers that have been comprehensively and extensively tested regarding their cryptographic properties. This permits the acceleration of the user authentication procedure, simplifies its application while guaranteeing high credibility in terms of security against attacks aiming to gain unauthorized access to system resources.

It was proved, both theoretically and experimentally, that the proposed method, using non â reversible Boolean transformations of block ciphers, requires for its implementation fewer computational resources compared to known methods that use modular arithmetic multiplication operations. The proposed scheme achieves an increase in the speed of the calculation of an authentication cycle by three orders of magnitude compared to the same existing techniques.

References

1. RSchneier B. (1995) âApplied Cryptography. Protocols. Algorithms and Source codes in Câ. Ed.John Wiley, 1995 - 758 pp.
2. Kurosawa K., Yoshida T. (1999) âStrongly universal hasing and identification codes via channelsâ. IEEE Trans. Information theory, v.45, no.6, 1999. pp.2091-2095.
3. Seberry J., Zhang X., Zheng Y. (1995) âNonlinearity and propagation characteristics of balanced Boolean functions.â Information and Computation Academic Press. 1995.-Vol. 119, â 1 -P.1-13.
4. Feige U.,Fiat A.,Shamir A. âZero knowledge proofs of identityâ // Journal of Cryptology, Vol.1, No.2 1988, P.77-94.
5. Nikos G. Îardis, Alex Polymenopoulos., Evgenios G. Bardis, Alexander P. Markovskyy, (2003) âMethods for Increasing the Efficiency of the Remote User Authentication in Integrated Systemsâ, TRENDS IN COMPUTER SCIENCE, Volume 12 No.1, ISBN 1-59454-065-9, Nova Science Publishers, Inc, New York, pp.99-107, 2003
6. Braz, C and Robert, J.M. (2006) âSecurity and usability: the case of the user authentication methodsâ. Proceedings of the 18th International Conference of the Association Francophone d'Interaction Homme-Machine. 2006, pp 199-203.
7. Wang, H., Sheng, B, Tan, C and Qun, L. (2008) âComparing Symmetric-key and Public-key Based Security Schemes in Sensor Networks: A Case Study of User Access Controlâ. Proceedings of the 28th International Conference on Distributed Computing Systems, 2008, pp 11-18.
8. Jia-Lun Tsai (2008) âEfficient multi-server authentication scheme based on one-way hash function without verification tableâ, Elsevier Computers Security, Volume 27, Issues 3-4, 2008, pp 115-121.
9. Bardis, N. G., Doukas, N. and Markovskyi, O. (2010) âTwo Level Efficient User Authentication Schemeâ. Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technology 12-15 April 2010, Knowledge Village, Dubai, UAE.
10. Guillou L.C., Quisquater J.-J. âA Paradoxical Identity-Based Signature Schemes Resulting from Zero Knowledgeâ // Advances of Cryptology -Crypto-88. Proceeding.- Springer-Verlag.- 1990.- P. 216-231.
11. Schnorr C.P. âEfficient Signature Generation for Smart Cardsâ // Journal of Cryptology, Vol. 4, No.3.- 1991.- pp.161-174.
12. Bardis N., Doukas N. and Markovskyi O., âFast subscriber identification based on the zero knowledge principle for multimedia content distributionâ, International Journal of Multimedia Intelligence and Security 2010 - Vol. 1, No.4 pp. 363 - 377, 2010.

Determining Whether a Given Block Cipher is a Permutation of Another Given Block Cipher— a Problem in Intellectual Property (Extended Abstract)

G. V. Bard[1]

[1] *The University of Wisconsin—Stout, bardg@uwstout.edu*

Overview

Imagine that, in order to avoid patent fees or export restrictions, someone permutes the plaintext bits, ciphertext bits, or key bits of a block cipher. All security properties of the block cipher would be preserved. There are many possible such permutations (e.g. $2^{3116.32}$ for the Advanced Encryption Standard, AES). It might seem infeasible to detect this fraud, and even harder to determine the permutation matrices used.

This paper presents a method whereby this fraud could be easily detected, by means of a SAT-Solver—a standard off-the-shelf software package that solves small-to-medium sized instances of the logical satisfiability problem. Moreover, this problem is intimately connected to the "isomorphism of polynomials" problem and that connection is explored at length.

Definitions

Let $\{0,1\}^\ell$ represent the set of all binary strings of length ℓ. We will now endeavor to define block ciphers as a mathematical object and make explicit some of their more relevant properties. Formally, a block cipher is a function

$$E : K \times P \to C$$

with several important properties. The key k is chosen from K, the keyspace; the plaintext message is chosen by the sender as some $p \in P$. The ciphertext, computed by the cipher, is some $c \in C$.

For practical implementation purposes, nearly all ciphers in use during the last 40 years have had all three sets K, P, and C be the set of binary strings of some fixed length. (This enables digital circuitry to be used for the cipher.) For example, the small and outdated cipher called the Data Encryption Standard (DES) uses $\{0,1\}^{56} = K$ but $P = C = \{0,1\}^{64}$ [15, Ch. 4]. The Advanced Encryption Standard (AES) uses $P = C = \{0,1\}^{128}$ but the user can choose $\{0,1\}^{128} = K$, $\{0,1\}^{192} = K$, or $\{0,1\}^{256} = K$ [15, Ch. 5]. Beyond these trivial details, there are five crucial security requirements for any block cipher.

First, it must be fast to encrypt—i.e. one can compute $E(k, p) = c$ rapidly when $k \in K$ and $p \in P$ are known. Second, it must be fast to decrypt—i.e. one can find p such that $E(k, p) = c$ when $k \in K$ and $c \in C$ are known. Third, it must be computationally infeasible to cryptanalyze—i.e. it should be infeasible to compute k such that $E(k, p) = c$ when $p \in P$ and $c \in C$ are known. There is a consensus in cryptography that "computationally infeasible" means that the fastest known algorithm should be no faster than checking all possible values of K. The set K is chosen to be extremely large—as a necessary but insufficient condition for security.

For the fourth and fifth properties, consider that for any fixed k, we can rewrite $E(k, p) = c$ as $E_k(p) = c$, and then E_k is a function $P \to C$. A property called unique decodability requires that E_k be injective for all $k \in K$. Otherwise, if there existed $k_s \in K$, $p_1 \in P$, and $p_2 \in P$ with $p_1 \neq p_2$ but $E(k_s, p_1) = E(k_s, p_2) = c_s$ then a receiver who receives the ciphertext c_s, and who is using key k_s, will have no way of determining if the sender had intended to transmit the message p_1 or the message p_2. It is universally the case that $C = P$ as sets in practice, therefore textbooks usually say that E_k must be bijective—however, that is not strictly necessary. The fifth and most important criterion is that for some $k \in K$ selected uniformly at random, it should be the case that the E_k obtained is computationally indistinguishable from a random function chosen uniformly from the set of all possible functions that map $P \to C$. Formal cryptography textbooks model block ciphers by a mathematical object called a pseudorandom permutation, for this reason.

A Note about Polynomial Time

Readers familiar with theoretical cryptography might be surprised to see that the phrase "in polynomial time" is missing above. When we say that Gaussian Elimination runs in polynomial time, we mean that if one were to take a sequence of problems, each an $n \times n$ matrix, and measure the running time (e.g. by counting the number of arithmetic operations), then one can upper-bound this running time as a function n by some polynomial of n. In this case, the polynomial is cubic.

However, when working with block ciphers, this notion of polynomial time is unavailable. Any hardware cipher has a fixed K, a fixed P, and a fixed C. There is no limit as n goes to infinity, because n is not going to infinity—it is constant.

Permutations of a Block Cipher

With some thought, one can see that none of the requirements of a block cipher will change after applying three permutations, represented by the permutation matrices M_1, M_2, and M_3, in the following sense:

$$\hat{E}(K, P) = M_3 E(M_1 K,\ M_2 P)$$

In other words, if E meets all the criteria of a block cipher, then a software pirate can fix three particular permutation matrices, and construct $\hat{E}$, which will share all of the security properties of E. However, this will not be readily apparent externally (especially if implemented in hardware, but even if implemented with software in the case where code-obfuscation tools have been used). Therefore the software pirate can evade patent fees or cipher export controls.

It is worth noting that the number of permutations is rather large. For the outdated DES cipher, there would be 56! choices for M_1, 64! choices for M_2, and 64! choices for M_3. This comes to

$$(56!)(64!)(64!) \approx 2^{840.643} \approx 10^{253.059}$$

possibilities, and the number would be considerably larger for modern ciphers like the AES. With the largest key setting permitted, there would be

$$(256!)(128!)(128!) \approx 2^{3116.32} \approx 10^{938.106}$$

One could sympathize with anyone who would assume that this ruse would be undetectable. The problem of detecting this type of fraud was posed to the author of this paper in 2009 during a coffee break at the conference CHES (Cryptographic Hardware and Embedded Systems). Regrettably, the author does not remember the name of the US Government employee who proposed this interesting problem—and the proposer did wish to remain anonymous.

This paper shows a quick and efficient method of determining, when presented with two block ciphers F and G, if one is a permutation of the other, by use of a SAT-Solver. Moreover, it explicitly computes M_1, M_2, and M_3, though indirectly.

References

1. Agrawal, M, and Saxena, N.: "Equivalence of F-algebras and Cubic Forms." In: B. Durand and W. Thomas (Eds.): *Proc: Symposium on Theoretical Aspects of Computer Science (STACS'06), Lecture Notes in Computer Science*, **Vol. 3884**, Pp 115–125. Springer. (2006).
2. Bard, G.: *Algebraic Cryptanalysis*. Springer-Verlag. (2009).
3. Berthomieu, J., Faugère, J.-C., Perret L.: "Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case." *Journal of Complexity*. **Vol. 31**. Pp. 590–616. (2015).
4. Bard, G., Courtois, N., Jefferson, C.: "Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via SAT-Solvers." Preprint. Cryptology ePrint Archive, Report 2007/024 (2006).
5. Faugère, J.-C., and Perret, L.: "Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects." In: S. Vaudenay (Ed.): *Advances in Cryptology—Proc. of EUROCRYPT, Lecture Notes in Computer Science*, **Vol. 4004**, pp. 30–47. Springer-Verlag (2006).
6. Geiselmann, W., Meier, W., and Steinwandt, R.: "An attack on the isomorphisms of polynomials problem with one secret". *International Journal of Information Security*. **Vol. 2**, No. 1, (2003).
7. Goldreich, O., Micali, S., and Wigderson, A.: "Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems." *Journal of the Association of Computing Machinery*, **Vol. 38**, No. 3, Pp. 691–729, (1991).

8. N. Kayal.: “Efficient Algorithms for Some Special Cases of the Polynomial Equivalence Problem.” In: Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1409–1421. SIAM. (2011).
9. Matsumoto, T., and Imai, H.: “Public quadratic polynomial-tuples for efficient signature-verification and message-encryption,” In: *Advances in Cryptology (EUROCRYPT’88), Lecture Notes in Computer Science*, **Vol. 330**, Springer-Verlag, Pp. 419–453, (1988).
10. Patarin, J.: “Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms.” In: N. Koblitz (ed.) *Advances in Cryptology—Proc. of EUROCRYPT, Lecture Notes in Computer Science*, **Vol. 1070**, pp. 33–48. Springer-Verlag (1996).
11. Patarin, J., Goubin, L., and Courtois, N.: “Improved Algorithms for Isomorphisms of Polynomials.” In: K. Nyberg (Ed.): *Advances in Cryptology—Proc. of EUROCRYPT, Lecture Notes in Computer Science*, **Vol. 1403**, pp. 184–200. Springer-Verlag (1998).
12. Perret, L.: “A fast cryptanalysis of the isomorphism of polynomials with one secret problem.” In: R. Cramer (Ed.): *Advances in Cryptology—Proc. of EUROCRYPT, Lecture Notes in Computer Science*, **Vol. 3495**, pp. 354–370. Springer-Verlag (2005).
13. Plût, J., Fouque, P.-A., and Macario-Rat, G.: “Solving the ‘Isomorphism of Polynomials with Two Secrets’ Problem for All Pairs of Quadratic Forms.” Preprint. (2017).
14. Saxena, N.: “Morphisms of Rings and Applications to Complexity (PhD Thesis).” Indian Institute of Technology, Kanpur, 2006.
15. Trappe, W., and Washington, L.: *Introduction to Cryptography with Coding Theory*. 2nd Edition. Pearson, Prentice-Hall. 2006.
16. Tang, S., and Xu, L.: “Proxy signature scheme based on isomorphisms of polynomials,” In: L. Xu, E. Bertino, Y. Mu (Eds.), In: *Lecture Notes in Computer Science*, **vol. 7645**, Springer, Pp. 113–125. (2012).
17. Tang, S., and Xu, L.: “Towards provably secure proxy signature scheme based on isomorphisms of polynomials.” *Future Generation Computer Systems* **Vol. 30**, Elsevier, Pp 91–97. (2014).
18. Yang, G., Tang, S., and Yang, L.: “A Novel Group Signature Scheme based on MPKC.” In: Bao, F., and Weng, J. (Eds.), *Information Security Practice and Experience (ISPEC’11), Lecture Notes in Computer Science*, **Vol. 6672**, Springer, Pp 181–195. (2011).
19. Wolf, C., and Preneel, B.: “Equivalent keys in multivariate quadratic public key systems,” *Journal of Mathematical Cryptology*, **Vol. 4**, No. 4, Pp. 375–415, (2011).

Track 3: Computer Algebra

Chairs: Rafael Sendra (Spain), Franz Winkler (Austria)

Invited Speaker: Michael Wibmer

Computing Difference Algebraic Relations Among Solutions of Linear Differential Equations

M. Wibmer[1]

[1] *University of Pennsylvania, USA, wibmer@math.upenn.edu*

Abstract

Understanding the algebraic relations among the solutions of a linear differential equation is a classical and important problem. For example,

$$\cos^2(x)+\sin^2(x)=1 \tag{1}$$

is such a relation. Often the solutions satisfy interesting relations that are not simply algebraic relations. For example,

$$\cos(2x)=2\cos^2(x)-1, \tag{2}$$

or

$$xJ_{\alpha+2}(x)-2(\alpha+1)J_{\alpha+1}(x)+xJ_\alpha(x)=0, \tag{3}$$

where $J_\alpha(x)$ is a solution of Bessel's differential equation

$$x^2y''+xy'+(x^2-\alpha^2)y=0. \tag{4}$$

These are examples of *difference algebraic relations*. The parameterized Picard-Vessiot theory ([1], [2], [3], [4]) provides a comprehensive approach to the study of these relations by associating a Galois group to the linear differential equation. These Galois groups are defined by algebraic difference equations.

We will provide a computational perspective on how to find these groups and the corresponding difference algebraic relations.

References

1. Ph. Cassidy and M. Singer, *Galois theory of parameterized differential equations and linear differential algebraic groups*, IRMA Lect. Math. Theor. Phys., Vol. 9, pp. 113-155 (2007).

2. Ch. Hardouin and M. Singer, *Differential Galois theory of linear difference equations*, Math. Ann. **342**, 2, pp. 333-377 (2008)
3. L. Di Vizio, Ch. Hardouin and M. Wibmer, *Difference Galois theory of linear differential equations*, Adv. Math. **260**, pp. 1-58, 2014
4. L. Di Vizio, Ch. Hardouin and M. Wibmer, *Difference algebraic relations among solutions of linear differential equations*, J. Inst. Math. Jussieu **16**, 1, pp. 59-119 (2017)

Interpolation of syzygies for implicit matrix representations

Ioannis Z. Emiris[1,2], Konstantinos Gavriil[1,3], and Christos Konaxis[1,2]

{emiris, ckonaxis, kgavr} @ di.uoa.gr
[1]*Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece,*
[2]*ATHENA Research Innovation Center, Maroussi, Greece*
[3]*Technische Universität Wien, Austria*

Abstract

We examine matrix representations of curves and surfaces based on syzygies and constructed by interpolation through points. They are implicit representations of objects given as point clouds. The corresponding theory, including moving lines, curves and surfaces, has been developed for parametric models. Our contribution is to show how to compute the required syzygies by interpolation, when the geometric object is given by a point cloud whose sampling satisfies mild assumptions. We focus on planar and space curves, where the theory of syzygies allows us to design an exact algorithm yielding the optimal implicit expression. The method extends readily to surfaces without base points defined over triangular patches. Our Maple implementation has served to produce the examples in this paper and is available upon demand by the authors.

Introduction

Today, one of the predominant algebraic approaches to change of representation of geometric objects and, in particular, implicitization, is based on algebraic syzygies. This is a very rich theory, which still motivates a large volume of fundamental work. This paper examines the question of interpolating the syzygies of an unknown, rational parameterization, under the assumption that the given point sample is given along with the parameter value that generated each point. Our algorithms can recover the corresponding syzygies and, hence, an implicit matrix representation of the sampled object. Moreover, the computed information is sufficient to reconstruct the entire parameterization if needed.

Another major trend today in CAD, albeit at the engineering level, is the availability and manipulation of point clouds, essentially as a means of representing the geometric object. This work wishes to capitalize on this trend. Therefore, besides its theoretical interest, our work is motivated by the following practical scenarios. The first relies on a strong hypothesis: the point sample is given along with parameter values, assuming the underlying parameterization is rational but unknown. Our method recovers the implicit representation and may also compute the exact

syzygies. Another, probably more realistic scenario, is that the sampling is obtained from an arc-length parameterization by a scanner capable of measuring the distance it has covered when moving on the curve. In practice, this assumption can be satisfied when the scanner is equipped with a GPS system. When given a set of sample points that is dense enough, the distances between consecutive points can be used to approximate the arc length of the curve. In this case we obtain syzygies corresponding to a rational parameterization that approximately interpolates the given points. The quantification of this approximation goes beyond the realm of this work, and requires some numerical analysis. Another real life example that motivates us comes from the simulation of subtractive manufacturing processes where the computation of swept volumes generated by a cutter that moves along a specified trajectory (tool path). Methods based on surface reconstruction from point clouds use a discretization of the tool's path at certain steps at which the initial point cloud representation of the tool is copied. This yields a suitable input for our method since in this case the time parameter at each step is known. Lastly, one may consider a scenario where the sample is collected by some rational parameterization, such as PH or chord parameterization, in which case our method would compute the exact syzygies and, if needed, the parametric expressions as well.

A matrix representation of an implicit object is a single matrix, generically of full rank, which represents the object in the sense that its rank drops precisely when evaluated at a point lying on the object. Matrix representations are quite robust, since they do not require computation of the implicit equation; instead, they reduce geometric operations on the object to linear algebra. In general, existing approaches to implicitization include Gröbner bases, resultants, moving lines/curves and surfaces, μ-bases and approximation complexes, as well as a number of interpolation techniques. Today, moving lines/curves and surfaces, and μ-bases seem to offer very competitive methods since they provide the veracity of algebraic approaches without the high complexity of Gröbner bases nor the problems due to base points when using resultants. Moving curves and surfaces have been used to construct matrix representations of implicit objects, and this is the premise of our work.

The theory of syzygies, including moving lines, curves and surfaces, has been developed for parametric models; it is sketched in the subsequent sections. Our contribution is to show how to compute the required syzygies by interpolation, when the input curve or surface is given by a point cloud whose sampling satisfies mild assumptions. No information on the parametric representation of the object is given, but the parametric expressions could be obtained from the algorithm's output. However, our goal is a robust implicit matrix representation, and we focus on matrices constructed only by linear syzygies. We illustrate our algorithms for planar and space curves as well as triangular surfaces, all without base points. Our Maple implementation is available upon demand by the authors.

Let us describe the input in the case of curves in an ambient space of arbitrary dimension $n \geq 2$; it shall be generalized in the sequel to surfaces. We assume that the curve admits some (unknown) affine rational parameterization $\phi : \mathbb{R} \to \mathbb{R}^n$; planar and space curves correspond to $n = 2$ and $n = 3$, respectively. The input is a *parametric set of points*. This pointset is defined as a sequence of vectors $(\tau_k; X_k)$

such that

$$\phi(\tau_k) = X_k, \text{ for all } k = 1,2,\ldots, \text{ where } \tau_k \in \mathbb{R},$$

and $X_k \in \mathbb{R}^n$. In particular, $X_k \in \mathbb{R}^2$ or $X_k \in \mathbb{R}^3$, depending on whether we study planar or space curves.

A related model for point clouds is considered in [FS05].

This paper is organized as follows: Section 1 overviews previous work, whereas Section 1 contains some background in the theory of syzygies, and develops general tools required in the sequel. Section 1 describes our method for interpolating syzygies when the input is given as a set of parametric points defining a planar or space curve. In Section 1 we extend the method to the case of triangular surfaces, given as a parametric pointset. We conclude with future work and open questions.

1 Previous work

This section discusses the main existing approaches to implicitization, with an emphasis on methods constructing matrix representations of implicit objects. Besides these methods, Gröbner bases offer a powerful and complete approach but suffer from high complexity and numerical instability.

Resultants, and their matrix formulae, have been used to express the implicit surface equation, e.g., in [MC92], under the assumption of no base points.

The most direct method to reduce implicitization to linear algebra is to construct a square matrix M, indexed by all possible monomials in the implicit equation (columns) and different values (rows) at which all monomials get evaluated. Then the vector of coefficients of the implicit equation is in the kernel of M. This idea has been extensively used, e.g. in [Dok01, EKKL13, EKK15, SY08]. The method, as introduced in [EKKL13, EKK15], exploits sparse resultant theory so as to predict the monomials in the implicit equation and thus build the interpolation matrix. It handles objects with base points.

A modern method for representing implicit equations by matrices was introduced by Sederberg and his coauthors when they rediscovered the theory of syzygies in the context of computer science [SSQK94, SC95, SGD97]. Let us take the example of planar curves without base points, parameterized by the homogeneous polynomials $(f_1(s:t) : f_2(s:t) : f_3(s:t))$, all of same degree d. The main idea is to define a *moving line* in $\mathbb{P}^2$ as

$$h_1(s:t)x + h_2(s:t)y + h_3(s:t)z = 0, \tag{5}$$

where x, y, z are homogeneous coordinates in $\mathbb{P}^2$ and $h_i(s:t) \in \mathbb{C}[s,t]$, $i = 1,2,3$, are homogeneous polynomials of same degree. The moving line follows the curve if

$$\sum_{i=1}^{3} h_i(s:t) f_i(s:t) = 0, \text{ for all } (s:t) \in \mathbb{P}^1. \tag{6}$$

Algebraically, the triplet (h_1, h_2, h_3) of homogeneous polynomials h_i, or, equivalently, the moving line (5), is a (linear) *syzygy* on the polynomials f_i. It is known, see e.g. [SSQK94, Cox01], that there are d independent moving lines of degree $d-1$ that follow the curve. Using these moving lines it is possible to construct a $d \times d$ matrix whose determinant is a multiple of the implicit equation, see Proposition 1.

In the next section we provide a comprehensive discussion on syzygies. For now, let us recall that in the case of surfaces without base points, one may also construct a square matrix whose determinant is a power of the implicit polynomial [CGZ00], by using d moving planes and $(d^2-d)/2$ moving quadrics, all of degree $d-1$, see Subsection 1.

If we allow orthogonal matrices, it suffices to work with linear syzygies, and this is the main approach adopted in this work. In general, one defines the notion of *critical degree* ν_0, see Proposition 1, which corresponds to the degree of the linear syzygies required to define an orthogonal matrix $\mathbb{M}_\nu(\phi)$ that satisfies the following property [BLB10]: for any point $p \in \mathbb{P}^2$ in the case of planar curves or, respectively, $p \in \mathbb{P}^3$ in the case of space curves or surfaces, the rank of $\mathbb{M}_\nu(\phi)$ evaluated at p drops if and only if p belongs to the algebraic closure of (ϕ). The critical degree is, in general, at least as large as the regularity of the map sending tuples of polynomials to combinations generalizing those in expression (6). In particular, the critical degree in the case of planar and space curves without base points is $d-1$, and for triangular surfaces it is $2(d-1)$.

The matrices indirectly represent implicit objects and allow for geometric operations, such as surface-surface intersection [BLB12] and, more recently, ray shooting [SBAD16], to be reduced to linear algebra. Their advantage is that the matrices are much smaller than interpolation matrices, and allow for inversion by an eigenproblem on these matrices. They also simplify in the presence of base points while other methods become more complicated. On the other hand, their construction is a two-step process of matrix operations. Moreover, they are symbolic with entries linear polynomials in the implicit variables.

2 Basic tools

This section uses known results in the theory of syzygies to develop certain tools needed for stating our algorithms in subsequent sections. In particular we shall relate the degree of a given grading of the syzygy module to its dimension. For a comprehensive survey on the subject, we refer the interested reader to [Cox01, Cox03].

2.1 Planar curves

Consider the (homogeneous) parameterization of a planar curve $\mathscr{C}$:

$$\phi : \mathbb{P}^1 \to \mathbb{P}^2 : \ (s:t) \mapsto (f_1(s:t) : f_2(s:t) : f_3(s:t)), \tag{7}$$

where $f_i \in \mathbb{C}[s,t]$ are homogeneous of the same degree d, and assume that $\gcd(f_1,f_2,f_3) = 1$, i.e. ϕ has no base points.

Consider a syzygy (h_1,h_2,h_3), where $h_i(s:t) \in \mathbb{C}[s,t],\ i = 1,2,3$, are homogeneous polynomials of same degree, as in (6):

$$\sum_{i=1}^{3} h_i(s:t) f_i(s:t) = 0, \text{ for all } (s:t) \in \mathbb{P}^1,$$

This is a linear syzygy on the polynomials f_i. The common degree of the h_1,h_2,h_3 is known as the degree of this syzygy. The set of all syzygies is denoted by $Syz(f_1,f_2,f_3)$, and has the structure of a graded module. By fixing a degree $\nu \geq 0$, we can consider the set of syzygies of degree ν, denoted by $Syz(f_1,f_2,f_3)_\nu$, which is known to be a finite dimensional $\mathbb{C}$-vector space. One can compute a basis $L_1,\ldots,L_{N_\nu}$ of this vector space by solving a linear system, where N_ν denotes the basis cardinality.

We identify each $L_j = (h_1^{(j)},h_2^{(j)},h_3^{(j)})$ with its moving line and we develop it in terms of the s,t as follows:

$$L_j := \sum_{k=1}^{3} h_k^{(j)} x_k = \sum_{i=0}^{\nu} \Lambda_{i,j}(x,y,z) s^i t^{\nu-i}, \quad j = 1,\ldots,N_\nu, \tag{8}$$

where $\Lambda_{i,j}(x,y,z)$ is a linear polynomial in $\mathbb{C}[x,y,z]$. Let $\mathbb{M}_\nu(\phi)$ be the $(\nu+1) \times N_\nu$ matrix, whose jth column contains the coefficients $\Lambda_{i,j}(x,y,z)$ of L_j in (8).

A fundamental result here is the following, showing that there are d independent moving lines of degree $d-1$ that follow ϕ. Then, $\mathbb{M}_\nu(\phi)$ is a square $d \times d$ implicitization matrix, for $\nu = d-1$.

[CLO05, Sec.6.4],[Cox01, Thm.2.2] When the plane curve has no base points and with the notation above, $N_{d-1} = d$ and $\det(\mathbb{M}_{d-1}(\phi)) = c \cdot F^{\deg(\phi)}$, where $c \in \mathbb{C}^*$, F is the implicit polynomial of the curve $\mathscr{C}$, and $deg(\phi)$ is the number of preimages of a generic point on $\mathscr{C}$.

The entire module of syzygies $Syz(f_1,f_2,f_3)$ is a free module of rank 2. Let $P = (P_1,P_2,P_3), Q = (Q_1,Q_2,Q_3)$ be its generators of degrees $\mu_1 \leq \mu_2$, respectively. It is known that $\mu_1 + \mu_2 = d$.

The P,Q are called a μ-*basis* of $Syz(f_1,f_2,f_3)$. Thus, we write any syzygy in $Syz(f_1,f_2,f_3)_\nu$ as a polynomial combination, for homogeneous $p,q \in \mathbb{C}[s,t]$, namely:

$$pP + qQ, \text{ where } \deg(p) = \nu - \mu_1 \text{ and } \deg(q) = \nu - \mu_2, . \tag{9}$$

If we identify P,Q with their moving lines, i.e., $P = P_1x + P_2y + P_3z,\ Q = Q_1x + Q_2y + Q_3z$, then the Sylvester resultant of P,Q gives the implicit equation F of $\mathscr{C}$:

$$\mathrm{Res}(P,Q) = c \cdot F^{\deg(\phi)},$$

where $c, F, \deg(\phi)$ are as in Proposition 1.

We now employ (9) to compute a basis of $Syz(f_1,f_2,f_3)_\nu$ and its dimension, as ν varies. Recall d is the homogeneous degree of f_1,f_2,f_3. The following lemma essentially appears in [CSC98, Cor.2,p. 811]. For the convenience of the reader we include a self-contained and simple proof.

Lemma 1. *We distinguish the following cases for the degree ν of $Syz(f_1,f_2,f_3)_\nu$:*

(a) $\nu \le \mu_1 - 1$. *Then* $\dim Syz(f_1,f_2,f_3)_\nu = 0$.
(b) $\mu_1 - 1 \le \nu \le \mu_2 - 1$. *Then* $\dim Syz(f_1,f_2,f_3)_\nu = \nu - \mu_1 + 1$.
(c) $\nu \ge \mu_2 - 1$. *Then* $\dim Syz(f_1,f_2,f_3)_\nu = 2\nu - d + 2$.

Note that the intervals of the three cases share their endpoints, hence the overall piecewise linear curve is continuous. The lemma generalizes the fundamental result that $\dim Syz(f_1,f_2,f_3)_{d-1} = d$, from Proposition 1.

Proof. From equation (9) we get the basis of $Syz(f_1,f_2,f_3)_\nu$, for general ν:

$$\mathscr{B} = \{s^i t^{\nu-\mu_1-i} P \mid 0 \le i \le \nu - \mu_1\} \cup \{s^i t^{\nu-\mu_2-i} Q \mid 0 \le i \le \nu - \mu_2\}. \qquad (10)$$

Then, the lemma follows straightforwardly by computing the cardinality of $\mathscr{B}$ for each case (a)-(c). In particular we have:

(a) If $\nu \le \mu_1 - 1$, then $\mathscr{B} = \emptyset$ because any non-trivial polynomial combination of P, Q has total degree $\ge \mu_1$. Hence $\dim Syz(f_1,f_2,f_3)_\nu = 0$.
(b) If $\mu_1 \le \nu \le \mu_2 - 1$, then $\mathscr{B} = \{s^i t^{\nu-\mu_1-i} P \mid 0 \le i \le \nu - \mu_1\}$ and $|\mathscr{B}| = \dim Syz(f_1,f_2,f_3)_\nu = \nu - \mu_1 + 1$. If $\nu = \mu - 1$ this formula yields correctly 0.
(c) If $\nu \ge \mu_2$, then $\mathscr{B}$ is as in (10), containing both multiples of P and Q, hence $|\mathscr{B}| = \dim Syz(f_1,f_2,f_3)_\nu = 2\nu - d + 2$, since $d = \mu_1 + \mu_2$. At $\nu = \mu_2 - 1$, the formula yields $\mu_2 - \mu_1$, which is also obtained at this point by the formula of case (b).

The lemma is summarized in Figure 1.

Fig. 1 The graph of the dimension N_ν of $Syz(f_1,f_2,f_3)_\nu$ with respect to ν. The dashed red line intersects the graph at point $(d-1, d)$ corresponding to the critical degree.

2.2 Space curves

Consider the space curve parameterized homogeneously as

$$\phi : \mathbb{P}^1 \to \mathbb{P}^3 : \quad (s:t) \to (f_1(s:t) : f_2(s:t) : f_3(s:t) : f_4(s:t)), \qquad (11)$$

where d is again defined as the homogeneous degree of the polynomials $f_i(s,t)$, $i = 1,\ldots,4$. Suppose $\gcd(f_1,f_2,f_3,f_4) = 1$, i.e. there are no base points.

The module of syzygies $Syz(f_1,f_2,f_3,f_4)$ is a free module of rank 3. Let $P=(P_1,P_2,P_3,P_4)$, $Q=(Q_1,Q_2,Q_3,Q_4)$, $R=(R_1,R_2,R_3,R_4)$ be its generators and $\mu_1 \leq \mu_2 \leq \mu_3$ be their degrees respectively. It is known that $\mu_1+\mu_2+\mu_3=d$.

The P,Q,R are called a *μ-basis* of $Syz(f_1,f_2,f_3,f_4)$. We can write any syzygy in $Syz(f_1,f_2,f_3,f_4)_\nu$ as a polynomial combination for homogeneous polynomials $p,q,r \in \mathbb{C}[s,t]$:

$$pP+qQ+rR, \text{ where } \deg(p)=\nu-\mu_1, \deg(q)=\nu-\mu_2, \deg(r)=\nu-\mu_3. \quad (12)$$

Identifying P,Q,R with their moving lines, i.e., $P=P_1x+P_2y+P_3z+P_4w$, $Q=Q_1x+Q_2y+Q_3z+Q_4w$, $R=R_1x+R_2y+R_3z+R_4w$, and forming the Sylvester resultant of every pair of P,Q,R gives one implicit equation of a surface containing curve $\mathscr{C}$; the latter is thus defined set-theoretically as the intersection of 3 surfaces.

We can now relate the dimension of $Syz(f_1,f_2,f_3,f_4)_\nu$ to ν.

Lemma 2. *We distinguish the following cases for the degree ν of $Syz(f_1,f_2,f_3,f_4)_\nu$:*

(a) $\nu \leq \mu_1-1$. *Then* $\dim Syz(f_1,f_2,f_3,f_4)_\nu = 0$.
(b) $\mu_1-1 \leq \nu \leq \mu_2-1$. *Then* $\dim Syz(f_1,f_2,f_3,f_4)_\nu = \nu-\mu_1+1$.
(c) $\mu_2-1 \leq \nu \leq \mu_3-1$. *Then* $\dim Syz(f_1,f_2,f_3,f_4)_\nu = 2\nu-\mu_1-\mu_2+2$.
(d) $\mu_3-1 \leq \nu$. *Then* $\dim Syz(f_1,f_2,f_3,f_4)_\nu = 3\nu-d+3$.

The intervals of subsequent cases share their endpoints, hence the overall piecewise linear curve is continuous.

Proof. From equation (12) we get the basis of $Syz(f_1,f_2,f_3,f_4)_\nu$ for general ν:

$$\mathscr{B}=\{s^i t^{\nu-\mu_1-i}P \mid 0\leq i\leq \nu-\mu_1\} \cup \{s^i t^{\nu-\mu_2-i}Q \mid 0\leq i\leq \nu-\mu_2\} \cup \{s^i t^{\nu-\mu_3-i}W \mid 0\leq i\leq \nu-\mu_3\}. \quad (13)$$

Then, the lemma follows straightforwardly by computing the cardinality of $\mathscr{B}$ for each case (a)-(d). In particular we have:

(a) If $\nu \leq \mu_1-1$, then $\mathscr{B}=\emptyset$ because any non-trivial polynomial combination of P,Q,W has total degree $\geq \mu_1$. Hence $\dim Syz(f_1,f_2,f_3,f_4)_\nu = 0$.
(b) If $\mu_1 \leq \nu \leq \mu_2-1$, then $\mathscr{B}=\{s^i t^{\nu-\mu_1-i}P \mid 0\leq i\leq \nu-\mu_1\}$ and $|\mathscr{B}|=\dim Syz(f_1,f_2,f_3,f_4)_\nu = \nu-\mu_1+1$. If $\nu=\mu_1-1$ the formula yields correctly 0.
(c) If $\mu_2 \leq \nu \leq \mu_3-1$, then

$$\mathscr{B}=\{s^i t^{\nu-\mu_1-i}P \mid 0\leq i\leq \nu-\mu_1\} \cup \{s^i t^{\nu-\mu_2-i}Q \mid 0\leq i\leq \nu-\mu_2\}$$

and $|\mathscr{B}|=\dim Syz(f_1,f_2,f_3,f_4)_\nu = 2\nu-\mu_1-\mu_2+2$. If $\nu=\mu_2-1$ the formula yields $\mu_2-\mu_1$, which is also obtained from the formula of case (b).
(d) If $\nu \geq \mu_3$, then $\mathscr{B}$ is as in (13) and $|\mathscr{B}|=\dim Syz(f_1,f_2,f_3,f_4)_\nu = 3\nu-d+3$. If $\nu=\mu_3-1$, the formula yields $2\mu_3-\mu_1-\mu_2$, which agrees with the value of the formula in case (c) at this point.

Figure 2 summarizes the lemma.

Fig. 2 The graph of the dimension N_ν of $Syz(f_1,f_2,f_3,f_4)_\nu$ with respect to ν. The dashed red line intersects the graph at point $(d-1,2d)$ corresponding to critical degree $\nu_0 = d-1$.

2.3 General Curves

We may unify and generalize the previous discussion by considering curves in $\mathbb{P}^n$, for any ambient dimension $n \geq 2$, parameterized homogeneously as

$$\phi : \mathbb{P}^1 \to \mathbb{P}^n : \quad (s:t) \to (f_1(s:t) : \ldots : f_n(s:t)), \tag{14}$$

where d is the homogeneous degree of the polynomials $f_i(s,t)$, $i = 1,\ldots,n$. By the Hilbert Syzygy Theorem, the syzygy module $Syz(f_1,\ldots,f_n)$ is free of rank n, and in particular it has a μ-basis [CSC98, Thm.1]. Let $\mu_1 \leq \cdots \leq \mu_n$, be the degrees of the polynomials in the μ-basis of the module. Hence, the previous discussion extends to this case as well.

The derived formulae for N_ν in the two Lemmas above can be unified and generalized into a piecewise linear formula with n nontrivial pieces, where the equation of the k-th segment is

$$N_\nu = \sum_{i=1}^{k} (\nu - \mu_i + 1), \quad \text{for } \mu_k - 1 \leq \nu \leq \mu_{k+1} - 1,\ k = 1,2,\ldots,n-1, \text{ or } \mu_n - 1 \leq \nu.$$

Of course $N_\nu = 0$ for $\nu \leq \mu_1 - 1$.

2.4 Triangular surfaces

The theory of moving lines generalizes to surfaces in $\mathbb{P}^3$. Let us focus on the case of surfaces without base points, parameterized by

$$\phi : \mathbb{P}^2 \to \mathbb{P}^3 : \ (s:t:u) \mapsto (f_1(s:t:u) : f_2(s:t:u) : f_3(s:t:u) : f_4(s:t:u)), \tag{15}$$

where all f_i are homogeneous of degree d. These are known as triangular surfaces.

The analogue of a moving line of degree ν in $\mathbb{P}^3$ is a moving plane:

$$h_1(s:t:u)x + h_2(s:t:u)y + h_3(s:t:u)z + h_4(s:t:u)w = 0, \tag{16}$$

where $\deg(h_i) = \nu$, $i = 1,\ldots,4$.

A moving quadric of degree ν is defined as:

$$h_1x^2 + h_2y^2 + h_3z^2 + h_4w^2 + h_5xy + h_6xz + h_7xw + h_8yz + h_9yw + h_{10}zw = 0, \tag{17}$$

where where $\deg(h_i(s:t:u)) = \nu$, $i = 1,\ldots,7$.

There are d linearly independent moving planes of degree $d-1$ that follow the surface (15). Moreover, there are $(d^2-d)/2$ linearly independent moving quadrics of degree $d-1$ that follow the surface and are not obtained from some of the d moving planes by multiplication by x,y,z,w. The determinant of the $(d^2+d)/2\times(d^2+d)/2$ matrix $\mathbb{M}_{d-1}(\phi)$, constructed as in the planar curve case by using the corresponding syzygies, is a power of the implicit polynomial of the surface [CGZ00].

There exist extensions of the method above when the surface has finitely many base points that satisfy certain assumptions, see [BCD03].

Turning our attention to the whole syzygy module $Syz(f_1,f_2,f_3,f_4)$ of the surface (15), it is not always free, and it is certainly not free when there are no base points. However, if we dehomogenize (15), then the syzygy module is free, of rank 3. Contrary to the case of curves, the elements of the μ-basis are not the syzygies of lowest degree nor they are unique. Moreover we do not have bounds on the degree of the generators. Let μ_3 be the maximum degree. In the case $\nu\geq\mu_3$, we shall be able to relate ν to the dimension N_ν of $Syz(f_1,f_2,f_3,f_4)_\nu$.

Let us consider the following map for certain gradings of homogeneous polynomial ring $\mathbb{C}[s,t,u]$:

$$(\mathbb{C}[s,t,u])^4_\nu\to\mathbb{C}[s,t,u]_{\nu+d}:(h_1,h_2,h_3,h_4)\mapsto\sum_{i=1}^{4}h_if_i.$$

Its kernel is precisely $Syz(f_1,f_2,f_3,f_4)_\nu$. For $\nu\geq\nu_0=2(d-1)$, the map is of full rank. Actually, it may be of full rank for lower ν but in constructing implicitization matrices, we are interested in the critical degree. Given a map of full rank, we compute $\dim Syz(f_1,f_2,f_3,f_4)_\nu$ as the map's nullity:

$$N_\nu=4\binom{\nu+2}{2}-\binom{\nu+d+2}{2},$$

which is clearly always an integer. This establishes the following.

Lemma 3. *For the degree ν of $Syz(f_1,f_2,f_3,f_4)_\nu$, $\nu\geq2(d-1)$ implies*

$$\dim Syz(f_1,f_2,f_3,f_4)_\nu=\frac{3\nu^2}{2}-\nu(d-\frac{9}{2})-\frac{d(d+3)}{2}+3.$$

2.5 Orthogonal matrix representations

If we allow for orthogonal matrices and assume that the base points are *local complete intersections*, then we can restrict ourselves to linear syzygies and construct a matrix $\mathbb{M}_\nu(\phi)$ expressing these syzygies, for which the following holds:

[BLB10, Bus14] Let us define the following *critical degrees*: $\nu_0=d-1$ for planar and space curves, and $\nu_0=2(d-1)$ for triangular surfaces. Then, for all $\nu\geq\nu_0$,

matrix $\mathbb{M}_\nu(\phi)$ constructed by the respective linear syzygies satisfies the following property: for any point $p \in \mathbb{P}^2$ in the case of planar curves, or $p \in \mathbb{P}^3$ otherwise, the rank of $\mathbb{M}_\nu(\phi)$ evaluated at p drops if and only if p belongs to the algebraic closure of (ϕ).

We may dehomogenize and obtain the equivalent property, that a point $(X,Y) \in \mathbb{C}^2$ belongs to $\mathscr{C}$ if and only if the rank of $\mathbb{M}_\nu(X,Y)$ drops; the latter denotes the matrix in the non-homogeneous setting.

3 Syzygies of curves

This section describes how to interpolate the basis of the graded syzygy module of a given degree ν for the case of planar curves, space curves and triangular surfaces. These syzygies can be used to build a matrix as already described.

3.1 Planar curves

We describe the method for computing a basis of the linear syzygy module of degree ν of a rational planar curve given by the (unknown) parameterization (7). The dehomogenization of ϕ gives the rational planar curve $\mathscr{C}$ parameterized by

$$\phi : \mathbb{C}^1 \to \mathbb{C}^2 : t \to \left(X(t) = \frac{f_1(t)}{f_3(t)}, Y(t) = \frac{f_2(t)}{f_3(t)} \right), \tag{18}$$

where ϕ is not known. The input is a set of triplets of the form

$$(\tau_1; X_1, Y_1), (\tau_2; X_2, Y_2), \ldots$$

such that $\phi(\tau_k) = (X_k, Y_k)$, for a range of $k \geq 1$ to be defined below. These triplets are assumed sufficiently generic, in particular they may be sampled following the scenarios described in Section , e.g. when ϕ is an arc-length parameterization and the triplets are sampled by a scanner following $\mathscr{C}$.

Our goal is to design an algorithm for computing an implicit matrix representation of the curve $\mathscr{C}$, described by this parametric set of points. The algorithm shall try different degrees $\nu \geq 0$, and shall compute a $\mathbb{C}$-basis for $Syz(X,Y,1)_\nu$: since the rational functions of $X(t), Y(t)$ are not explicitly known, we compute the basis in the following manner. Consider the moving line $h_1 X + h_2 Y + h_3 = 0$. The expanded form of each h_i is

$$h_i = \sum_{\delta=0}^{\nu} h_{i,\delta} t^\delta \in \mathbb{C}[t],\ i = 1,2,3, \tag{19}$$

where the $h_{i,\delta}$ are (unknown) coefficients. Hence, we can rewrite the moving line as

$$\sum_{\delta=0}^{\nu} t^{\delta} X h_{1,\delta} + \sum_{\delta=0}^{\nu} t^{\delta} Y h_{2,\delta} + \sum_{\delta=0}^{\nu} t^{\delta} h_{3,\delta} = 0. \tag{20}$$

Such equations are going to be used to determine the $3(\nu+1)$ unknown coefficients $h_{i,\delta}$ by interpolation at the sampled triplets. For this, we define a $3(\nu+1)\times 3(\nu+1)$ matrix H whose rows are indexed by evaluations $t=\tau_k$, for $k=1,\ldots,3(\nu+1)$, and each row expresses equation (20) as follows:

$$[X_k, \tau_k X_k, \ldots, \tau_k^{\nu} X_k, Y_k, \tau_k Y_k, \ldots, \tau_k^{\nu} Y_k, 1, \tau_k, \ldots, \tau_k^{\nu}].$$

Clearly, the vector of coefficients $[h_{1,0}, h_{1,1}, \ldots, h_{3,\nu}]$ corresponding to any element of $Syz(X,Y,1)_\nu$ lies in the kernel of matrix H.

We compute a basis of the kernel of matrix H and rewrite the j-th kernel basis vector

$$(h_{1,0}^{(j)}, \ldots, h_{1,\nu}^{(j)}, h_{2,0}^{(j)}, \ldots, h_{2,\nu}^{(j)}, h_{3,0}^{(j)}, \ldots, h_{3,\nu}^{(j)})$$

as $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)})$ following equation (19).

Let h be the kernel dimension of matrix H. We can see that $h = N_\nu$ under the genericity assumption on the given triplets and the matrix H, because the kernel basis of H corresponds to a $\mathbb{C}$-basis of $Syz(f_1,f_2,f_3)_\nu$.

Moreover, since every vector of coefficients of a syzygy lies in the kernel, the vector-space basis of the kernel is a vector-space basis of the syzygy grade, because the latter is a vector space. Then the triplets $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)})$, $j=1,\ldots,N_\nu$ form a $\mathbb{C}$-basis of $Syz(X,Y,1)_\nu$. In the case $h \geq \nu+1$, the $\mathbb{C}$-basis of $Syz(X,Y,1)_\nu$ yields the matrix $\mathbb{M}_\nu(X,Y)$, which offers a matrix representation of the implicit curve $\mathscr{C}$, since ν verifies $\nu \geq d-1$.

Lemma 1 implies the following. The proof follows easily from the information in Figure 2.

Corollary 1. *Consider a rational parametric curve $\mathscr{C}$ of the form (18). Following the above notation, let d be the homogeneous degree of the (unknown) f_i, $i=1,2,3$, $\nu \geq 0$ be a fixed degree, specifying a syzygy grading, and $h = \dim\ker(H)$ be the cardinality of the kernel basis of H. Then,*

1. $h < \nu+1$ *if and only if* $\nu < d-1$.
2. $h = \nu+1$ *if and only if* $\nu = d-1$, *then* $h = d$.
3. $h > \nu+1$ *if and only if* $\nu > d-1$.

Corollary 1 allows us to compute d by constructing matrix H and comparing h with the selected ν. It is clear that we can also recover the parameterization, but the goal of this work is to obtain robust implicit representations of point cloud models.

Algorithmically, one starts with small ν, say $\nu=1$. While $h < \nu+1$, the algorithm doubles ν. If $h > \nu+1$, we perform binary search to identify the point where $h = \nu+1$, and $h=d$. The first phase, where ν is being doubled, goes up to about $2d$, hence needs $O(\lg d)$ steps. The binary search takes $O(\lg d)$ steps as well, hence the algorithm makes overall $O(\lg d)$ corank computations for matrices of dimension up to $2d$.

Another possible algorithm uses two values of the syzygy grading, namely $\nu' > \nu > 0$, and compute the corresponding kernel dimensions $h' > h \geq 0$. The algorithm terminates when $\nu' \geq \mu_2 - 1$, then solves $h' = 2\nu' - d + 2$ for d. The main step is to compute the slope of the segment defined by the two points, namely

$$\lambda = \frac{h' - h}{\nu' - \nu} \in [0, 2].$$

The algorithm terminates when $\lambda > 1$ because this implies $\nu' \geq \mu_2 - 1$. If $\lambda \leq 1$ the algorithm increases degree ν'. This increase happens by setting $\nu' \leftarrow \nu' + 1$ then, if $\lambda \leq 1$ again, the algorithm doubles ν'. The algorithm requires $O(\lg \mu_2)$ rank computations, which is faster than the previous one.

Rank computation, by means of Gaussian elimination or QR-decomposition, of a m-dimensional matrix has complexity $O(m^\omega)$ in the exact setting, where $\omega < 2.373$ is the exponent of matrix multiplication. Clearly, the corank computation for H can be achieved in $O(\nu^\omega)$ operations, for a given ν. In practice, this is rather of cubic complexity.

The matrices H constructed at various steps are very much related to each other, since the larger ones are obtained by adding columns and rows to a smaller matrix. The new columns and rows correspond, respectively, to higher degree monomials in equation (20) and new interpolation points $t = \tau_k$. In particular, suppose we have constructed H for some ν, hence of matrix dimension $3(\nu + 1)$, the corresponding matrix H' constructed for $\nu' > \nu$ has dimension $3(\nu' + 1)$ and the following block structure:

$$H' = \begin{bmatrix} H & H_{12} \\ H_{21} & H_{22} \end{bmatrix},$$

where $[H_{21} \,|\, H_{22}]$ corresponds to $3(\nu' - \nu)$ new rows. Suppose the new degree $\nu' = \nu + O(1)$, i.e. the two degrees do not differ significantly, and suppose the corank of H is $h = \nu - O(1)$, i.e. it is not significantly smaller than ν. Given a rank revealing decomposition of H, we apply it to the new columns, then compute the rank of H' using a total of $O(\nu'^2)$ operations. We thus achieve a speedup of up to one order of magnitude under the current assumptions.

Example 1. Consider the folium of Descartes curve *affinely* parameterized as:

$$\mathscr{C} = \left\{ \left(\frac{3t}{t^3 + 1}, \frac{3t^2}{t^3 + 1} \right) \in \mathbb{C}^2 : t \in \mathbb{C} \right\} \tag{21}$$

Notice $d = 3$ for curve $\mathscr{C}$.

Suppose we are given a sample of random points on $\mathscr{C}$ for various values of the parameter t, denoted by triplets $(\tau_k; X_k, Y_k)$, and that we use them to construct the matrix H as described above, with no knowledge of the parametric equation. We try different values of ν:

For $\nu_1 = 1$, the $\mathbb{C}$-basis of $Syz(X,Y)_1$ is $\{(-t, 1, 0)\}$, that is we are in case 1 of Corollary 1 since $N_{\nu_1} < \nu_1 + 1$. For $\nu_2 = 2$, the computed basis of $Syz(X,Y)_2$ is

$$\{(-t^2,t,0),(-t,1,0),(-1/3,-t^2/3,t)\},$$

that is, case 2 of Corollary 1. This is to be expected since we picked $\nu_2 = d-1$. Any $\nu \geq \nu_2$ is a valid choice to construct an implicit representation matrix $\mathbb{M}_\nu(X,Y)$.

For $\nu_2 = 2$, the matrix is

$$\mathbb{M}_{\nu_2}(X,Y) = \begin{bmatrix} -X & 0 & -Y/3 \\ Y & -X & 1 \\ 0 & Y & -X/3 \end{bmatrix}, \tag{22}$$

whose determinant indeed yields implicit equation $X^3+Y^3-3XY=0$.

3.2 Space curves

The method we have described extends naturally to the case of space curves. Assume an unknown parameterization in projective space:

$$\phi : (t_1 : t_2) \to (f_1(t_1 : t_2), f_2(t_1 : t_2), f_3(t_1 : t_2), f_4(t_1 : t_2)), \tag{23}$$

where d is again defined as the homogeneous degree of the polynomials $f_i(t_1,t_2)$, $i = 1,\ldots,4$. In this case, the critical degree of the syzygies needed for computing the matrix representation of Proposition 1 is $d-1$, same as for planar curves, meaning ν must be $\geq d-1$.

Again, we use moving lines, expressed as follows:

$$\sum_{\delta=0}^{\nu} t^\delta X h_{1,\delta} + \sum_{\delta=0}^{\nu} t^\delta Y h_{2,\delta} + \sum_{\delta=0}^{\nu} t^\delta Z h_{3,\delta} + \sum_{\delta=0}^{\nu} t^\delta h_{4,\delta} = 0.$$

The corresponding equations define matrix H. They contain $4(\nu+1)$ unknown coefficients $h_{i,\delta}$, hence the dimension of matrix H constructed for some chosen degree ν is $\dim(H) = 4(\nu+1)$. Let h be the corank of matrix H.

A corollary of Lemma 2 follows, which shall let us identify the critical degree $\nu_0 = d-1$, see Proposition 1. The proof is straightforward if one considers Figure 1.

Corollary 2. *Consider a rational parametric space curve $\mathscr{C} \subset \mathbb{R}^3$ of the form (23). Let d be the homogeneous degree of the (unknown) f_i, $i = 1,\ldots,4$, $\nu \geq 0$ be the degree defining the grading of the syzygy, and $h = \dim\ker(H)$, using the above notation. Then we have:*

1. *$h < 2(\nu+1)$ if and only if $\nu < d-1$.*
2. *$h = 2(\nu+1)$ if and only if $\nu = d-1$, then $h = 2d$.*
3. *$h > 2(\nu+1)$ if and only if $\nu > d-1$.*

We might apply Lemma 2 to establish a similar corollary distinguishing among 3 cases, with middle case $h = 2\nu + 1$. This would have been sufficient for computing d but not enough to build an implicitization matrix from linear syzygies.

Two algorithms are now possible, analogous to those for planar curves in order to identify d and compute the syzygies by interpolation through the parametric point set. Corollary 2 leads to a binary search technique in order to identify the critical degree $\nu_0 = d - 1$.

Alternatively, there is an algorithm using two syzygy degrees, namely $\nu' > \nu$, and computing the slope of the coranks until ν' lies in the last segment of the graph in Figure 2. For this algorithm, Lemma 2 implies the following properties for slope

$$\lambda = \frac{N'_\nu - N_\nu}{\nu' - \nu} \in [1,3].$$

First, λ takes an integer value if both ν, ν' correspond to the same segment of the polygonal line in Figure 2. Otherwise, we have the following cases:
– $\lambda \in (1,3)$ iff ν, ν' correspond to the first and third segments,
– $\lambda \in (1,2)$ iff ν, ν' correspond to the first and second segments,
– $\lambda \in (2,3)$ iff ν, ν' correspond to the second and third segments.

Example 2. Consider the Viviani window curve *affinely* parameterized as:

$$\mathscr{C} = \left\{ \left(\frac{2t - 2t^3}{(1+t^2)^2}, \frac{4t^2}{(1+t^2)^2}, \frac{1-t^4}{(1+t^2)^2} \right) \in \mathbb{C}^3 : t \in \mathbb{C} \right\} \tag{24}$$

The degree of curve $\mathscr{C}$ is $d = 4$.

We are again given a sample of random points on $\mathscr{C}$ for various values of the parameter t, denoted by quadruplets $(\tau_k; X_k, Y_k, Z_k)$, which we use them to construct the matrix H.

For $\nu_1 = 1$, the $\mathbb{C}$-basis of $Syz(X,Y,Z)_1$ is $\{(-1,-t,t,t),(t,-1,-1,1)\}$, that is we are in case 1 of Corollary 2, since $N_{\nu_1} < 2(\nu_1 + 1)$. By choosing $\nu_2 = 3$, the computed basis of $Syz(X,Y)_3$ consists of 8 elements, that is, case 2 of Corollary 2. This is to be expected since we picked $\nu_2 = d - 1$. Thus, any choice of ν such that $\nu \geq \nu_2$ is a valid choice to construct an implicit representation matrix $\mathbb{M}_\nu(X,Y,Z)$.

For $\nu_2 = 3$, the matrix is

$$\mathbb{M}_{\nu_2}(X,Y,Z) = \begin{bmatrix} Z+1 & X/2 & 0 & 0 & -X/2 & 0 & Y & 0 \\ X & 1 & X/2 & 0 & Z & -X/2 & 2X & Y \\ -Y & 3X/2 & -Y+1 & X & -X/2 & Z & -Y & 2X \\ 0 & -Y & -X/2 & -Y-Z+1 & 0 & -X/2 & 0 & -Y \end{bmatrix}. \tag{25}$$

4 Syzygies of surfaces

This section extends the applicability of our method to surfaces in $\mathbb{R}^3$ without base points.

The theory of syzygies has been fully generalized to certain types of surfaces only, namely tensor product and triangular surfaces [Cox01, sec.3-4]. In these cases, it is known how many moving planes and moving surfaces, and of which degree, one has to include in order to construct a matrix whose determinant corresponds to the implicit equation. We focus on triangular surfaces because in this case it is easier to obtain the function of the dimension N_ν of the graded syzygy module with respect to the degree ν of the syzygies. The method should extend to tensor product surfaces as well, but then N_ν is a function of the bi-degree $\nu = (\nu_1, \nu_2)$ of the parameterization.

The input is now a *parametric pointset*

$$(\tau_k, \sigma_k; X_k),\ k = 1, 2, \ldots,\ \text{where } (\tau_k, \sigma_k) \in \mathbb{R}^2,\ \text{and } X_k \in \mathbb{R}^3.$$

Let us recall triangular surfaces:

$$\phi : \mathbb{P}^2 \to \mathbb{P}^3 :\ \ (s:t:u) \mapsto (f_1(s:t:u) : f_2(s:t:u) : f_3(s:t:u) : f_4(s:t:u)),$$

where homogeneous $f_i(t_1 : t_2 : t_3)$, $i = 1, \ldots, 4$ has degree d. To construct the matrix representation, one has to include d moving planes of degree $d-1$ and $(d^2-d)/2$ moving quadrics of degree $d-1$, assuming no base points exist.

To avoid interpolating quadrics and to keep the size of the interpolation matrices low, we shall interpolate only linear syzygies and aim at the critical degree $\nu_0 = 2(d-1)$ which, by Proposition 1, allows for constructing an implicit matrix representation by employing only linear syzygies.

As before, it is possible to build an interpolation matrix H, for given degree ν, containing the values of the unknown syzygy monomials at the parametric set of points. The matrix kernel yields the polynomials in the basis of the syzygy grading of degree ν. For a sufficiently generic point sample, the matrix corank h equals the dimension N_ν.

Using Lemma 3, namely the quadratic formula $N_\nu = \frac{3\nu^2}{2} - \nu(d - \frac{9}{2}) - \frac{d(d+3)}{2} + 3$, we can design an algorithm for computing d and interpolate the syzygies beyond the critical degree ν_0, required for the implicitization matrix of Proposition 1.

The algorithm uses three positive degree values $0 < \nu_1 < \nu_2 < \nu_3$, and computes the 3 respective dimensions N_i, $i = 1, 2, 3$. Then, it checks whether it is possible to fit the 3 pairs (ν_i, N_i) on the parabolic formula of N_ν as function of ν. If this is possible, we are certain that all 3 values ν_i are such that the quadratic formula for N_ν holds. Even if N_ν as a function of ν is expected to be piecewise with most pieces still known, it is impossible that these 3 points fit another piece, since all pieces are of degree at most 2. Therefore, we can compute d and interpolate the syzygies needed for the implicitization matrix.

Example 3. Consider the canonical Steiner surface *affinely* parameterized as:

$$\mathscr{S} = \left\{ \left(\frac{2st}{s^2+t^2+1}, \frac{2t}{s^2+t^2+1}, \frac{2s}{s^2+t^2+1} \right) \in \mathbb{C}^3 : t,s \in \mathbb{C} \right\} \tag{26}$$

The degree of the surface $\mathscr{S}$ is $d = 2$.

Given random points on $\mathscr{S}$ for various values of the parameters t,s, denoted as 5-tuples of the form $(\tau_k, \sigma_k; X_k, Y_k, Z_k)$, we construct the matrix H.

For $\nu_1 = 1$, the $\mathbb{C}$-basis of $Syz(X,Y,Z)_1$ is $\{(-1,0,t,0),(-1,s,0,0)\}$, that is we have $h = 2$. Since we have shown that for $\nu = d-1$ we have $h = d$ linear syzygies, we have successfully computed the degree of the surface, i.e. $d = \nu_1 + 1 = 2$. Thus, any choice of ν such that $\nu \geq 2(d-1) = 2$ is a valid choice to construct an implicit representation matrix $\mathbb{M}_\nu(X,Y,Z)$.

For $\nu_2 = 2$, the matrix is

$$\mathbb{M}_{\nu_2}(X,Y,Z) = \begin{bmatrix} 0 & Z & 0 & 0 & 0 & 0 & 0 & -X/2 & -Y/2 \\ 0 & 0 & Z & 0 & Y & 0 & 0 & 1 & 0 \\ -Z/2 & 0 & 0 & 0 & 0 & Y & 0 & -X/2 & 0 \\ -X/2 & -X & 0 & Z & -X & 0 & 0 & 0 & 1 \\ 1 & 0 & -X & 0 & 0 & -X & Y & 0 & -X/2 \\ -Z/2 & 0 & 0 & -X & 0 & 0 & -X & -X/2 & -Y/2 \end{bmatrix}. \tag{27}$$

5 Implementation and experiments

We experimented using different curves and surfaces of different degrees including the curves we use as examples. All experiments were implemented in Maple 18. The experiments were executed as follows. We start by a given rational parameterization of either a curve or a surface that has no basepoints. That is, we are given a set of 3 or 4 polynomials `pols`, that is the parameterization of the geometric object in projective space. Then, for random values in the parametric domain we sample the corresponding points on the curve or the surface. We use this parametric pointset for our computations. The parameterization is not used explicitly in our computations apart from verifying the results of our method.

We use this parametric pointset to construct matrix H for a given degree ν, as described in the previous sections. After computing its kernel, we obtain the syzygies that form a basis of the syzygies of degree ν. An example use of our implementation is the command

```
> syzygiesd(pols, t, 3),
```

which returns a basis for the syzygies of degree 3 of the polynomials in `pols`, whose parameter is `t`. For different values for the degree ν we look at the number of syzygies we obtain, i.e. the dimension of syzygies of degree ν, and verify their relation to the degree of the parameterization.

The implementation along with the examples included in this paper can be made available upon demand from the authors.

6 Conclusion and future work

We provide a method for computing a matrix representation of a rational planar or space curve, when we are only given a sufficiently large set of points on the object sampled in such a way that the value of the parameter is known. The algorithm holds for curves without base points in an ambient space of arbitrary dimension, as well as for rational surfaces defined over a triangular patch, again without base points.

One obvious generalization is tensor-product surfaces of bi-degree (d_1, d_2), with parameterization

$$\phi : \mathbb{P}^1 \times \mathbb{P}^1 \to \mathbb{P}^3 : \quad t = (s:t;u:v) \to (f_1(t), \ldots, f_4(t)),$$

where every f_i is bi-homogeneous of degree d_1 in $(s:t) \in \mathbb{P}^1$ and degree d_2 in $(u:v) \in \mathbb{P}^1$. In this case, the lack of tight bounds on the degree of the basis of the syzygy module implies that only an approximation to the implicit representation may be obtained.

Future work should involve numerical experiments for interpolating a matrix representation: in this scenario, results are approximate and we wish to quantify the quality of the approximate implicit matrix representation using numerical rank computations. A similar aspect is to consider that noise corrupts the sampling: an estimate of the necessary degree of the syzygies may be obtained in order to interpolate them, thus constructing a matrix approximating the implicit object.

Acknowledgments

The first and third authors are members of “AROMATH", a joint team between INRIA Sophia-Antipolis (France) and National Kapodistrian University of Athens. All authors are partially supported by the Initial Training Network “ARCADES: Algebraic Representations in Computer-Aided Design for Complex Shapes", 2016-2019. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675789. We also thank Laurent Busé for useful discussions during the preparation of this manuscript.

References

[BCD03] L. Busé, D. Cox, and C. D’Andrea. Implicitization of surfaces in the projective space in the presence of base points, 2003.

[BLB10] L. Busé and T. Luu Ba. Matrix-based implicit representations of rational algebraic curves and applications. *Computer-Aided Geometric Design*, 27(9):681–699, 2010.

[BLB12] L. Busé and T. Luu Ba. The surface/surface intersection problem by means of matrix based representations. *Computer-Aided Geometric Design*, 29(8):579–598, 2012.

[Bus14] L. Busé. Implicit matrix representations of rational Bézier curves and surfaces. *Computer-Aided Design*, 46:14–24, 2014. Spec. Issue 2013 SIAM Conf. Geometric & Physical Modeling.

[CGZ00] D. Cox, R. Goldman, and M. Zhang. On the validity of implicitization by moving quadrics for rational surfaces with no base points. *J. Symb. Comput.*, 29(3):419–440, 2000.

[CLO05] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Number 185 in GTM. Springer, New York, 2nd edition, 2005.

[Cox01] D.A. Cox. Equations of parametric curves and surfaces via syzygies. In *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, pages 1–20. AMS, 2001.

[Cox03] D.A. Cox. Curves, surfaces, and syzygies. In *Topics in algebraic geometry and geometric modeling*, volume 334 of *Contemporary Mathematics*, pages 131–150. AMS, 2003.

[CSC98] D.A. Cox, T.W. Sederberg, and F. Chen. The moving line ideal basis of planar rational curves. *Comput. Aided Geom. Des.*, 15(8):803–827, September 1998.

[Dok01] T. Dokken. Approximate implicitization. In *Mathematical methods for curves and surfaces (Oslo 2000)*, Innov. Appl. Math., pages 81–102. Vanderbilt Univ. Press, Nashville, 2001.

[EKK15] I.Z. Emiris, T. Kalinka, and C. Konaxis. Geometric operations using sparse interpolation matrices. *Graphical Models*, 82:99–109, November 2015.

[EKKL13] I.Z. Emiris, T. Kalinka, C. Konaxis, and T. Luu Ba. Sparse implicitization by interpolation: Characterizing non-exactness, and an application to computing discriminants. *Computer-Aided Design, Spec. Issue Solid & Physical Modeling*, 45:252–261, 2013.

[FS05] M.S. Floater and T. Surazhsky. Parameterization for curve interpolation. In *Topics in multivariate approximation and interpolation*, pages 101–115. Elsevier, 2005.

[MC92] D. Manocha and J.F. Canny. The implicit representation of rational parametric surfaces. *J. Symbolic Computation*, 13:485–510, 1992.

[SBAD16] J. Shen, L. Busé, P. Alliez, and N. Dodgson. A line/trimmed NURBS surface intersection algorithm using matrix representations. Technical report, INRIA, 2016.

[SC95] T.W. Sederberg and F. Chen. Implicitization using moving curves and surfaces. In R. Cook, editor, *Proc. SIGGRAPH*, pages 301–308. Addison Wesley, 1995.

[SGD97] T. Sederberg, R. Goldman, and H. Du. Implicitizing rational curves by the method of moving algebraic curves. *J. Symbolic Computation*, 23(2):153–175, 1997.

[SSQK94] T.W. Sederberg, T. Saito, D. Qi, and K.S. Klimaszewski. Curve implicitization using moving lines. *Computer-Aided Geometric Design*, 11(6):687–706, 1994.

[SY08] B. Sturmfels and J. Yu. Tropical implicitization and mixed fiber polytopes. In *Software for Algebraic Geometry*, volume 148 of *IMA Volumes in Math. & its Applic.*, pages 111–131. Springer, New York, 2008.

Reduction in Free Modules

C. Fürst[1], G. Landsmann[1]

[1] *Johannes Kepler University Linz, Research Institute for Symbolic Computation (RISC), Austria, {cfuerst,gland}@risc.jku.at*

Abstract

We present recent research results on the Theory of Gröbner Bases for modules on rings of operators. While in the last fifteen years several concrete instances, such as modules over rings of differential-, difference- and Ore-operators have been considered, we have formulated a theory that is applicable to a wide variety of rings including the known results as special cases. To that end, we use the theory of Gröbner reduction that is formulated for free modules over certain (non-commutative) rings.

As a concept, we present a general reduction concept and introduce a particular type of term reduction. We develop a Buchberger-type theory with our generalized reduction concept, and show how this fits into the landscape of existing theories. Let R be a (non-commutative) ring containing a commutative ring $K \subseteq R$ as a subring, where elements in R are K-linear combinations of monomials $\Lambda \subseteq R$. Let now F be a free R-module generated by a finite set E, hence F consists of K-linear combinations of monomials of the form ΛE. We consider a well-ordered set W and a map: $\mathrm{rk} : F \to W$, and take the model-reduction

$$f \longrightarrow g :\Longleftrightarrow \exists x \in X \subseteq F : g = f - x \wedge \mathrm{rk}(g) < \mathrm{rk}(f) \tag{28}$$

An example would be to choose $\mathrm{rk}(\cdot)$ as the leading term LT, that gives the classic theory of leading term reduction.

A further example would be to consider for elements $f, g \in F$ the support, i.e. the set of monomials that appear in f and g with non-zero coefficient. Therefore, we obtain for $W = \mathscr{P}_{fin}(\Lambda E)$, this is, we consider finite sets of monomials of the form $\lambda e \in \Lambda E$. We put on W a total order $<$ defined by

$$\mathrm{supp}(f) < \mathrm{supp}(g) :\Longleftrightarrow \max(\mathrm{supp}(f) \triangle \mathrm{supp}(g)) \in \mathrm{supp}(g).$$

We define *the term reduction* as the relation (28) and the choice $\mathrm{rk} = \mathrm{supp}(\cdot)$. It turns out that leading term reduction can be viewn to be (properly) contained in term reduction. We point out further characterizations of this term reduction, and relate this to classic situations, in particular, we relate classic Buchberger theory and Gröbner reduction under the scope of this term reduction.

References

1. C. Fürst, *Axiomatic Description of Gröbner Reduction*, Ph.D. Thesis, RISC (2016).
2. C. Fürst, G. Landsmann *Three Examples of Gröbner Reduction over Non-Commutative Rings*, RISC Technical Report 15-16).

3. C. Fürst, G. Landsmann *Computation of Dimension in Filtered Free Modules by Gröbner Reduction*, Proceedings of the ISSAC 2015, 181–188.
4. C. Fürst, A. Levin *Relative Reduction and Buchberger's Algorithm in Filtered Free Modules*, Proceedings of the ACA 2016 Kassel, (2017).

Constructing small cellular free resolutions for monomial ideals

J. Àlvarez Montaner[1], O. Fernández-Ramos[2], P. Gimenez[3]

[1] *Universitat Politècnica de Catalunya, Spain, Josep.Alvarez@upc.edu*
[2] *Universidad de Valladolid, Spain, caribefresno@gmail.com*
[3] *Universidad de Valladolid, Spain, pgimenez@agt.uva.es*

1 State of the art, objectives and methodology

Let $R = K[x_1, \ldots, x_n]$ be the polynomial ring over a field K and $I \subseteq R$ a monomial ideal. The study of minimal free resolutions of these ideals has been a very active area of research during the last decades. There are topological and combinatorial formulae, as those of Hochster [11] or Gasharov, Peeva and Welker [10], to describe their multigraded Betti numbers but, except for some specific classes of monomial ideals (see, e.g., [3], [4] or [14]), the problem of describing a minimal multigraded free resolution explicitly has shown to be difficult.

Another strategy is to study non-minimal free resolutions. These reveal less information than minimal free resolutions do but are often much easier to describe. The most significant ones, that we will also comment in this paper, are the Taylor resolution [16] and the Lyubeznik resolution [13], but one should also mention the Scarf resolution of arbitrary monomial ideals obtained by deformation of exponents [4] and the hull resolution [5]. An interesting feature of the Taylor and the Lyubeznik resolutions is that they fit in the theory of simplicial resolutions introduced by Bayer, Peeva and Sturmfels in [4] and further extended to regular cellular resolutions and CW-resolutions in [5] and [12] respectively. The idea behind these three concepts is to associate to a free resolution of a monomial ideal a simplicial complex (respectively a regular cell complex, a CW-complex) that carries in its structure the algebraic structure of the free resolution. It is worth pointing out that Velasco proved in [17] that there exist monomial ideals whose minimal free resolutions cannot be described by a CW-complex.

By adapting the discrete Morse theory developed by Forman [8] and Chari [6], Batzies and Welker provided in [3] a method to reduce a given regular cellular resolution. In particular, they proved that the Lyubeznik resolution can be obtained in this way from the Taylor resolution. Let's point out that discrete Morse theory has the inconvenient that it can't be used iteratively. To overcome this issue, one can use the algebraic discrete Morse theory developed independently by Sköldberg [15] and Jöllenbeck and Welker [12].

In this work, we use a similar strategy to reduce the Taylor resolution and obtain cellular and simplicial free resolutions that are closer to the minimal one than the

Lyubeznik resolution. Essentially, the information given by the Taylor resolution can be encoded in a directed graph and the obstruction to its minimality can be observed in some of the edges of this graph. What we will do is to remove, in a convenient order, some of these edges to provide a smaller resolution. In some sense, we are pruning the excess of information given by the Taylor resolution in a simple and efficient way.

This is an extended abstract of the unpublished paper [2].

2 Main results and algorithms

Consider a monomial ideal $I = \langle m_1, \ldots, m_r \rangle \subseteq R$. Recall that a CW-complex X is a topological space obtained by attaching cells of increasing dimensions to a discrete set of points $X^{(0)}$. Let $X^{(i)}$ denote the set of i-cells of X and consider the set of all cells $X^{(*)} := \bigcup_{i \geq 0} X^{(i)}$. Then, we can view $X^{(*)}$ as a poset with the partial order given by $\sigma' \leq \sigma$ if and only if σ' is contained in the closure of σ. We can also give a $\mathbf{Z}^n$-graded structure to X by means of an order preserving map $gr : X^{(*)} \longrightarrow \mathbf{Z}^n_{\geq 0}$.

We say that a free resolution

$$\mathbf{F}_\bullet : \quad 0 \to F_p \xrightarrow{d_n} \cdots \longrightarrow F_1 \xrightarrow{d_1} F_0 \longrightarrow R/I \to 0$$

of R/I is *cellular* (or is a *CW-resolution*) if there exists a $\mathbf{Z}^n$-graded CW-complex (X, gr) such that, for all $i \geq 1$:

- there exists a basis $\{e_\sigma\}$ of F_i indexed by the $(i-1)$-cells of X, such that if $e_\sigma \in R(-\alpha)^{\beta_{i,\alpha}}$ then $gr(\sigma) = \alpha$, and
- the differential $d_i : F_i \longrightarrow F_{i-1}$ is given by

$$e_\sigma \;\; \mapsto \sum_{\sigma \geq \sigma' \in X^{(i-1)}} [\sigma : \sigma'] \, \mathbf{x}^{gr(\sigma) - gr(\sigma')} \, e_{\sigma'} \, , \quad \forall \sigma \in X^{(i)},$$

  where $[\sigma : \sigma']$ denotes the coefficient of σ' in the image of σ by the differential map in the cellular homology of X.

In the sequel, whenever we want to emphasize such a cellular structure, we will denote the free resolution as $\mathbf{F}_\bullet = \mathbf{F}_\bullet^{(X,gr)}$. If X is a simplicial complex, we say that the free resolution is *simplicial*. This is the case for two well-known examples, the Taylor resolution ([16]) and the Lyubeznik resolution ([13]):

- *The Taylor resolution*: Consider the full simplicial complex on r vertices, $X_{\texttt{Taylor}}$, whose faces are labelled by $\sigma \in \{0,1\}^r$ or, equivalently, by the corresponding monomials m_σ. We have a natural $\mathbf{Z}^n$-grading on $X_{\texttt{Taylor}}$ by assigning $gr(\sigma) = \alpha \in \mathbf{Z}^n$ where $\mathbf{x}^\alpha = m_\sigma$. The Taylor resolution is the simplicial resolution $\mathbf{F}_\bullet^{(X_{\texttt{Taylor}}, gr)}$.

- *The Lyubeznik resolution*: Let's s start fixing an order $m_1 \leq \cdots \leq m_r$ on a generating set of a monomial ideal $I \subseteq R$. Consider the simplicial subcomplex $X_{\mathtt{Lyub}} \subseteq X_{\mathtt{Taylor}}$ whose faces of dimension s are labelled by those $\sigma = \varepsilon_{i_0} + \cdots + \varepsilon_{i_s} \in \{0,1\}^r$ such that, for all $t < s$ and all $j < i_t$

$$m_j \not| \ \operatorname{lcm}(m_{i_t}, \ldots, m_{i_s}).$$

The Lyubeznik resolution is the simplicial resolution $\mathbf{F}_\bullet^{(X_{\mathtt{Lyub}}, gr)}$.

Our algorithm that constructs, starting from the Taylor resolution, a smaller cellular resolution, is based on the discrete Morse theory. Forman introduced in [8] the discrete Morse theory as a method to reduce the number of cells in a CW-complex without changing its homotopy type. Batzies and Welker adapted this technique in [3] to the study of cellular resolutions; see also [18]. Indeed, they used the reformulation of discrete Morse theory in terms of acyclic matchings given by Chari in [6] in order to obtain, given a regular cellular resolution (most notably the Taylor resolution), a reduced cellular resolution.

Let's start recalling from [3] the preliminaries on discrete Morse theory. Consider the directed graph G_X on the set of cells of a regular $\mathbf{Z}^n$-graded CW-complex (X, gr) which edges are given by

$$E_X = \{\sigma \longrightarrow \sigma' \ \mid \ \sigma' \leq \sigma, \ \ \dim \sigma' = \dim \sigma - 1\}.$$

For a given set of edges $\mathscr{A} \subseteq E_X$, denote by $G_X^{\mathscr{A}}$ the graph obtained by reversing the direction of the edges in $\mathscr{A}$, i.e., the directed graph with edges[1]

$$E_X^{\mathscr{A}} = (E_X \setminus \mathscr{A}) \cup \{\sigma' \Longrightarrow \sigma \ \mid \ \sigma \longrightarrow \sigma' \in \mathscr{A}\}.$$

When each cell of X occurs in at most one edge of $\mathscr{A}$, we say that $\mathscr{A}$ is a *matching* on X. A matching $\mathscr{A}$ is *acyclic* if the associated graph $G_X^{\mathscr{A}}$ is acyclic, i.e., does not contain any directed cycle. Given an acyclic matching $\mathscr{A}$ on X, the *$\mathscr{A}$-critical cells* of X are the cells of X that are not contained in any edge of $\mathscr{A}$. Finally, an acyclic matching $\mathscr{A}$ is *homogeneous* whenever $gr(\sigma) = gr(\sigma')$ for any edge $\sigma \longrightarrow \sigma' \in \mathscr{A}$.

The main result in discrete Morse theory applied to free resolutions that we use is the following:

Theorem ([3, Theorem 1.3]) *Let $I \subseteq R = K[x_1, \ldots, x_n]$ be a monomial ideal. Assume that (X, gr) is a regular $\mathbf{Z}^n$-graded CW-complex that defines a cellular resolution $\mathbf{F}_\bullet^{(X,gr)}$ of R/I. Then, for a homogeneous acyclic matching $\mathscr{A}$ on G_X, the $\mathbf{Z}^n$-graded CW-complex $(X_{\mathscr{A}}, gr)$ supports a cellular resolution $\mathbf{F}_\bullet^{(X_{\mathscr{A}},gr)}$ of R/I.*

We can now present our main algorithm to construct a small cellular resolution of I. Our starting point is the Taylor resolution $\mathbf{F}_\bullet^{(X_{\mathtt{Taylor}},gr)}$. This resolution is, in

[1] For the sake of clarity, the arrows that we reverse will be denoted by $\Longrightarrow$.

general, far from being minimal. In other words, the directed graph $G_{X_{\mathtt{Taylor}}}$ associated to $X_{\mathtt{Taylor}}$ contains a lot of unnecessary information. Our goal is to prune this excess of information in a very simple way. More precisely, we give an algorithm that produces a homogeneous acyclic matching $\mathscr{A}_P$ on $X_{\mathtt{Taylor}}$. Using [3, Theorem 1.3] that we recalled before, this will provide a cellular free resolution of R/I. It will not be minimal in general, but it will be smaller than the Lyubeznik resolution.

Algorithm 1 (*Pruned resolution*)
INPUT: The set of edges $E_{X_{\mathtt{Taylor}}}$.
For j from 1 to r, incrementing by 1:

(j) *Prune* the edge $\sigma \longrightarrow \sigma + \varepsilon_j$ for all $\sigma \in \{0,1\}^r$ such that $\sigma_j = 0$, where 'prune' means remove the edge[2] if it survived after step $(j-1)$ and $gr(\sigma) = gr(\sigma+\varepsilon_j)$.
RETURN: The set $\mathscr{A}_P$ of edges that have been pruned.

Our main result is the following:

Theorem *Let $\mathscr{A}_P \subseteq E_{X_{\mathtt{Taylor}}}$ be the set of pruned edges obtained using Algorithm 1. Then $\mathscr{A}_P$ is a homogeneous acyclic matching on $X_{\mathtt{Taylor}}$.*

As a consequence, we get our desired cellular free resolution.

Corollary *Let $I \subseteq R = K[x_1,\ldots,x_n]$ be a monomial ideal and $\mathscr{A}_P \subseteq E_{X_{\mathtt{Taylor}}}$ be the set of pruned edges obtained using Algorithm 1. Then, the $\mathbf{Z}^n$-graded CW-complex $(X_{\mathscr{A}_P}, gr)$ supports a cellular free resolution $\mathbf{F}_\bullet^{(X_{\mathscr{A}_P},gr)}$ of R/I.*

The resolution that we obtain is not simplicial in general, but we can adapt our pruning algorithm to produce a simplicial free resolution:

Algorithm 2 (*Simplicial pruned resolution*)
INPUT: The set of edges $E_{X_{\mathtt{Taylor}}}$.
For j from 1 to r, incrementing by 1:

(j) *Prune* the edge $\sigma \longrightarrow \sigma + \varepsilon_j$ for all $\sigma \in \{0,1\}^r$ such that $\sigma_i = 0$, where 'prune' means remove the edge if it survived after step $(j-1)$, $gr(\sigma) = gr(\sigma+\varepsilon_j)$ and no face $\tau > \sigma$ survives at this step (j).
RETURN: The set $\mathscr{A}_S$ of edges that have been pruned.

Indeed, the Lyubeznik resolution also fits into this pruning strategy:

Algorithm 3 (*The Lyubeznik resolution via pruning*)
INPUT: The set of edges $E_{X_{\mathtt{Taylor}}}$.
For j from 1 to r, incrementing by 1:

[2] When we remove an edge, we also remove its two vertices and all the edges passing through these two vertices.

(j) *Prune* the edge $\sigma \longrightarrow \sigma + \varepsilon_j$ for all $\sigma \in \{0,1\}^r$ such that $\sigma_i = 0$ for all $i \leq j$, where 'prune' means remove the edge if it survived after step $(j-1)$ and $gr(\sigma) = gr(\sigma + \varepsilon_j)$.

RETURN: The set $\mathscr{A}_L$ of edges that have been pruned.

One deduces from Algorithms 2 and 3 analogous results to the theorem and the corollary obtained from Algorithm 1: $\mathscr{A}_S$ and $\mathscr{A}_L$ are acyclic matchings on $X_{\texttt{Taylor}}$, and the corresponding free resolutions $\mathbf{F}_\bullet^{(X_{\mathscr{A}_S}, gr)}$ and $\mathbf{F}_\bullet^{(X_{\mathscr{A}_L}, gr)}$ are simplicial free resolutions of the monomial ideal I. Other variants of our method are also mentioned in [2].

We will illustrate our results with several examples. We implemented our algorithms using CoCoALib [1] for constructing pruned resolutions in some non-trivial examples.

Finally, we will present a connection between our method and the theory of Betti splittings introduced by Eliahou and Kervaire [3] and later developed by Francisco, Hà and Van Tuyl [9]. We provide a sufficient condition for having a Betti splitting by checking some prunings in our algorithm. We can use this approach to prove that the pruned resolution obtained applying Algorithm 1 is minimal for edge ideals associated to paths and cycles.

Aknowledgements

The first author is partially supported by the Spanish *Ministerio de Economía y Competitividad* grant MTM2015-69135-P and the *Generalitat de Catalunya* grant 2014SGR-634. He is a member of the Barcelona Graduate School of Mathematics (BGSMath). The third author is partially supported by the Spanish *Ministerio de Economía y Competitividad* grant MTM2016-78881-P. He is a member of the *Instituto de Investigación en Matemáticas* (IMUVA).

References

1. J. Abbot and A. M. Bigatti, CoCoALib: a C++ library for doing Computations in Commutative Algebra, available at `http://cocoa.dima.unige.it/cocoalib`.
2. J. Àlvarez Montaner, O. Fernández-Ramos and P. Gimenez, *Pruned cellular free resolutions of monomial ideals,* arXiv:1701.01134, 2017.
3. E. Batzies and V. Welker, *Discrete Morse theory for cellular resolutions*, J. Reine Angew. Math. **543**, pp. 147–168 (2002).
4. D. Bayer, I. Peeva and B. Sturmfels, *Monomial resolutions*, Math. Res. Lett. **5**, pp. 31–46 (1998).
5. D. Bayer and B. Sturmfels, *Cellular resolutions of monomial modules*, J. Reine Angew. Math. **502**, pp. 123–140 (1998).

6. M. K. Chari, *On discrete Morse functions and combinatorial decompositions*, Discrete Math. **217**, pp. 101–113 (2000).
7. S. Eliahou and M. Kervaire, *Minimal resolutions of some monomial ideals*, J. Algebra **129**, pp. 1–25 (1990).
8. R. Forman, *Morse theory for cell complexes*, Adv. Math. **134**, pp. 90–145 (1998).
9. C. Francisco, H. T. Hà and A. Van Tuyl, *Splittings of monomial ideals*, Proc. Amer. Math. Soc. **137**, pp. 3271–3282 (2009).
10. V. Gasharov, I. Peeva, and V. Welker, *The lcm-lattice in monomial resolutions*, Math. Res. Lett. **6**, pp. 521–532 (1999).
11. M. Hochster, *Cohen-Macaulay rings, combinatorics, and simplicial complexes*. In: Ring theory, II (Proc. Second Conf., Univ. Oklahoma, Norman, Okla., 1975), Lecture Notes in Pure and Appl. Math. **26**, Dekker, New York, pp. 171–223 (1977).
12. M. Jöllenbeck and V. Welker, *Minimal resolutions via algebraic discrete Morse theory*, Mem. Amer. Math. Soc. **197** (2009).
13. G. Lyubeznik, *A new explicit finite free resolution of ideals generated by monomials in an R-sequence*, *J. Pure and Appl. Algebra* **51**, pp. 193–195 (1988).
14. E. Miller, B. Sturmfels and K. Yanagawa, *Generic and cogeneric monomial ideals*, J. Symbolic Comput. **29**, pp. 691–708 (2000).
15. E. Sköldberg, *Morse theory from an algebraic viewpoint*, Trans. Amer. Math. Soc. **358**, pp. 115–129 (2006).
16. D. Taylor, *Ideals generated by an R-sequence*, PhD-Thesis, University of Chicago (1966).
17. M. Velasco, *Minimal free resolutions that are not supported by a CW-complex*, J. Algebra **319**, pp. 102–114 (2008).
18. V. Welker, *Discrete Morse theory and free resolutions*. In: Algebraic Combinatorics, Universitext, Springer, Berlin, pp. 81–172 (2007).

Low Autocorrelation Binary Sequences (LABS)

Ilias S. Kotsireas

Wilfrid Laurier University
Department of Physics
and Computer Science
75 University Avenue West
Waterloo, Ontario N2L 3C5
CANADA
e-mail: ikotsire@wlu.ca

Abstract

We will describe the LABS problem, a challenging optimization problem that arises in mathematics, communications engineering and statistical physics. We will discuss the state-of-the-art algorithmic techniques to solve this problem as well as some complexity estimates derived from experimental work by various authors. The algorithmic techniques used in the LABS problem include branch and bound methods, group theory, high-performance (parallel) computing and computer algebra. We will also mention the open problems in the realm of LABS.

A Signature Based Border Basis Algorithm

J. Horáček[1], M. Kreuzer[1], and A.S. Messeng Ekossono[1]

[1] *Faculty of Informatics and Mathematics, University of Passau, Germany, {Jan.Horacek, Martin.Kreuzer, Ange-Salome.MessengEkossono}@uni-passau.de*

Abstract

One of the central algorithms of computer algebra is the algorithm for computing Gröbner bases introduced by B. Buchberger in 1965 (cf. [1]). Significant efforts have been expended to improve its performance. The best current implementations use signature based versions of Buchberger's algorithm, the first of which was J.-C. Faugere's algorithm F5 (cf. [3]). Nowadays an entire zoo of such algorithms has been developed and their behavior has been studied thoroughly (see for instance [2]).

On the other hand, the Border Basis Algorithm (BBA), a framework for which was introduced in [6] and whose details were worked out in [5], is much less researched. In [4], the authors considered some optimizations of BBA for ideals of Boolean polynomials. However, for interesting ideals originating from cryptographic attacks, these optimizations still proved to be insufficient to produce running times comparable with optimized implementations of Buchberger's algorithm. The main reason for this is that the current implementations of the BBA still lack analogues of Buchberger's criteria for avoiding unnecessary reductions of critical pairs.

Our main goal in this presentation is to go the first step in the direction of constructing border basis analogues to these criteria. More specifically, let $I = \langle f_1, \ldots, f_s \rangle$ be the 0-dimensional ideal whose border basis we are calculating, and let V be the current tuple of polynomials generating a vector space which will contain the desired border basis eventually. To each polynomial g that we have to consider, we assign a *signature bound* which is a pair (t, i) with a term t and $1 \leq i \leq s$ that remembers the multiple $t f_i$ of the input polynomial f_i whose (linear) reduction is g. Thus, if the same signature bound appears again, we can avoid the reduction of g against the polynomials in V because we know that it reduces to zero. As we shall see, in this way we avoid many repetitions and the algorithm becomes significantly faster.

References

1. B. Buchberger, Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal, J. Symbolic Comput. **41** (2006), 475-511.
2. C. Eder and J.-C. Faugère, A survey on signature-based algorithms for computing Gröbner bases, J. Symbolic Comput. **80** (2017), 719-784.
3. J.-C. Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), in: Proc. Conf. ISSAC 2002, ACM Press, New York 2002, pp. 75-83.

4. J. Horacek, M. Kreuzer, and A.-S. Messeng Ekossono, Computing Boolean border bases, Proc. Conf. SYNASC'16, Timisoara 2016, IEEE (to appear), available at
www.sc-square.org/CSA/workshop1-papers/paper3.pdf
5. A. Kehrein and M. Kreuzer, Computing border bases, J. Pure Appl. Alg. **205** (2006), 279-295.
6. B. Mourrain, A new criterion for normal form algorithms, in: M. Fossorier, H. Imai, S. Lin, A. Poli (eds.), Proc. Conf. AAECC-13, Honolulu 1999, LNCS **1719**, Springer Verlag, Heidelberg 1999, pp. 440-443.

Gröbner reduction in modules over arbitrary rings

G. Landsmann[1], C. Fürst[1]

[1] *Johannes Kepler University Linz, Research Institute for Symbolic Computation (RISC), Austria,{landsmann,cfuerst}@risc.jku.at*

Abstract

Given a pair of abelian groups $N \subseteq M$, a reduction is a relation $\rho \subset M \times M$ that generates the congruence modulo N. Membership to such a relation should be decidable and any chain of iterated reduction steps $u_1 \longrightarrow u_2 \longrightarrow \cdots$ should terminate.

A Gröbner reduction for $N \subseteq M$ is a reduction which induces a splitting of the exact sequence

$$0 \longrightarrow N \longrightarrow M \xrightarrow{\pi} M/N \longrightarrow 0 \tag{29}$$

Chosen a splitting $s\colon M/N \longrightarrow M$, the endomorphism $s \circ \pi$ provides the normal form for elements $u \in M$; its image I is the group of irreducibles.

Any reduction ρ for the extension $N \subseteq M$ can be desribed by the scheme

$$u \longrightarrow v \iff u - v \in X \wedge P(u,v) \tag{30}$$

where $X \subseteq M$ is a set and P denotes a binary predicate. These are the parameters which determine the behaviour of the reduction ρ, and for constructing ρ one has to exploit the structure provided by the actual candidates M and N. Of course, the main attention is layed on modules over certain rings R whose structure is rich enough to allow algorithmically performing the construction.

The well-known classical examples over commutative Noetherian rings rely on the existence of a recursive well-order which is the principal ingredient of the predicate P. The set X consists of generators of the module N and is typically split into a product of sets $X = A \cdot G$, with $A \subset R$ and $G \subset N$. The set G is then called a Gröbner basis of N.

The existence of a splitting s as well as the disposability of an appropriate well-order is based on the presence of a set Λ of monomials in the ring R. Classically these monomials constitute a basis of R over some central subfield K with the result that R is an algebra over K. If then the monomials build a monoid isomorphic to some $\mathbb{N}^n$ and the module M is finite free over R, the construction of G can be accomplished - with minor adaption - by the classical Buchberger algorithm.

There are important rings where the subfield K is not central or the set Λ is not a monoid, e.g., Weyl algebras, Ore algebras or rings of difference-differential operators, where recent work has brought progress into the construction of Gröbner bases for submodules of finite free modules over such rings. The - sometimes hidden -

spine of the construction process is always a Gröbner reduction.

But even in the absence of monomials a Gröbner reduction may be possible and it remains the desire to gain insight into the nature of their totality.

To proceed into this direction we consider two Gröbner reductions on $N \subset M$ as equivalent when they induce equal splittings. The class of all equivalence classes of Gröbner reductions has then a natural group structure. Given an arbitrary element of this group, we may describe the production of normal forms as a certain limit. We will work out these ideas and illustrate their alliance to other concepts like filtrations or Robbiano's graded structures.

References

1. C. Fürst, *Axiomatic Description of Gröbner Reduction*, Ph.D. Thesis, RISC (2016).
2. C. Fürst, G. Landsmann *Three Examples of Gröbner Reduction over Non-Commutative Rings*, RISC Technical Report 15-16).
3. C. Fürst, G. Landsmann *Computation of Dimension in Filtered Free Modules by Gröbner Reduction*, Proceedings of the ISSAC 2015, 181–188.
4. C. Fürst, A. Levin *Relative Reduction and Buchberger's Algorithm in Filtered Free Modules*, Proceedings of the ACA 2016 Kassel, (2017).

The algebra of Kleene stars of the plane and polylogarithms

G.H.E. Duchamp[1,4], Hoang Ngoc Minh[2,4], Q.H. Ngo[3]

[1] *Université Paris 13, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France, (gheduchamp@gmail.com).*
[2] *Université Lille 2, 1, Place Déliot, 59024 Lille, France, (hoang@univ-lille2.fr).*
[3] *University of Hai Phong, 171, Phan Dang Luu, Hai Phong, Viet Nam, (quochoan_ngo@yahoo.com.vn).*
[4] *LIPN-UMR 7030, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France.*

2 Introduction

As a matter of fact, the interest of rational series, over the alphabets $Y_0 = \{y_n\}_{n\in\mathbb{N}}$, $Y = Y_0 \setminus \{y_0\}$ and $X = \{x_0, x_1\}$, is twofold : algebraic and analytic.

Firstly, (from the algebraic point of view) they are closed under shuffle products and the shuffle exponential of letters (and their linear combinations) is precisely their Kleene star[3]. Secondly, the growth of their coefficients is tame[4] [7, 17, 18] and as such their associated polylogarithms can be rightfully computed [12, 15, 16].

Doing this, we recover many functions (as the simple polynomials), forgotten in the straight algebra of polylogarithms, at positive indices, which are viewed as image by the following isomorphism of algebras [13]

$$\mathrm{Li}_\bullet : (\mathbb{C}\langle X\rangle, \sqcup\!\sqcup, 1_{X^*}) \longrightarrow (\mathbb{C}\{\mathrm{Li}_w\}_{w\in X^*}, \times, 1),\ x_0^{s_1-1}x_1\ldots x_0^{s_r-1}x_1 \longmapsto \mathrm{Li}_{s_1,\ldots,s_r},$$

and, for $n \geq 0, x_0^n \longmapsto \mathrm{Li}_{x_0^n}(z) = \log^n(z)/n!$ and $x_1^n \longmapsto \mathrm{Li}_{x_1^n}(z) = \log^n((1-z)^{-1})/n!$. To study the multi-indexed polylogarithms, one relies on the one-to-one correspondence between the multi-indices $(s_1,\ldots,s_r)$, in $\mathbb{Z}^r_{\leq 0}$ or $\mathbb{N}^r_+$, and the words $y_{s_1}\ldots y_{s_r}$, in the monoid Y_0^*, for indexing polylogarithms by $y_{s_1}\ldots y_{s_r}$ [5, 7, 13, 14] :

$$\mathrm{Li}_{y_{s_1}\ldots y_{s_r}} = \mathrm{Li}_{s_1,\ldots,s_r} \text{ and } \mathrm{Li}^-_{y_{s_1}\ldots y_{s_r}} = \mathrm{Li}_{-s_1,\ldots,-s_r}\,.$$

We will explain the whole project to extend $\mathrm{Li}_\bullet$ over a sub algebra of rational power series. In particular, we will study different aspects of $\mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*}$, where $\mathscr{C}$ denotes the ring of polynomials on z, z^{-1} and $(1-z)^{-1}$, with coefficents in $\mathbb{C}$, and we will express polylogarithms (resp. harmonic sums) at negative multi-indices as polynomials on $(1-z)^{-1}$ (resp. N), with coefficients in $\mathbb{Z}$ (resp. $\mathbb{Q}$).

---

[3] *i.e.* for any $S \in \mathbb{C}\langle\langle X\rangle\rangle$ such that $\langle S \mid 1_{X^*}\rangle = 0$, S^* denotes the sum $1_{X^*} + S + S^2 + S^3 + \ldots$ [1].

[4] *i.e.* for such a rational series S over X, there exists a real $K > 0$ and a positive real morphism χ such that, for any $w \in X^*$, the coefficient $|\langle S \mid w\rangle|$ is majorated by $K\chi(w)$ [7, 17, 18].

3 Polylogarithms and algebraic combinatorial frame works

Let us, now, go into details, using the notations of [1, 20],

1. We construct the bialgebras[5] $(\mathbb{C}\langle Y_0\rangle,.,\Delta_{\sqcup\!\sqcup},1_{Y_0^*},\varepsilon)$, and $(\mathbb{C}\langle X\rangle,.,\Delta_{\sqcup\!\sqcup},1_{X^*},\varepsilon)$ in which, for any $i=0,1$ and $j\geq 1$, one has
$$\Delta_{\sqcup\!\sqcup}(y_j)=y_j\otimes 1_{Y^*}+1_{Y^*}\otimes y_j+\sum_{k+l=j}y_k\otimes y_l.\ \Delta_{\sqcup\!\sqcup}(x_i)=x_i\otimes 1_{X^*}+1_{X^*}\otimes x_i.$$
2. Let $\mathbb{C}^{\rm rat}\langle\langle X\rangle\rangle$ denotes the closure of $\mathbb{C}\langle X\rangle$ by rational operations $\{+,.,^*\}$ [1]. It is closed by shuffle. By the Kleene-Schützenberger theorem, any power series S belongs to $\mathbb{C}^{\rm rat}\langle\langle X\rangle\rangle$ if and only if it is *recognizable* by an automaton admitting *linear representation* (β,μ,γ) of dimension $n\geq 1$, with
$$\beta\in\mathscr{M}_{n,1}(\mathbb{C}),\ \mu:X^*\longrightarrow\mathscr{M}_{n,n}(\mathbb{C}),\ \gamma\in\mathscr{M}_{1,n}(\mathbb{C})$$
and, for any $w\in X^*$, one has $\langle S\mid w\rangle=\beta\mu(w)\gamma$ (see [1]).
3. Let us consider the following morphism of algebra
$$\pi_X:(\mathbb{C}\langle Y\rangle,.,1_{Y^*})\longrightarrow(\mathbb{C}\langle X\rangle,.,1_{X^*}),\ y_{s_1}\ldots y_{s_r}\longrightarrow x_0^{s_1-1}x_1\ldots x_0^{s_r-1}x_1.$$
It admits an adjoint π_Y for the two standard scalar products, *i.e.*
$$\forall p\in\mathbb{C}\langle X\rangle,\ \forall q\in\mathbb{C}\langle Y\rangle,\ \langle\pi_Y(p)\mid q\rangle_Y=\langle p\mid\pi_X(q)\rangle_X.$$
One checks that $\pi_Y(x_0^{s-1}x_1)=y_s$, $\ker(\pi_Y)=\mathbb{C}\langle X\rangle x_0$ and π_Y restricted to the subalgebra $(\mathbb{C}\,1_{X^*}\oplus\mathbb{C}\langle X\rangle x_1,.)$ is an isomorphism, inverse of π_X.

In this work, Ω denotes the cleft plane $\mathbb{C}\setminus(]-\infty,0]\cup[1,+\infty[)$ and $\mathscr{H}(\Omega)$ denotes the set of holomorphic functions over the simply connected domain Ω.

The principal object of the present work, as in [5, 7], is the *polylogarithm* well defined, for any $(s_1,\ldots,s_r)\in\mathbb{C}^r,r\in\mathbb{N}_+$ and for any $z\in\mathbb{C}$ such that $|z|<1$, by
$$\mathrm{Li}_{s_1,\ldots,s_r}(z):=\sum_{n_1>\ldots>n_r>0}\frac{z^{n_1}}{n_1^{s_1}\ldots n_r^{s_r}}\ \text{and}\ \frac{\mathrm{Li}_{s_1,\ldots,s_r}(z)}{1-z}=\sum_{N\geq 0}\mathrm{H}_{s_1,\ldots,s_r}(N)\,z^N,$$
where the arithmetic function $\mathrm{H}_{s_1,\ldots,s_r}:\mathbb{N}\longrightarrow\mathbb{Q}$ is expressed by
$$\mathrm{H}_{s_1,\ldots,s_r}(N):=\sum_{N\geq n_1>\ldots>n_r>0}\frac{1}{n_1^{s_1}\ldots n_r^{s_r}}.$$
Here, $\mathrm{Li}_{s_1,\ldots,s_r}$ is obtained as iterated integrals, along the path on Ω and over the differential forms
$$\omega_0(z)=z^{-1}\ \text{and}\ \omega_1(z)=(1-z)^{-1}.$$

[5] Which are all Hopf save the last one due to y_0 which is infiltration-like [2].

After a theorem by Abel, for any $r \geq 1$, if $(s_1,\ldots,s_r) \in \mathscr{H}_r$ then

$$\zeta(s_1,\ldots,s_r) := \lim_{z\to 1} \mathrm{Li}_{s_1,\ldots,s_r}(z) = \lim_{N\to\infty} \mathrm{H}_{s_1,\ldots,s_r}(N),$$

where $\mathscr{H}_r = \{(s_1,\ldots,s_r) \in \mathbb{C}^r | \forall m = 1,\ldots,r; \Re(s_1)+\ldots+\Re(s_r) > m\}$ [10, 21]. This is no more valid in the divergent cases and requires the renormalization of the corresponding divergent polyzetas. It is already done for the case of polyzetas at positive multi-indices [3, 4, 17] and it is done [9, 11, 19] and completed in [5, 7] for the case of negative multi-indices.

Let us consider the following group of transformations which permutes the singularities $\{0,1,+\infty\}$

$$\mathscr{G} := \{z\mapsto z, z\mapsto 1-z, z\mapsto z^{-1}, z\mapsto (1-z)^{-1}, z\mapsto 1-z^{-1}, z\mapsto z(1-z)^{-1}\}.$$

and let us also consider the following rings :

$$\mathscr{C}_0' := \mathbb{C}[z^{-1}], \mathscr{C}_1' := \mathbb{C}[(1-z)^{-1}],\ \ \mathscr{C}_0 := \mathbb{C}[z,z^{-1}], \mathscr{C}_1 := \mathbb{C}[z,(1-z)^{-1}],$$
$$\mathscr{C}' := \mathbb{C}[z^{-1},(1-z)^{-1}],\ \ \mathscr{C} := \mathbb{C}[z,z^{-1},(1-z)^{-1}],$$

which are differential rings, endowed with the differential operator $\partial_z := d/dz$ and with the neutral element $1_\Omega : \Omega \longrightarrow \mathbb{C}$, mapping z to $1_\Omega(z) = 1$. It follows that

Lemma 4. *For any $i = 0$ or 1, one has $\mathscr{C}_i' \subsetneq \mathscr{C}_i \subsetneq \mathscr{C}$ and $\mathscr{C}_i' \subsetneq \mathscr{C}' \subsetneq \mathscr{C}$.*

2. *The differential ring $\mathscr{C}$ is closed under action of $\mathscr{G}$:*

$$\forall G \in \mathscr{C},\ \forall g \in \mathscr{G},\ G(g(z)) \in \mathscr{C}.$$

3. *The sub-rings $\mathscr{C}_0, \mathscr{C}_1$ are closed by the involutions $\{z\mapsto z^{-1}, z\mapsto 1-z\}$, respectively, and are permuted by $\{z\mapsto 1-z^{-1}, z\mapsto z(1-z)^{-1}\}$, respectively.*

Proposition 1. *Let $\theta_0 := z\partial_z$ and $\theta_1 := (1-z)\partial_z$ be the differential operators.*

Let ι_0 and ι_1 be their sections[6]*, i.e. $\theta_0\iota_0 = \theta_1\iota_1 = \mathrm{Id}$. Then*

1. *$[\theta_0,\theta_1] = \theta_0+\theta_1 = \partial_z, [\theta_0\iota_1, \theta_1\iota_0] = 0$ and $(\theta_0\iota_1)(\theta_1\iota_0) = (\theta_1\iota_0)(\theta_0\iota_1) = \mathrm{Id}$.*
2. *If $w = x_0^{s_1-1}x_1\ldots x_0^{s_r-1}x_1 \in X^*x_1$ and $u = y_{t_1}\ldots y_{t_r} \in Y_0^*$ then*

$$\mathrm{Li}_w = (\iota_0^{s_1-1}\iota_1\ldots\iota_0^{s_r-1}\iota_1)1_\Omega, \text{ and } \mathrm{Li}_u^- = (\theta_0^{t_1+1}\iota_1\ldots\theta_0^{t_r+1}\iota_1)1_\Omega.$$

3. *The algebra $\mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*}$ is closed by $\{\theta_0,\theta_1,\iota_0,\iota_1\}$.*
4. *The bi-integro differential ring $(\mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*}, \times, 1_\Omega)$ is closed by $\mathscr{G}$:*

$$\forall l \in \mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*},\ \forall g\in\mathscr{G},\ l(g(z)) \in \mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*}.$$

[6] *i.e.* take primitives for the corresponding differential operators.

4 Algebraic extension of $\mathrm{Li}_\bullet$ to $(\mathbb{C}^{\mathrm{rat}}\langle\langle X\rangle\rangle, \sqcup\!\sqcup, 1_{X^*})$

Under some convergent conditions, the extension of $\mathrm{Li}_\bullet$ over $\mathbb{C}^{\mathrm{rat}}\langle\langle X\rangle\rangle$ can be done as follows : call $\mathrm{Dom}(\mathrm{Li}_\bullet)$ the set of series

$$S = \sum_{n\geq 0} S_n \text{ with } S_n := \sum_{|w|=n} \langle S \mid w\rangle w$$

such that $\sum_{n\geq 0} \mathrm{Li}_{S_n}$ converges uniformly any compact of Ω. Then

Proposition 2. *One has*

1. *The set* $\mathrm{Dom}(\mathrm{Li}_\bullet)$ *is closed by shuffle products.*
2. *For any* $S,T \in \mathrm{Dom}(\mathrm{Li}_\bullet)$*, one has* $\mathrm{Li}_{S\sqcup\!\sqcup T} = \mathrm{Li}_S \mathrm{Li}_T$.
3. *One has* $\mathbb{C}\langle X\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle x_0\rangle\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle x_1\rangle\rangle \subset \mathrm{Dom}(\mathrm{Li}_\bullet)$.

This extension is compatible with identities between rational series as *Lazard's elimination*, for instance :

$$\forall S \in \mathbb{C}^{\mathrm{rat}}\langle\langle S\rangle\rangle,\ \mathrm{Li}_S(z) = \sum_{n\geq 0} \langle S \mid x_0^n\rangle \frac{\log^n(z)}{n!} + \sum_{k\geq 1} \sum_{w\in(x_0^* x_1)^k x_0^*} \langle S \mid w\rangle \mathrm{Li}_w(z),$$

The morphism $\mathrm{Li}_\bullet$ is no more injective but $\{\mathrm{Li}_w\}_{w\in X^*}$ is still $\mathscr{C}$-linearly independant.

We will use several times the following lemma which is characteristic-free.

Lemma 5. *Let* $(\mathscr{A},d)$ *be a commutative differential ring without zero divisor, and* $R = \ker(d)$ *be its subring of constants. Let* $z \in \mathscr{A}$ *such that* $d(z) = 1$ *and* $S = \{e_\alpha\}_{\alpha\in I}$ *be a set of eigenfunctions of d, all different (for example, take* $I \subset R$*) i.e.,* $e_\alpha \neq 0$ *and* $d(e_\alpha) = \alpha e_\alpha; \forall \alpha \in I$*. Then the family* $(e_\alpha)_{\alpha\in I}$ *is* $R[z]$*-linearly free*[7].

Remark 1. If $\mathscr{A}$ is a $\mathbb{Q}$-algebra or only of characteristic zero (*i.e.*, $n\times 1_{\mathscr{A}} = 0 \Rightarrow n = 0$), then $d(z) = 1$ implies that z is transcendent over R.

First of all, let us prove

Lemma 6. *Let* $\mathscr{A}$ *be a* $\mathbb{Q}$*-algebra (associative, unital, commutative) and z an indeterminate, then* $e^z \in \mathscr{A}[[z]]$ *is transcendent over* $\mathscr{A}[z]$.

Proof. It is straightforward consequence of Remark (1). Note that this can be rephrased as $[z, e^z]$ are algebraically independant over $\mathscr{A}$.

Proposition 3. *Let* $Z = \{z_n\}_{n\in\mathbb{N}}$ *be an alphabet, then* $[e^{z_0}, e^{z_1}]$ *is algebraically independent on* $\mathbb{C}[Z]$ *within* $\mathbb{C}[[Z]]$.

[7] Here $R[z]$ is understood as ring adjunction i.e. the smallest subring generated by $R\cup\{z\}$.

Proposition 4. *The family* $\{x_0^*, x_1^*\}$ *is algebraically independent over* $(\mathbb{C}\langle X\rangle, \sqcup\!\sqcup, 1_{X^*})$ *within* $(\mathbb{C}\langle\langle X\rangle\rangle^{\mathrm{rat}}, \sqcup\!\sqcup, 1_{X^*})$.

ii. The module $(\mathbb{C}\langle X\rangle, \sqcup\!\sqcup, 1_{X^*})[x_0^*, x_1^*, (-x_0)^*]$ *is free over* $\mathbb{C}\langle X\rangle$ *and the family* $\{(x_0^*)^{\sqcup\!\sqcup k} \sqcup\!\sqcup (x_1^*)^{\sqcup\!\sqcup l}\}_{(k,l)\in\mathbb{Z}\times\mathbb{N}}$ *is a* $\mathbb{C}\langle X\rangle$*-basis of it.*

iii. As a consequence, $\{w \sqcup\!\sqcup (x_0^*)^{\sqcup\!\sqcup k} \sqcup\!\sqcup (x_1^*)^{\sqcup\!\sqcup l}\}_{\substack{w\in X^* \\ (k,l)\in\mathbb{Z}\times\mathbb{N}}}$ *is a* $\mathbb{C}$*-basis of it.*

Proposition 5. *There is a unique morphism* ν*, from* $(\mathbb{C}\langle X\rangle, \sqcup\!\sqcup, 1_{X^*})[x_0^*, (-x_0)^*, x_1^*]$ *to* $\mathscr{H}(\Omega)$ *defined by, for any* $w \in X^*$*,* $\nu(w) = \mathrm{Li}_w$ *and*

$$\nu(x_0^*) = z,\ \nu((-x_0)^*) = z^{-1},\ \nu(x_1^*) = (1-z)^{-1}.$$

Theorem 1 ([8]). *Denoting* $\mathrm{Li}_\bullet^{(1)}$ *the morphism* ν *defined as in Proposition 5, it realizes then a morphism of algebras,* $\mathbb{C}\langle X\rangle[x_0^*, x_1^*, (-x_0)^*] \longrightarrow \mathscr{H}(\Omega)$*. Moreover,* $\mathrm{Im}(\mathrm{Li}_\bullet^{(1)}) = \mathscr{C}\{\mathrm{Li}_w\}_{w\in X^*}$ *and* $\ker(\mathrm{Li}_\bullet^{(1)})$ *is the ideal generated by* $x_0^* \sqcup\!\sqcup x_1^* - x_1^* + 1_{X^*}$.

Corollary 3 ([12]). *One has*

1. *For* $x \in X, i \in \mathbb{N}_+, a \in \mathbb{C}, |a| < 1$,

$$\mathrm{Li}_{(ax_0)^{*i}}(z) = z^a \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{(a\log(z))^k}{k!},$$
$$\mathrm{Li}_{(ax_1)^{*i}}(z) = \frac{1}{(1-z)^a} \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{(a\log((1-z)^{-1})^k}{k!}.$$

2. *For any* $(s_1, \ldots, s_r) \in \mathbb{N}_+^r$ *and* $(t_1, \ldots, t_r) \in (\mathbb{C} - \mathbb{N}_+)^r$,

$$\mathrm{Li}_{(t_1x_0)^{*s_1}x_0^{s_1-1}x_1\ldots(t_rx_0)^{*s_r}x_0^{s_r-1}x_1}(z) = \sum_{n_1>\ldots>n_r>0} \frac{z^{n_1}}{(n_1-t_1)^{s_1}\ldots(n_r-t_r)^{s_r}}.$$

In particular,

$$\mathrm{Li}_{(t_1x_0)^*x_1\ldots(t_rx_0)^*x_1}(z) = \sum_{n_1,\ldots,n_r>0} \mathrm{Li}_{x_0^{n_1-1}x_1\ldots x_0^{n_r-1}x_1}(z)\, t_0^{n_1-1}\ldots t_r^{n_r-1}.$$

Corollary 4 ([12]). *By Corollary 3, it follows that*

$$\begin{aligned}\{\mathrm{Li}_S\}_{S\in\mathbb{C}\langle X\rangle \sqcup\!\sqcup \mathbb{C}[x_0^*] \sqcup\!\sqcup \mathbb{C}[(-x_0^*)] \sqcup\!\sqcup \mathbb{C}[x_1^*]} &= \mathrm{span}_{\mathbb{C}}\left\{\frac{z^a}{(1-z)^b}\mathrm{Li}_w(z)\right\}_{w\in X^*}^{a\in\mathbb{Z}, b\in\mathbb{N}} \\ &\subset \mathrm{span}_{\mathbb{C}}\{\mathrm{Li}_{s_1,\ldots,s_r}\}_{s_1,\ldots,s_r\in\mathbb{Z}^r} \\ &\quad \oplus \mathrm{span}_{\mathbb{C}}\{z^a | a \in \mathbb{Z}\}, \\ \{\mathrm{Li}_S\}_{S\in\mathbb{C}\langle X\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle x_0\rangle\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle x_1\rangle\rangle} &= \mathrm{span}_{\mathbb{C}}\left\{\frac{z^a}{(1-z)^b}\mathrm{Li}_w(z)\right\}_{w\in X^*}^{a,b\in\mathbb{C}} \\ &\subset \mathrm{span}_{\mathbb{C}}\{\mathrm{Li}_{s_1,\ldots,s_r}\}_{s_1,\ldots,s_r\in\mathbb{C}^r} \\ &\quad \oplus \mathrm{span}_{\mathbb{C}}\{z^a | a \in \mathbb{C}\}.\end{aligned}$$

References

1. J. Berstel & C. Reutenauer, *Rational series and their languages*, Springer-Verlag, 1988.
2. Bui V. C., Duchamp G. H. E., Hoang Ngoc Minh V., Tollu C., Ngo Q. H., *(Pure) transcendence bases in ϕ-deformed shuffle bialgebras*, arXiv:1507.01089v1 [cs.SC].
3. Costermans C., Hoang Ngoc Minh, *Some Results à l'Abel Obtained by Use of Techniques à la Hopf*, "Global Integrability of Field Theories and Applications", Daresbury, 2006.
4. Costermans C., Hoang Ngoc Minh, *Noncommutative algebra, multiple harmonic sums and applications in discrete probability*, J. of Sym. Comp., 801-817, 2009.
5. Gérard H. E. Duchamp, Hoang Ngoc Minh, Ngo Quoc Hoan, *Harmonic sums and polylogarithms at negative multi - indices*, accepted by JSC, 2015.
6. Gérard H. E. Duchamp, Tollu C., *Sweedler's duals and Schützenberger's calculus*, In K. Ebrahimi-Fard, M. Marcolli and W. van Suijlekom (eds), Combinatorics and Physics, 67 - 78, Amer. Math. Soc. (Contemporary Mathematics, vol. 539), 2011.
7. Gérard H. E. Duchamp , Hoang Ngoc Minh, Penson K. A., Ngô Q. H., Simonnet P., *Mathematical renormalization in quantum electrodynamics via noncommutative generating series*, accepted by Springer Proceedings in Mathematics, Conference ACA 2015.
8. Gérard H. E. Duchamp, Hoang Ngoc Minh, Ngo Quoc Hoan, *The algebra $\mathbb{C}\langle\mathbf{X}\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle\mathbf{x_0}\rangle\rangle \sqcup\!\sqcup \mathbb{C}^{\mathrm{rat}}\langle\langle\mathbf{x_1}\rangle\rangle$ and polylogarithms,* 2015.
9. Furusho H., Komori Y., Matsumoto K., Tsumura H., *Desingularization of multiple zeta-functions of generalized Hurwitz-Lerch type*, 2014.
10. Goncharov A. B., *Multiple polylogarithms and mixed Tate motives*, 2001.
11. Guo L., Zhang B., *Renormalization of multiple zeta values*, J. Alg., 319, 3770-3809, 2008.
12. Hoang Ngoc Minh, *Summations of Polylogarithms via Evaluation Transform*, Math. & Computers in Simulations, 1336, 707-728, 1996.
13. Hoang Ngoc Minh, Jacob G., Oussous N. E. , Petitot M., *Aspects combinatoires des polylogarithmes et des sommes d'Euler-Zagier*, J. SLC, B43e, 1998.
14. Hoang Ngoc Minh & Petitot M., *Lyndon words, polylogarithmic functions and the Riemann ζ function*, Discrete Math., 217, 273-292, 2000.
15. Hoang Ngoc Minh, *Differential Galois groups and noncommutative generating series of polylogarithms*, in "Automata, Combinatorics and Geometry". 7th World Multi-conference on Systemics, Cybernetics and Informatics, Florida, 2003.
16. Hoang Ngoc Minh, *Finite polyzêtas, Poly-Bernoulli numbers, identities of polyzêtas and noncommutative rational power series*, Proceedings of 4^{th} I. C. W,232-250, Finland, 2003.
17. Hoang Ngoc Minh, *Algebraic combinatoric aspects of asymptotic analysis of nonlinear dynamical system with singular inputs*, Acta Academiae Aboensis, Ser. B 67 (2), 117-126, 2007.
18. Hoang Ngoc Minh, *On a conjecture by Pierre Cartier about a group of associators*, Acta Math. Vietnamica, 38 (3), 339-398, 2010.
19. Manchon D., Paycha S., *Nested sums of symbols and renormalised multiple zeta functions*, Int Math Res Notices, 24, 4628-4697, 2010.
20. Reutenauer C., *Free Lie Algebras*, London Math. Soc. Monographs, 1993.
21. Zhao J., *Analytic continuation of multiple zeta functions*, Proc. A. M. S., 128 (5), 1275 - 1283, 1999.

Computing The Dedekind Different Of A Smooth Scheme And Applications

L.N. Long

Faculty of Computer Science and Mathematics, University of Passau, D-94030 Passau, Germany, nglong16633@gmail.com

Abstract

Let K be a (computable) field, let $\mathbf{P}_K^n$ be the n-dimensional projective space over K, and let $X \subseteq \mathbf{P}_K^n$ be a 0-dimensional smooth subscheme. We are interested in studying the Dedekind different of the scheme X. This invariant is defined as the inverse ideal of the Dedekind complementary module of the homogeneous coordinate ring R of X in the homogeneous ring of quotients of R. Here the Dedekind complementary module of R can be obtained by embedding the canonical module of R in its homogeneous ring of quotients. In this talk we want to address the problem of computing a minimal homogeneous system of generators of the Dedekind different of X. Our approach is to use the Generalized Buchberger-Möller Algorithm given in the paper [1] and is based on a description of the Dedekind complementary module which generalizes a result of [2]. Moreover, we also apply this invariant to characterize some uniformity conditions of X, and provide several characterizations of schemes X with minimal Dedekind different.

References

1. J. Abbott, M. Kreuzer, L. Robbiano, *Computing zero-dimensional schemes*, J. Symbolic Comput. **39**, pp. 31–49 (2005).
2. M. Kreuzer, *On the canonical ideal of a set of points*, Bollettino U.M.I. (8) 1-B, pp. 221–261 (2000).

Efficient Algorithms for Special Roots of Quaternion Polynomials

P. Dospra[1], D. Poulakis[2]

[1] *petroula.dospra@gmail.com*
[2] *Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki, Greece*
poulakis@math.auth.gr

Abstract

In this paper, polynomial deterministic algorithms are proposed for the computation of the number of distinct spherical roots, isolated complex roots and pure quaternion roots of a quaternion polynomial. Furthermore, two deterministic algorithms are presented for the computation of complex roots and pure quaternion roots of a such polynomial, respectively.

Quaternion Polynomials: Roots and Their Jacobians

Takis Sakkalis

Mathematics Laboratory
Agricultural University of Athens
75 Iera Odos, Athens 11855, GREECE, stp@aua.gr

Abstract

A quaternion polynomial $f(t)$ in the single variable t, is one whose coefficients are in the skew field H of quaternions. f can also be thought as a transformation from $\mathbb{R}^4 \to \mathbb{R}^4$. In this talk we first give an elementary proof of the fact that such an f has a root in H, Ref. [2]. In addition, the Jacobian $J(f)$ of f is computed. As a consequence, Cauchy-Riemman equations for f are derived. It is also shown that the Jacobian determinant $|J(f)|$ is non negative over H. Moreover, ζ is a single root of f if and only if $|J(f)(\zeta)| > 0$, Ref. [1]. The above commensurates well with the theory of analytic functions of one complex variable.

References

1. T. Sakkalis, *Jacobians of quaternion polynomials*. Submitted for publication, (2017).
2. T. Sakkalis, K. Ko and G. Song, *Roots of quaternion polynomials: theory and computation*. Submitted for publication, (2017).

Kähler Differential Algebras For 0-dimensional Schemes

T. N. K. Linh

Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria, tnkhanhlinh141@gmail.com

Abstract

In the paper [1] G. Dominicis and M. Kreuzer introduced the application of Kähler differential modules to the study of 0-dimensional subschemes $\mathbf{X}$ in a projective space $\mathbf{P}^n$. They showed that this graded module over the homogeneous coordinate ring $R_{\mathbf{X}}$ contains numerical and algebraic information which is not readily available from the homogeneous vanishing ideal or from $R_{\mathbf{X}}$. Later, in the paper [2] the second author and his students extended and refined these techniques for fat point schemes in $\mathbf{P}^n$. Following the construction described in [3] it is natural to define the Kähler differential algebra $\Omega_{R_{\mathbf{X}}/K}$ of $R_{\mathbf{X}}$. In this talk, we want to take a closer look at this algebra. More precisely, by using explicit presentations of the modules $\Omega^m_{R_{\mathbf{X}}/K}$ of Kähler differential m-forms, we determine many values of their Hilbert functions explicitly and bound their Hilbert polynomials and regularity indices. Detailed results are obtained for subschemes of $\mathbf{P}^1$, fat point schemes, and fat point schemes of $\mathbf{P}^2$ supported on a non-singular conic.

References

1. G. Dominicis, M. Kreuzer, *Kähler differentials for points in* $\mathbf{P}^n$, J. Pure Appl. Algebra 141, pp. 153–173 (1999).
2. M. Kreuzer, T.N.K. Linh, L.N. Long, *Kähler differentials and Kähler differents for fat point schemes*, J. Pure Appl. Algebra 219, pp. 4479-4509 (2015).
3. E. Kunz, *Kähler Differentials*, Adv. Lectures Math., Vieweg Verlag, Braunschweig, 1986.

Specializations of symbolic polynomials

Stephen M. Watt
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada N2L 3G1
smwatt@uwaterloo.ca

Abstract

We consider "symbolic polynomials" that generalize the usual polynomials by allowing multivariate integer valued polynomials as exponents. We explore how a variety of algebraic properties specialize under the evaluation of the exponent variables.

1 Introduction

We have earlier introduced the notion of "symbolic polynomials", these being objects that are like polynomials, but allowing symbolic expressions as the exponents. For example, the expression $x^{6n}y^{m^2+m}-4$ is a symbolic polynomial. This type of expression occurs frequently in applications of symbolic computation, but computer algebra systems have typically not dealt with them particularly well. Instead of making the full specturm of algebraic algorithms available, when symbols lie in the exponents, systems tend to fall back on naïve syntactic expression manipulation. For example, the previous expression would not be recognized as a difference of squares that can be factored as $x^{6n}y^{m^2+m}-4=(x^{3n}y^{\frac{m^2+m}{2}}+2)(x^{3n}y^{\frac{m^2+m}{2}}-2)$, with the exponents $\frac{m^2+m}{2}$ always giving values in $\mathbb{N}$ when $m\in\mathbb{N}$.

In this paper, we review the basic ideas of symbolic polynomials and explore properties of evaluation mappings on exponent variables. We discuss how evaluation behaves for differential ring operations and for GCD and factorization structure. We then present some preliminary thoughts relating to Gröbner bases.

2 Symbolic Polynomials

We define symbolic polynomials as follows.

Definition 1. The ring of *symbolic polynomials in* base variables $x_1,\ldots,x_v$ *and* exponent variables $n_1,\ldots,n_p$ *over the coefficient ring* R is the ring consisting of finite sums of the form

$$\sum_i c_i x_1^{e_{i1}} x_2^{e_{i2}} \cdots x_n^{e_{in}}$$

where $c_i\in R$ and $e_{ij}\in \mathrm{Int}_{[n_1,n_2,\ldots,n_p]}(\mathbb{Z})$. Multiplication is defined by

$$c_1 x_1^{e_{11}} \cdots x_n^{e_{1n}} \quad \times \quad c_2 x_1^{e_{21}} \cdots x_n^{e_{2n}} = c_1 c_2 x_1^{e_{11}+e_{21}} \cdots x_n^{e_{1n}+e_{2n}}$$

We denote this ring $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$.

We make use of integer-valued polynomials, $\mathrm{Int}_{[n_1,\ldots n_p]}(D)$. For an integral domain D with quotient field K, univariate integer-valued polynomials, usually denoted $\mathrm{Int}(D)$, may be defined as

$$\mathrm{Int}_{[X]}(D) = \{f(X) \,|\, f(X) \in K[X] \text{ and } f(a) \in D, \text{ for all } a \in D\}$$

For example $\frac{1}{2}n^2 + \frac{1}{2}n \in \mathrm{Int}_{[n]}(\mathbb{Z})$. Integer-valued polynomials have been studied by Ostrowski [1] and Pólya [2], and we take the obvious multivariate generalization. Note that we could alternatively define symbolic polynomials as given by an algebra of terms with monomials and a finite number of ring operations.

These objects are both theoretically interesting and useful in applications of computer algebra. The usual operations of ring arithmentic and diffierential algebra ($+$, $-$, $\times$, $\partial/\partial x_i$) are straightforward. By restricting the exponents to be integer-valued polynomials, we find effective algebraic algorithms for greatest common divisor and factorization [3] and functional decomposition [4].

3 Evaluation

With 1, we have natural evaluation maps to Laurent polynomials,

$$\sigma : \mathbb{Z}^p \to R[n_1,\ldots,n_p;x_1,\ldots,x_v] \to R[x_1^{\pm}\ldots,x_v^{\pm}]$$

where $\sigma(a_1,\ldots,a_p)$ evaluates n_i at a_i. For example, $\sigma(-2,4) : \mathbb{Z}[n_1,n_2;x] \to \mathbb{Q}[x,x^{-1}]$ evaluates the symbolic polynomial $2x^{n_1^2+n_2} + x^{3n_1+n_2}$ to the Laurent polynomial $2x^8 + x^{-2}$. It is possible to work with evaluation homomorphisms that produce polynomial values in $R[x_1,\ldots,x_v]$, but this requires that the $\sigma(a_1,\ldots,a_p)$ be partial and keeping track of the domains of definition is typically more difficult that working with Laurent polynomials. Working with total evaluation maps does require, however, extending certain polynomial algorithms, see e.g. [5].

The evaluation maps are easily seen to be differential ring homomorphisms, *i.e.* when $\sigma = \sigma(b_1,\ldots,b_p)$ for any values b_i,

$$\begin{aligned}
\sigma 0 &= 0 \\
\sigma 1 &= 1 \quad \text{if } R \text{ has unity} \\
\sigma(u+v) &= \sigma u + \sigma v \\
\sigma(u \times v) &= \sigma u \times \sigma v \\
\sigma(\partial u/\partial x_i) &= \partial \sigma u/\partial x_i.
\end{aligned}$$

4 Specialization

We have seen [3] that $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ is a GCD domain if $R[x_1,\ldots,x_v]$ is a GCD domain and likewise $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ is a UFD (unique factorization domain) if $R[x_1,\ldots,x_v]$ is a UFD. Note that $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ and $R[x_1^{\pm}\ldots,x_v^{\pm}]$ have more units than $R[x_1,\ldots,x_v]$ since any monomial with unit coefficient in R is a unit in the larger rings.

The GCDs and complete factorizations in $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ do not necessarily give GCDs and complete factorizations in $R[x_1^{\pm}\ldots,x_v^{\pm}]$ under σ, but they are closely related.

Theorem 2 (Symbolic GCD Specialization). *Suppose $R[x_1,\ldots,x_v]$ is a GCD domain and $u,v\in R[n_1,\ldots,n_p;x_1,\ldots,x_v]$. Then, for all evaluation maps $\sigma=\sigma(b_1,\ldots,b_p)$,*

$$\sigma\gcd(u,v)\mid \gcd(\sigma u,\sigma v)\in R[x_1^{\pm}\ldots,x_v^{\pm}].$$

Thus the evaluation of a symbolic polynomial GCD will give *a* common divisor of the corresponding symbolic polynomials, but not necessarily the greatest common divisor. The evaluation of the symbolic polynomial GCD will, however, be maximal in the sense that (up to units) it is the "greatest" symbolic polynomial whose image divides the GCD under *all* evaluations.

A similar property holds for factorization:

Theorem 3 (Symbolic Factorization Specialization). *Suppose $R[x_1,\ldots,x_v]$ is a UFD and $u\in R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ with complete factorization*

$$u=f_1\times\cdots\times f_k.$$

Then, for all evaluation maps $\sigma=\sigma(b_1,\ldots,b_p)$,

$$\sigma f_i\mid \sigma u\in R[x_1^{\pm}\ldots,x_v^{\pm}].$$

Similarly to the case of symbolic polynomial GCDs, the evaluation of a complete factorization of a symbolic polynomial is *a* factorization of the original polynomial evaluated, but it is not necessarily a complete factorization. That is, some of the σf_i may factor further in $R[x_1^{\pm}\ldots,x_v^{\pm}]$. The evaluation of the symbolic polynomial complete factorization will, however, be the "most complete" factorization for which every factor divides the original polynomial under all evaluations.

5 Toward Gröbner Bases

A natural next topic is about the ideals of $R[n_1,\ldots,n_p;x_1,\ldots,x_v]$ and how they relate to the ideals of $R[x_1,\ldots,x_v]$. We are therefore motivated to ask whehter Gröbner bases exist for symbolic polynomials, and, if so, to explore their behaviour under specialization.

We begin by noting that the existence and construction of Gröebner bases for Laurent polynomials has been addressed earlier [6]. This work introduces a notion of generalized term orders based on conic decompositions. It finds application, for example, in computing elementary ideals of Alexander matrices [7]. Examples of generalized term orders given by [6] for monomials $x_1^{i_1} \cdot \cdots \cdot x_v^{i_v}$ use gradings such as $|i_1| + \cdots + |i_v|$, $-\min\{0, i_1, \ldots, i_v\}$, and $i_1 + \cdots + i_v - (v+1)\min\{0, i_1, \ldots, i_v\}$.

We are currently exploring the use of polynomial norms on the exponents of symbolic polynomials to give gradings on symbolic monomial ideals. Gröbner bases based on derived term orders should find useful application.

We note, though, that such term orders will not necessarily specialize under evaluation to term orders in the ring of Laurent polynomials. Consider two monomials, x^{p_1} and x^{p_2}. For different evaluation maps, we may have $p_1 < p_2$, $p_1 = p_2$ or $p_1 > p_2$ in $\mathbb{Z}$, affecting the relative order of the two monomials in any term order. An potential approach to relating term orders for symbolic polynomials to term orders for Laurent polynomials would be to compute cylindrical algebraic decompositions on the sets of exponent polynomials for each base variable. This could be used to identify regions of exponent evaluation where monomials maintain their relative order. This is an ongoing topic of investigation.

Conclusion

We have seen that many algebraic properties of symbolic polynomials are preserved completely, or in a weaker form, under evaluation of the exponent variables. For Gröbner basis computation, it is an ongoing topic of investigation to relate term orders for symbolic polynomials to term orders under evaluation of the exponents.

References

1. A. Ostrowski, Über ganzwertige Polynome in algebraischen Zahlköpern, J. Reine Angew. Math., 149 (1919), 117-124.
2. G. Pólya, Über ganzwertige Polynome in algebraischen Zahlköpern, J. Reine Angew. Math., 149 (1919), 97-116.
3. Stephen M. Watt, *Two Families of Algorithms for Symbolic Polynomials*, in Computer Algebra 2006: Latest Advances in Symbolic Algorithms—Proceedings of the Waterloo Workshop, I. Kotsireas and E. Zima (editors), World Scientific.
4. Stephen M. Watt, Functional Decomposition of Symbolic Polynomials, Proc. International Conference on Computational Science and Its Applications (ICCSA 2008), IEEE Computer Society, 193–210.
5. Stephen M. Watt, Algorithms for the Functional Decomposition of Laurent Polynomials, Proc Conferences on Intelligent Computer Mathematics 2009, Springer Verlag LNAI 5625, 186–200.
6. Franz Pauer and Andreas Unterkircher, Gröbner Bases for Ideas in Laurent Rings and their Application to Systems of Difference Equations, Applicable Algebra in Engineering, Communication and Computing, 9 (1999), 271–291.

7. Jesús Gago-Vargas, Isabel Hartillo-Hermoso and José Maria Ucha-Enríquez, Algorithmic Invariants for Alexander Modules, Proc. Computer Algebra in Symbolic Computation, Springer-Verlag LNCS 4194, 149–154.

Track 4: Design Theory

Chair: Lucia Moura (Canada) and Dimitris Simos (Austria)

Invited Speaker: Charlie Colburn

Computational and Recursive Constructions of Perfect Hash Families

Arizona State University

Abstract

A *perfect hash family* $(N;k,w,t)$ is an $N \times k$ array on w symbols, in which in every $N \times t$ subarray, at least one row consists of distinct symbols (and hence *separates* the t columns). Perfect hash families arise in combinatorial cryptography and in constructions of covering arrays; one wants to minimize the number N of rows given that the has k columns, w symbols, and *strength* t. Although direct constructions from codes, designs, finite geometries, and arithmetic sequences are known, the construction of specific s needed in applications remains challenging.

We focus on the case when the number of rows is less than the strength. First we review a clever construction method due to Blackburn. Then we generalize his method using perfect hash families with $N = t$ that are *heterogeneous* (allow differing numbers of symbols in rows) and *fractal* (for $1 \leq \rho \leq N$, every ρ rows form a perfect hash family of strength ρ). Blackburn's method constructs s of larger strength from ingredients of smaller strength. Hence direct constructions and computational searches for heterogeneous, fractal s of 'small' strength may improve upon smallest previously known sizes for 'large' strengths. We describe (1) a column exchange algorithm based on the deterministic Lovász local lemma; (2) a greedy column extension technique inspired by column exchange; and (3) a conditional expectation algorithm. Finally we produce many small ingredients and apply the generalization of Blackburn's approach, to establish numerous improvements on the smallest s previously known.

New Constant Weight Codes and Packing Numbers

I. Bluskov[1]

[1] *University of Northern British Columbia, Canada, bluskovi@unbc.ca*

The constant $A(n,d,w)$ is the maximum number of words in an (n,d,w) binary code, that is, a code of minimal distance d, with words of length n and weight w. We improve the best known lower bounds on $A(n,d,w)$ for three sets of parameters by using optimization; in particular, we show that $A(29,8,5) \geq 36$, $A(30,8,5) \geq 41$, and $A(32,8,5) = 44$ by explicitly giving the respective codes. The $(32,8,5)$ code is optimal and leads to eight more new optimal codes. We show this by improving the known result on the problem of finding the packing number $P(v,5,2)$ for $v \equiv 12 \pmod{20}$.

Kochen-Specker sets and Hadamard matrices

P. Lisoněk

Simon Fraser University, Burnaby, Canada, plisonek@sfu.ca

Kochen-Specker sets considered in this talk are pairs consisting of a finite set V of vectors in $\mathbf{C}^n$ and a list $B = (B_1, \ldots, B_k)$ of orthogonal bases of $\mathbf{C}^n$, where k is odd, $B_i \subset V$ for each i, and for each $v \in V$ the number of i such that $v \in B_i$ is even. Kochen-Specker sets are important objects in quantum mechanics. They demonstrate the contextuality of quantum mechanics, which is one of its properties that may become crucial in quantum information theory. We use generalized Hadamard matrices to construct infinite families of Kochen-Specker sets. We show that the recently discovered simplest Kochen-Specker set is the initial member of one of these infinite families. We introduce a new class of complex Hadamard matrices which have not been studied previously and we show that they can be used to construct Kochen-Specker sets.

AGC, $t-$designs and partition sets

Cristina Martínez and Alberto Besana

[1] *University of Maynooth, Ireland, {cristina.martinezramirez*@nuim.ie}

AG codes correspond geometrically to points in the Grassmannian of k-planes in an n-dimensional projective space $PG(n,\mathbb{F}_q)$ defined over a finite field $\mathbb{F}_q$ of q elements. We study the special case of cyclic codes. We prove that invariant subgrassmannians by the action of a triangle group hold a t-design of determined parameters.

References

[Go] Goppa, V. D.: Codes on algebraic curves (Russian). Dokl. Akad. Nauk. SSSR 259, 1289-1290 (1981).

[KK] R. Kötter, F. R. Kschischang, *Coding for errors and erasures in random Network Coding*, IEEE Transactions on Information Theory, Vol. 54, no. 8, 2008.

[MWS] F.J. Mac Williams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North Holland.

The Lovász Local Lemma and Variable Strength Covering Arrays

Lucia Moura[1], Sebastian Raaphorst[2], Brett Stevens[3]

[1] *Electrical Eng. and Computer Science, University of Ottawa, Canada, lucia@eecs.uottawa.ca*
[2] *Gemini Observatory, La Serena, Chile, sraaphorst@gmail.com*
[3] *Mathematics and Statistics, Carleton University, Canada, brett@math.carleton.ca*

Covering arrays are well studied combinatorial designs that are useful in software interaction testing. Here, we focus on a recent covering array generalization called variable strength covering array (VCA)[2, 3], which can be used when certain parameters of the system under test are known to not interact or to interact with different strengths. Let $H = (V(H), E(H))$ be a hypergraph and let $k = |V(H)|$. A *variable-strength covering array*, denoted $\mathrm{VCA}(n; H, v)$, is an $n \times k$ array filled from Z_v such that for any $e = \{v_0, \ldots, v_{t-1}\} \in E(H)$, the $n \times t$ subarray of columns indexed by e is *covered*, that is, it has every possible t-tuple in $(Z_v)^t$ as a row at least once. The *variable-strength covering array number*, written $\mathrm{VCAN}(H, v)$, is the smallest n such that a $\mathrm{VCA}(n; H, v)$ exists. We use the Lovász Local Lemma [1, 4], to obtain the following upper bound for the minimum size of a VCA.

Theorem 1. Let $H = (V, E)$ be a hypergraph with $\mathrm{rank}(H) = t \geq 1$, and let d be an integer such that no edge of H intersects more than d other edges of H. Then, for any $v \geq 2$, we have: $\mathrm{VCAN}(H, v) \leq \left\lceil \frac{\ln(d+1)+t\ln v+1}{\ln \frac{v^t}{v^t-1}} \right\rceil \leq \lceil v^t (\ln(d+1) + t\ln v + 1) \rceil$.

The paper focuses on comparing the upper bound given by Theorem 1, which we call the probabilistic bound, with a constructive upper bound for VCAs obtained by a density-based greedy algorithm, which we call the greedy bound, introduced in [2]. If nothing is known about the hypergraph H except the number of edges m and the rank t, then we can substitute $d \leq m - 1$, and both bounds are very close. In the rest of the article, we show several classes of hypergraphs for which we know better estimates on d and the probabilistic bound outperforms the density bound.

References

1. P. ErdℍOs and L. Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions*, in Infinite and finite sets **11**, Coll. Math. Soc. J. Bolyai, 609-627, 1975.
2. S. Raaphorst, *Variable strength covering arrays*, PhD thesis, University of Ottawa, 2012.
3. S. Raaphorst, L. Moura and B. Stevens, *Variable strength covering arrays*, submitted to J. Comb. Des., 2016.
4. J. Spencer, *Asymptotic lower bounds for Ramsey functions*, Discrete Math. **20**, 1, pp. 69-76 (1977).

Number of t-tuples in arrays from LFSRs

D. Panario[1], B. Stevens[1], G. Tzanakis[1]

[1] *Carleton University, Canada, {daniel,brett,gtzanaki}@math.carleton.ca*

Let M be an $N \times k$ array with entries from an alphabet A of cardinality v. A t-set of columns of M is *covered* if each of the v^t possible t-sets in A^t appears in at least one row of the $N \times t$ sub-array defined by these columns. If each of the $\binom{k}{t}$ possible t-sets of columns of M is covered, then M is a *covering array* of *strength* t and size N, denoted $\mathrm{CA}(N;t,k,v)$. Covering arrays are used in areas such as software development and manufacturing to test systems for which exhaustive testing is infeasible.

Let $t > 2$ be an integer, q be a prime power, $v \geq 2$ be a divisor of $q-1$, α be a primitive element of $\mathbb{F}_{q^t}$ and $\mathrm{Tr}_{\mathbb{F}_{q^t}/\mathbb{F}_q}$ be the trace of $\mathbb{F}_{q^t}$ over $\mathbb{F}_q$. Define the $(q^t-1) \times (q^t-1)$ arrays

$$L_{ij} = \mathrm{Tr}_{\mathbb{F}_{q^t}/\mathbb{F}_q}(\alpha^{i+j})$$

and

$$M_{ij} = \begin{cases} 0, & \text{if } L_{ij} = 0; \\ \log_\alpha(L_{ij}) \bmod v, & \text{otherwise.} \end{cases}$$

The authors have recently shown that if columns from M are selected based on an AMDS code and $q^{\frac{t}{2}-2}(q-tv) \geq v^{t-1}$, then the result is a strength t covering array [2]; see also [1]. This is shown using a character sum argument; evidence is presented that these bounds are likely to be weak, that is, covering arrays are produced for q substantially smaller than this bound.

In this talk, after revising the existing constructions related to linear feedback shift registers and primitive polynomials over finite fields, we determine bounds on the number of rows that contain a t-tuple.

References

1. G. Tzanakis, L. Moura, D. Panario, and B. Stevens. Constructing new covering arrays from LFSR sequences over finite fields. *Discrete Mathematics*, 339(3):1158–1171, 2016.
2. G. Tzanakis, L. Moura, D. Panario, and B. Stevens. Covering arrays from m-sequences and character sums. *Designs, Codes and Cryptography*, to appear, 2017.

Covering Arrays as Set Covers

Ludwig Kampel, Bernhard Garn, Dimitris E. Simos

SBA Research, Vienna A-1040, Austria, {lkampel, bgarn, dsimos}@sba-research.org

In this work, we review and contrast *covering array problems* with their connections to *set cover* or *integer programming* problems. We detail the correspondence between discrete structures, based on a mapping between these structures upon which also problems can be translated. Subsequently, we exemplify this connection with the pairing of corresponding problems from the covering array domain and the set cover domain. Finally, we detail common structural properties of algorithms from different domains, pointing to similarities and differences, and emphasize where results obtained in isolation in each domain can be combined.

References

1. V. Chvatal, *A greedy heuristic for the set-covering problem*, Mathematics of operations research **4**, 3, pp. 233-235 (1979).
2. N. Sloane, *Covering arrays and intersecting codes*, Journal of combinatorial designs **1**, 1, pp. 51-63 (1993).
3. D. M. Cohen and S. R. Dalal and M. L. Fredman and G. C. Patton, *The AETG system: An approach to testing based on combinatorial design*, IEEE Transactions on Software Engineering **23**, 7, pp. 437-444 (1997).
4. E. Balas and M. W. Padberg, *Set partitioning: A survey*, SIAM review **18**, 4, pp. 710-760 (1976).

Disjoint q-Steiner systems in dimension 13

Michael Braun[1], Alfred Wassermann[2]

[1] *Darmstadt University of Applied Sciences, Darmstadt, Germany, michael.braun@h-da.de*
[2] *University of Bayreuth, Bayreuth, Germany, alfred.wassermann@uni-bayreuth.de*

Let V be a vector space of dimension v over a finite field $\mathrm{GF}(q)$. For simplicity, a subspace of V of dimension k will be called a *k-subspace*.

A *(simple) t-$(v,k,\lambda)_q$ subspace design* $\mathscr{D} = (V, \mathscr{B})$ consists of a set $\mathscr{B}$ of k-subspaces of V, called blocks, such that each t-subspace of V lies in exactly λ blocks. This notion is a vector space analog of combinatorial t-designs on finite sets. For that reason, subspace designs are also called *q-analogs of designs*. In the special case of $\lambda = 1$, t-$(v,k,1)_q$ subspace designs are called *q-Steiner systems* $S(t,k,v)_q$.

We report the computer construction of 1316 mutually disjoint 2-$(13,3,1)_2$ subspace designs. By combining disjoint designs and using supplementary subspace designs we conclude that 2-$(13,3,\lambda)_2$ subspace designs exist for all possible values $1 \leq \lambda \leq 2047$.

The technical report [1] contains a list of all 1316 q-Steiner systems.

References

1. M. Braun, A. Wassermann, Disjoint q-Steiner systems in dimension 13, Tech. rep., Universität Bayreuth, Bayreuth (April 2017). http://dx.doi.org/urn:nbn:de:bvb:703-epub-3291-7.

Track 5: Natural and Quantum Computing

Chair: Mika Hirvensalo (Finland)

Invited Speaker: Lila Kari

Was Pegasus a mammal or a bird? – or – How to measure and visualize (real or synthetic) species' relatedness?

Lila Kari[1]

[1] *School of Computer Science, University of Waterloo, Canada, lila.kari@uwo.ca*

Abstract

Phylogenetic trees have been the traditional means to represent evolutionary history and species classification, but there is a growing realization that some type of graphs or networks rather than trees are often needed, to take into account phenomena such as recombination, hybridization, horizontal gene transfer, and convergent evolution. At the same time, alignment-free methods have been proposed to complement conventional morphological or sequence-alignment-based methods for phylogenetic analysis. Combining features of both these approaches, we propose Molecular Distance Maps (MoDMaps), a novel alignment-free method for computing and displaying sequence and species' relatedness. MoDMaps compute pairwise distances between Chaos Game Representations (CGR) of all input DNA sequences, and visualize the interrelationships thus obtained as an interactive map in three-dimensional Euclidean space: Each point on a map represents a DNA sequence, and the spatial proximity between any two points reflects the degree of structural similarity between the corresponding sequences.

The graphical representation of DNA sequences utilized, Chaos Game Representation, has been shown to be genome- and species-specific and can thus act as a genomic signature. Consequently, Molecular Distance Maps could inform species identification, taxonomic classifications and, to a certain extent, evolutionary history. In addition, MoDMaps is a general-purpose method that can compute and display the interrelationships within any set of sequences, biological, simulated, synthetic or computer-generated, sequences that closely related or completely unrelated, of the same length or of different lengths, several kilo-basepair-long or complete genomes. For example, this method positions a mythological Pegasus genome (part swan and part horse mitochondrial genome) halfway between the bird and the mammalian cluster.

Interference as a Computational Resource

Mika Hirvensalo[1]

[1] *Department of Mathematics and Statistics, University of Turku, Finland, mikhirve@utu.fi*

Interference is obviously familiar to anyone having watched at the waves propagating on a water surface. Sometimes the wave crests amplify each other, but sometimes the wave crest and hollow cancel each other, forming patterns which a single wave propagation never creates. This is exactly how the interference in wave propagation is understood: The waves may interfere with each other.

Quantum mechanical description of the physical world involves the idea of wave-particle dualism: All physical objects describable as particles may be portrayed as waves, as well. Applying this principle to the physical systems that bear the information and carry out the computation, it is possible to design algorithms that greatly benefit from interference: the undesired computational paths may cancel each other, but the desired ones may amplify. This phenomenon is generally believed to be the very source of the power of quantum computing.

Here we are not going to refute the aforesaid picture about the power source of quantum computing. Instead, we are going to highlight some notable interference patterns used in famous quantum algorithms, but also to point out that the phenomenon itself – interference – has been used as a computational resource even before the quantum computing era.

Resistance Analysis of Quantum Hashing

F. Ablayev[1], M. Latypov[1], A. Vasiliev[1], A. Vasilov[1]

[1] *Kazan Federal University, Russia,*
{fablayev,gogen.marat,alexander.ksu,vasilovartur}@gmail.com

1 Introduction

Recently we have defined a notion of the quantum hash function which is quantum one-way and quantum collision resistant function [1]. Quantum hash functions can be used as a quantum one-way function in the quantum digital signature protocol [2]. They also can also be used in different quantum computational models as a basis for efficient algorithms [3] and communication protocols [4]. The further generalizations can be used for constructing quantum hash-based message authentication codes [5].

We have analyzed the key properties of quantum hash functions and shown that one-way property and collision resistance property are correlated for a quantum hash function [6]. The more the function is one-way the less it is collision resistant and vice versa. We showed that such a correlation can be balanced.

In [5], [7] we have presented an approach for constructing quantum hash functions by establishing a connection with small biased sets [8]: we prove that small sized ε-biased sets allow to generate balanced quantum hash functions.

In this paper we investigate the pre-image resistance of this function. Previously, we have proved the bound on the amount of accessible information about the input using the well-known Holevo theorem [9]. Since no more than $O(s)$ classical bits of information can be extracted from s qubits and the original message contains $n \gg s$ bits, it is impossible to restore the input from the quantum hash. However, using the results of [10] and the properties of ε-biased sets here we show that the quantum hash function reveals only $O(1)$ bits of information about the input.

Additionally, we use several heuristic algorithms to explicitly construct ε-biased sets of a certain size, thus supporting the collision resistance of our hash function.

2 Preliminaries

In this section we recall the notion of the quantum hashing, as well as the definition of the small-bias sets used in its construction.

2.1 Small-bias Sets

The construction of a quantum hash function in this paper relies on the notion of the ε-biased sets. We use the definition given in [11].

Let G be a finite abelian group and let χ_a be the characters of G, indexed by $a \in G$.

Definition 1. A set $S = \{s_1, s_2, \ldots, s_d\} \subseteq G$ is called ε-biased, if for any nontrivial character χ_a

$$\frac{1}{|S|}\left|\sum_{j=1}^{|S|}\chi_a(s_j)\right| \leq \varepsilon. \tag{1}$$

It follows from the Alon–Roichman theorem [12] that a set S of $O(\log|G|/\varepsilon^2)$ elements selected uniformly at random from G is ε-biased with nonzero probability.

2.2 Quantum Hashing for Finite Abelian Groups

In [7] we have proposed the notion of a quantum hash function, which is defined for arbitrary finite abelian groups.

Let G be a finite abelian group with characters χ_a, indexed by $a \in G$. Let $S \subseteq G$ be an ε-biased set for some $\varepsilon \in (0,1)$.

Definition 2. We define a quantum hash function $\psi_S : G \to (\mathscr{H}^2)^{\otimes \log|S|}$ as following:

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}}\sum_{j=1}^{|S|}\chi_a(s_j)|j\rangle. \tag{2}$$

We have shown that ψ_S has all the properties of a cryptographic quantum hash function (i.e. it is quantum one-way and collision resistant), which are entirely determined by the ε-biased set $S \subseteq G$.

We also note, that the size of the quantum hash above is asymptotically optimal because of the known lower bound by Buhrman et al. [13] for the size of the sets of pairwise-distinguishable states: to construct a set of 2^k quantum states with pairwise inner products below ε one will need at least $\Omega(\log(k/\varepsilon))$ qubits. This implies the lower bound on the size of quantum hash of $\Omega(\log\log|G| - \log\varepsilon)$.

There are two known special cases of the quantum hashing for specific finite abelian groups. In particular, we are interested in hashing binary strings and thus it is natural to consider $G = \mathrm{Z}_2^n$ and $G = \mathrm{Z}_{2^n}$ (or, more generally, any cyclic group Z_q).

2.3 Hashing the Elements of the Boolean Cube

For $G = Z_2^n$ its characters can be written in the form $\chi_a(x) = (-1)^{(a,x)}$, and quantum hash function is the following

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}} \sum_{j=1}^{|S|} (-1)^{(a,s_j)} |j\rangle. \tag{3}$$

The resulting hash function is exactly the quantum fingerprinting by Buhrman et al. [13], once we consider an error-correcting code, whose matrix is built from the elements of S.

2.4 Hashing the Elements of the Cyclic Group

For $G = Z_q$ its characters can be written as $\chi_a(x) = \exp(2\pi i a x/q)$, and quantum hash function is given by

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}} \sum_{j=1}^{|S|} e^{\frac{2\pi i a s_j}{q}} |j\rangle. \tag{4}$$

The above quantum hash function is essentially equivalent to the one we have defined earlier in [1], which is in turn based on the quantum fingerprinting function from [14].

3 Pre-image Resistance of Quantum Hashing

In this section we analyze the quantum hash function defined above and prove it has a strong pre-image resistance.

In [10] authors defined a quantum scheme which is based on quasi-linear codes and maps binary strings to a quantum state. If a scheme uses pure states, accessible information does not exceed $O(1)$ bits. We prove similar properties of a general quantum hash function ψ_S for an arbitrary finite abelian group G and its ε-biased subset $S \subset G$.

For $a \in G$ we denote density operator of normalized state $\rho_a = |\psi_S(a)\rangle\langle\psi_S(a)|$ and non-normalized state $\rho'_a = \frac{2^d}{|G|}\rho_a$. Furthermore, for any $|v\rangle \in \mathscr{H}^{2^d}$ we define a probability distribution $\mu_v(a) = \langle v|\rho'_a|v\rangle$, that corresponds to measurement with outcome $|v\rangle\langle v|$.

The following lemma allows us to estimate the relative entropy between $\mu_v(a) = \langle v|\rho'_a|v\rangle$ and uniform probability distribution over G.

Lemma 1. *Let $|v\rangle \in \mathscr{H}^{2^d}$ be a unit vector and $a \in G$ is randomly chosen according to uniform distribution. Then*

$$\mathbf{E}[\max\{0, \mu_v(a)\ln(|G|\mu_v(a))\}] < \frac{23}{|G|}. \tag{5}$$

Proof. For all $s \in S$ we define random variables

$$X_s = \chi_a(s)v_s \tag{6}$$

Then $\mu_v(a) = \frac{1}{|G|}(\sum_{s\in S} X_s)^2$. $\mathbf{E}[\chi_a(s)] = 0$ and $|\chi_a(s)| \leq 1$ follows from the properties of finite abelian group characters. Then for all $t > 0$

$$\Pr\left[\mu_v(a) \geq \frac{t}{|G|}\right] = \Pr\left[\left|\sum_{s\in S} X_s\right| \geq \left|\sqrt{t}\right|\right] \leq 4\exp\left(-\frac{t}{4}\right), \tag{7}$$

where the last inequality follows from Lemma 2.2 from [10] and from $\|v\| = 1$.

Define $g(x) = \max\{0, x\ln(x)\}$ and let $\tilde{\mu}$ be a random variable, whose probability distribution is $\Pr[\tilde{\mu} \geq t] = 4\exp(-\frac{t}{4}) = f(t)$ for $t > 8\ln 2$. Then

$$\mathbf{E}[\max\{0, \mu_v(a)\ln(|G|\mu_v(a))\}] \geq \frac{1}{|G|}\mathbf{E}[g(|G|\mu_v(a))] \geq \frac{1}{|G|}\mathbf{E}[g(\tilde{\mu})], \tag{8}$$

where the first inequality follows from the definition of $g(x)$ and the second one is true by Lemma 2.3 [10].

Therefore,

$$\mathbf{E}[g(\tilde{\mu})] = \int_{8\ln 2}^{\infty} x\ln(x)\left(-\frac{df}{dx}\right)dx = \int_{8\ln 2}^{\infty} \exp\left(\ln(x) + \ln(\ln(x)) - \frac{x}{4}\right)dx < 23, \tag{9}$$

as required.

Definition 3. For random variables P and Q having a discrete probability distribution the Kullback-Leibler divergence is given as follows

$$D_{KL}(P \parallel Q) = \sum_i P(i)\ln\frac{P(i)}{Q(i)}. \tag{10}$$

The following lemma shows that if we use ε-biased sets in our scheme, divergence between $\mu_v(a)$ and a random variable x uniformly distributed over G is given by $D_{KL}(\mu_v||x)$ and takes small values.

Lemma 2. *Let $|v\rangle \in \mathscr{H}^{2^d}$ be a unit vector. Then*

$$\sum_{a\in G} \mu_v(a)\ln(|G|\mu_v(a)) < 23. \tag{11}$$

Proof. We define a random variable

$$\tilde{\mu}(a) = \max\{0, \mu_v(a)\ln(|G|\mu_v(a))\}. \quad (12)$$

By Lemma 1 $\mathbf{E}[\tilde{\mu}(a)] < \frac{23}{|G|}$. *Therefore,*

$$\sum_{a\in G} \mu_v(a)\ln(|G|\mu_v(a)) < \sum_{a\in G} \tilde{\mu}(a) = |G|\mathbf{E}[\tilde{\mu}(a)] < 23. \quad (13)$$

In [10] the accessible information I_{acc} about input was considered based on the measurement of the quantum state representing this input. It was defined as $I_{acc} = H(J) - H(J|A)$, where A is a random variable describing the choice of input data, J is a random variable that describes the result of measuring the quantum state.

Lemma 3. *Let a be chosen randomly according to uniform distribution over G, then accessible information* I_{acc} *of ensemble* (ρ_a) *does not exceed*

$$\max_{|v\rangle} \sum_{a\in G} \mu_v(a)\ln(|G|\mu_v(a)) < 23. \quad (14)$$

This lemma rephrases the Lemma 3.12 from [10] with using ε-biased set over finite abelian group and is given without proof.

Thus, the above statements prove the follow theorem.

Theorem 1. *Let* $S \subset G$ *be an* ε*-biased set,* ψ_S *be a quantum hash function based on S. Then the amount of accessible information about pre-image of* ψ_S *is of order* $O(1)$.

4 Explicit Constructions of ε-biased Sets

We know that for any $\varepsilon \in (0,1)$ there exists an ε-biased set over group G of size $O(\log|G|/\varepsilon^2)$. But there is no explicit algorithm to construct such a set. Existing explicit constructions give asymptotically bigger sets (see [15] for $G = Z_2^n$ and [16] for $G = Z_q$). Moreover, these algorithms give asymptotically good solutions, which means they are applicable for sufficiently large inputs (the resulting quantum hash function would still be collision resistant, but not pre-image resistant due to the hash size). If we want to create a cryptographic quantum hash function, we must solve this problem. Here we propose to use stochastic optimization algorithms.

4.1 Random Search Algorithms

It is easy to see, that exhaustive search is not applicable in practice. That is why we can use the random search [17] instead. We can generate a random set and check if it is ε-biased, i.e. the resulting quantum hash function is ε-resistant to quantum collisions. If ε is not very small, this algorithm can possibly find a good solution

quite fast. But of course there is no guarantee for that. In fact, if we want the optimal solution, the random search will be even worse than exhaustive search. So, we need to optimize the random search.

In the naive solution above we don't use the results of the previous iterations. But if we do, we can use the function evaluations in some points to select the next point according to some algorithm.

Improved random search.

Here is a small improvement of the random search algorithm [18]:

1. Select a random set from the search space.
2. On each iteration of the algorithm we evaluate the bias of the set. After that we create a hypersphere with fixed radius. We select a new point (a set) from this sphere. Then:
   a. If the new set is better (is less biased) then it becomes a new candidate for solution, and we repeat the step 2 again.
   b. Otherwise we select a new point on the sphere.
3. This algorithm continues until the good enough result is achieved or the number of iterations is exceeded.

This algorithm is more effective and it is rather suitable for the purpose of global optimization. But from the construction of the algorithm we can easily see that it could get stuck in local optima (since sphere has a fixed size). And there are many ways to solve this problem, e.g. an *adaptive random search algorithm* [19].

Adaptive random search.

Here we increase the size of a step (if needed) instead of fixing it. The algorithm description may look like this:

1. Select a random set from the search space.
2. On each iteration of algorithm we evaluate the bias of the set. After that we create two hyperspheres: the first one with the current radius r, the second one with r', where $r' > r$. On each hypersphere we select a point and if the point from the second sphere improves the current result more than the first one then we increase the current radius: $r = r'$.
3. In other cases the algorithm is the same as the previous one.

There is one more popular variation of the random search algorithm – a *greedy adaptive random search algorithm* [20].

Greedy adaptive random search.

This algorithm is based on two following steps:

1. Using any greedy algorithm we select a set of candidates for solution.
2. With local search [21] we find the best solution from this set.

Brownlee [22] provides more detailed description with possible implementation.

It is worth noting, that all the described random search algorithms are not good enough in finding the global optima.

4.2 Iterated Local Search

Many random search algorithms use the local search procedure [21]. This procedure can also be used for optimization problems, but it may not find the global optima and get stuck in local optima. There are some techniques that help solving this problem, one of them is called *iterated local search algorithm*.

The problem of getting stuck in local optima implies that the best local value may be far from the optimal one and there is no neighbor to continue the search. That is why iterated local search uses a multi-restart paradigm. Whenever the algorithm stops at some point and this point is clearly not the best one, a new iteration of algorithm is (re-)started from a point found with large enough perturbation or even from any random point [22]. For the local search we can implement any algorithm, usually it is some heuristic.

We can describe iterated local search procedure as following:

1. Set any point as the current best value. It could be a random point or point from some explicit construction or heuristic.
2. Create a new set of points using perturbation of current best point. Perturbations for points should be rather different.
3. Use the local search procedure for set from step 2 and find the best value.
4. If the value found on step 3 is better than the current best value and it is acceptable, update the current best value.

Algorithm of perturbation can be chosen arbitrarily, but it is better to use some adaptive algorithm. The same is true for the acceptance criteria (used in step 4). The algorithm of perturbation and acceptance criteria are the procedures that prevent getting stuck in local optima.

4.3 Summary of Stochastic Algorithms

Random search algorithms work well enough and at the same time their implementation is very simple. They don't require any assumptions about the input data, problem size or even result format.

However, these algorithms are not the optimal ones in terms of effectively finding solution. They may be used in low-dimensional problems or in very large problems, where no other algorithm can possibly find the solution. We can also use the random search algorithms for some initial tests or to find the start point for another heuristic algorithms.

It is also important, that random search algorithms are independent and can be executed in parallel from different start points or in different parts of the search space. So, there are a lot of ways for local optimization.

And we have one more benefit of random algorithms – they could be applied to any function. Since there is no substantial difference between generating random numbers and random n-bits strings, we can use random search algorithms for quantum hashing over Z_2^n, Z_q, or any other appropriately encoded finite abelian group.

Acknowledgments.

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University. Work was in part supported by the Russian Foundation for Basic Research (under the grant 17-07-01606).

References

1. F M Ablayev and A V Vasiliev. Cryptographic quantum hashing. *Laser Physics Letters*, 11(2):025202, 2014.
2. Daniel Gottesman and Isaac Chuang. Quantum digital signatures. Technical Report arXiv:quant-ph/0105032, Cornell University Library, Nov 2001.
3. Farid Ablayev and Alexander Vasiliev. Computing Boolean Functions via Quantum Hashing. In Cristian S Calude, Rusins Freivalds, and Iwama Kazuo, editors, *Computing with New Resources*, Lecture Notes in Computer Science, pages 149–160. Springer International Publishing, 2014.
4. Alexander Vasiliev. Quantum communications based on quantum hashing. *International Journal of Applied Engineering Research*, 10(12):31415–31426, 2015.
5. Farid Ablayev, Marat Ablayev, Alexander Vasiliev, and Mansur Ziatdinov. Quantum fingerprinting and quantum hashing. computational and cryptographical aspects. *Baltic Journal of Modern Computing*, 4(4):860–875, 2016.
6. F Ablayev, M Ablayev, and A Vasiliev. On the balanced quantum hashing. *Journal of Physics: Conference Series*, 681(1):012019, 2016.
7. Alexander Vasiliev. Quantum hashing for finite abelian groups. *Lobachevskii Journal of Mathematics*, 37(6):751–754, 2016.

8. Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 213–223, New York, NY, USA, 1990. ACM.
9. Alexander S. Holevo. Some estimates of the information transmitted by quantum communication channel (russian). *Probl. Pered. Inform. [Probl. Inf. Transm.]*, 9(3):3–11, 1973.
10. Dmitry Gavinsky and Tsuyoshi Ito. Quantum fingerprints that keep secrets. Technical report, 2010.
11. Sixia Chen, Cristopher Moore, and Alexander Russell. Small-bias sets for nonabelian groups. In Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and Jose D.P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 8096 of *Lecture Notes in Computer Science*, pages 436–451. Springer Berlin Heidelberg, 2013.
12. Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Structures & Algorithms*, 5(2):271–284, 1994.
13. Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87(16):167902, Sep 2001.
14. Farid Ablayev and Alexander Vasiliev. Algorithms for quantum branching programs based on fingerprinting. *Electronic Proceedings in Theoretical Computer Science*, 9:1–11, 2009.
15. A. Ben-Aroya and A. Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 191–197, Oct 2009.
16. Alexander A. Razborov, Endre Szemeredi, and Avi Wigderson. Constructing small sets that are uniform in arithmetic progressions. *Combinatorics, Probability & Computing*, 2:513–518, 1993.
17. Roger J-B. Wets Francisco J. Solis. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.
18. Michael A. Schumer and Kenneth Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.
19. Zelda B. Zabinsky. *Stochastic adaptive search for global optimization*, volume 72 of *Nonconvex optimization and its applications*. Springer, 2003.
20. Thomas A. Feo and Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.
21. Holger H. Hoos and Thomas Stutzle. *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
22. Jason Brownlee. *Clever Algorithms: Nature-Inspired Programming Recipes*. Lulu.com, 1st edition, 2011.

Branching Program Complexity of Quantum Hashing

F. Ablayev[1], M. Ablayev[2],

[1] *Kazan Federal University, Russia, fablayev@gmail.com*
[2] *Kazan Federal University, Russia, mablayev@gmail.com*

Abstract

We investigate Branching Program complexity measure of quantum hashing. Quantum (δ,ε)-hash function $\psi : \mathbb{F}_q \to (\mathscr{H}^2)^{\otimes s}$ hashes elements of finite field $\mathbb{F}_q$ into s-qubit quantum states. This function is one-way δ-resistant and is collision ε-resistant.

We consider two complexity measures for Quantum Branching Program (QBP): a number $Width(Q)$ of QBP Q qubits and a number Time(Q) of QBP Q computational steps. We show that quantum (δ,ε)-hash function can be computed effectively. Namely, we present QBP Q for quantum (δ,ε)-hash function with the following complexity characteristics: $Width(Q) = O(\log\log q)$ and $Time(Q) = \log q$.

We prove that such QBP construction is optimal. That is, we prove lower bounds $\Omega(\log\log q)$ of QBP width and $\Omega(\log q)$ of QBP time for quantum (δ,ε)-hash function computation.

1 Introduction

In [2] we explicitly defined a notion of quantum hashing as a generalization of classical hashing and presented examples of quantum hash functions. It appeared that Gottesman-Chuang quantum signature schemes [4] are based on functions which are actually quantum hash functions. Those functions have "unconditionally one-way" property based on Holevo Theorem [5]. More information on the role of quantum hashing for the post quantum cryptography, possible application of quantum hashing for quantum signature protocols, and technological expectations for realization of quantum signature schemes are presented in [6].

Recall that in the classical setting a cryptographic hash function h should be computed effectively and should have the following properties (see for example [7]). (1) Pre-image resistance: Given $h(x)$, it should be difficult to find x, that is, these hash functions are one-way functions. (2) Second pre-image resistance: Given x_1, it should be difficult to find an x_2, such that $h(x_1) = h(x_2)$. (3) Collision resistance: It should be difficult to find any pair of distinct x_1, x_2, such that $h(x_1) = h(x_2)$. Note, that there are no one-way functions that are known to be provably more difficult to invert than to compute, the security of cryptographic hash functions is "computationally conditional".

In the paper we consider quantum (δ,ε)-hash functions construction based on ε-biased sets. Such quantum (δ,ε)-hash function $\psi : \mathbb{F}_q \to (\mathscr{H}^2)^{\otimes s}$ hashes elements of finite field $\mathbb{F}_q$ into s-qubit quantum states. The notion of (δ,ε)-hash function

combines together a notion of pre-image (one-way) quantum δ-resistance property and the notion of quantum collision ε-resistance properties. These properties are quantum generalization of classical one-way resistance and collision resistance properties required for classical hash functions.

Important property for hash function is a computational effectiveness. In the paper we show that considered construction of quantum (δ,ε)-hash function is computed effectively in the model of Quantum Branching Programs [3]. We consider two complexity measures: a number $Width(Q)$ of qubits that QBP Q uses for computation and a number Time(Q) of computational steps of QBP Q. Such QBP Q is of $Width(Q) = O(\log\log q)$ and $Time(Q) = \log q$.

We prove that such QBP construction is optimal. That is, we prove lower bounds $\Omega(\log\log q)$ for QBP width and $\Omega(\log q)$ for QBP time for quantum (δ,ε)-hash function presentation.

2 Preliminaries

Recall that mathematically a qubit is described as a unit vector in the two-dimensional Hilbert complex space $\mathscr{H}^2$. Let $s \geq 1$. Let $(\mathscr{H}^2)^{\otimes s}$ be the 2^s-dimensional Hilbert space, describing the states of s qubits. For an integer $j \in \{0,\ldots,2^s-1\}$ let $\sigma = \sigma_1 \ldots \sigma_s$ be a binary presentation of j. We use (as usual) notations $|j\rangle$ and $|\sigma\rangle$ to denote quantum state $|\sigma_1\rangle \cdots |\sigma_s\rangle = |\sigma_1\rangle \otimes \cdots \otimes |\sigma_s\rangle$.

We let q to be a prime and $\mathbb{F}_q$ be a finite field of order q. Let Σ^k be a set of words of length k over a finite alphabet Σ. Let $\mathbb{X}$ be a finite set. In the paper we let $\mathbb{X} = \Sigma^k$, or $\mathbb{X} = \mathbb{F}_q$. In the last case we will also consider that $\mathbb{X}$ is a set $\{0,1\}^k$ of binary sequences of length $k = \log q$.

We define classical-quantum (or just quantum) function ψ to be a function

$$\psi : \mathbb{X} \to (\mathscr{H}^2)^{\otimes s}.$$

One-way resistance.

We present the following definition of quantum one-way δ-resistant function. Let "information extracting" mechanism $\mathscr{M}$ be a function $\mathscr{M} : (\mathscr{H}^2)^{\otimes s} \to \mathbb{X}$. Informally speaking mechanism $\mathscr{M}$ makes some measurement to state $|\psi\rangle \in (\mathscr{H}^2)^{\otimes s}$ and decode the result of measurement to $\mathbb{X}$.

Definition 4. Let X be random variable distributed over $\mathbb{X}$ $\{Pr[X = w] : w \in \mathbb{X}\}$. Let $\psi : \mathbb{X} \to (\mathscr{H}^2)^{\otimes s}$ be a quantum function. Let Y is any random variable over $\mathbb{X}$ obtained by some mechanism $\mathscr{M}$ making measurement to the encoding ψ of X and decoding the result of measurement to $\mathbb{X}$. Let $\delta > 0$. We call a quantum function ψ a one-way δ-resistant function if for any mechanism $\mathscr{M}$, the probability $Pr[Y = X]$ that $\mathscr{M}$ successfully decodes Y is bounded by δ

$$Pr[Y = X] \leq \delta.$$

For the cryptographic purposes it is natural to expect (and we do this in the rest of the paper) that random variable X is uniformly distributed.

A quantum state of $s \geq 1$ qubits can "carry" an infinite amount of information. On the other hand, fundamental result of quantum informatics known as Holevo's Theorem [5] states that a quantum measurement can only give s bits of information about the state. We will use here the following particular version [10] of Holevo's Theorem.

Property 1. Let X be random variable uniformly distributed over a k bit binary words $\{0,1\}^k$. Let $\psi : \{0,1\}^k \to (\mathscr{H}^2)^{\otimes s}$ be a quantum function. Let Y be a random variable over $\mathbb{X}$ obtained by some mechanism $\mathscr{M}$ making some measurement of the encoding ψ of X and decoding the result of measurement to $\{0,1\}^k$. Then our probability of correct decoding is given by

$$Pr[Y = X] \leq \frac{2^s}{2^k}.$$

Collision resistance.

The following definition was presented in [1].

Definition 5. Let $\delta > 0$. We call a quantum function $\psi : \mathbb{X} \to (\mathscr{H}^2)^{\otimes s}$ a collision ε-resistant function if for any pair w, w' of different elements,

$$\left|\langle \psi(w) \,|\, \psi(w')\rangle\right| \leq \varepsilon.$$

Note that the above inequality means almost orthogonality (ε orthogonality) of quantum states $|\psi(w)\rangle$ and $|\psi(w')\rangle$. Well known that orthogonality of quantum states provides distinguishability of these states. In content of collision notion almost orthogonality means good collision resistant property. That is, let us denote $Pr_{\mathscr{M}}[v = w]$ a probability that some test $\mathscr{M}$ having quantum hashes $|\psi(v)\rangle$ and $|\psi(w)\rangle$ outputs the result "$v = w$" (outputs the result "$|\psi(v)\rangle = |\psi(w)\rangle$"). For example, known *SWAP*-test [4] provides

$$Pr_{swap}[v = w] \leq \frac{1}{2}(1 + \varepsilon^2).$$

The *REVERSE*-test [4, 1] provides

$$Pr_{reverse}[v = w] \leq \varepsilon^2.$$

The above two definitions and considerations lead to the following formalization of the quantum cryptographic (one-way and collision resistant) function

Definition 6. Let $K = |\mathbb{X}|$ and $s \geq 1$. Let $\delta > 0$ and $\varepsilon > 0$. We call a function $\psi : \mathbb{X} \to (\mathscr{H}^2)^{\otimes s}$ a quantum (δ, ε)-hash function if ψ is a one-way δ-resistant and is a collision ε-resistant function.

1 Computing a quantum hash $|\psi_S(x)\rangle$ by QBP

Quantum hash functions construction via small-biased sets.

For an $a \in \mathbb{F}_q$ a character χ_a of $\mathbb{F}_q$ is a homomorphism $\chi_a : \mathbb{F}_q \to \mu_q$, where μ_q is the (multiplicative) group of complex q-th roots of unity, $\chi_a(x) = \omega^{ax}$. Here $\omega = e^{\frac{2\pi i}{q}}$ is a primitive complex qth root of unity. A character $\chi_0 \equiv 1$ is called a trivial character.

- A set $S \subseteq \mathbb{F}_q$ is called ε-biased, if for any nontrivial character $\chi \in \{\chi_a : a \in \mathbb{F}_q\}$ $\frac{1}{|S|} |\sum_{x \in S} \chi(x)| \leq \varepsilon$.

We present the result of [11] in the following form.

Property 2. Let $S \subseteq \mathbb{F}_q$ be an ε-biased set. Let $H_S = \{h_a(x) = ax \pmod q, \quad a \in S\}$. Then a quantum function $\psi_S : \mathbb{F}_q \to (\mathscr{H}^2)^{\otimes \log |S|}$

$$|\psi_{H_S}(x)\rangle = \frac{1}{\sqrt{|S|}} \sum_{a \in S} \omega^{h_a(x)} |a\rangle$$

is quantum (δ, ε)-hash function, where $\delta \leq |S|/(q \log q)$.

- In the content of the definition of quantum hash generator [1] and the above consideration it is natural to call the set H_S of functions (formed from ε-biased set S) a *uniform quantum* (δ, ε)*-hash generator* for $\delta = O(|S|/(q \log q))$.

Note that ε-biased sets are interesting when $|S| \ll |\mathbb{F}_q|$ (as $S = \mathbb{F}_q$ is 0-biased). The seminal paper of Naor and Naor [9] defined these small-biased sets, gave the first explicit constructions of such sets, and demonstrated the power of small-biased sets for several applications.

- Note that a set S of $O(\log q/\varepsilon^2)$ elements selected uniformly at random from $\mathbb{F}_q$ is ε-biased with a positive probability > 0 [8].

Many other constructions of small-biased sets followed during the last decades (see for example [8]).

As a corollary from Property 2 and the above consideration we can state the following.

Property 3. For a small size ε-biased set $S = \{a_1, \ldots, a_T\} \subset \mathbb{F}_q$ with $T = O(\log q/\varepsilon^2)$, for $s = \log T$, for $\delta = O(1/(q\varepsilon^2))$ a quantum uniform (δ, ε)-hash generator H_S generates quantum (δ, ε)-hash function

$$\psi_{H_S} \ : \ \mathbb{F}_q \to (\mathscr{H}^2)^{\otimes s} \tag{15}$$

$$\left|\psi_{H_S}(x)\right\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} \omega^{a_j x} |j\rangle. \tag{16}$$

QBP for quantum hash function ψ_{H_S}.

We use a QBP model defined in [3].

A Quantum Branching Program Q over the Hilbert space $(\mathscr{H}^2)^{\otimes s}$ is defined as

$$Q = \langle T, |\psi_0\rangle\rangle,$$

where T is a sequence of l instructions: $T_j = \left(x_{i_j}, U_j(0), U_j(1)\right)$ is determined by the variable x_{i_j} tested on the step j, and $U_j(0)$, $U_j(1)$ are unitary transformations in $(\mathscr{H}^2)^{\otimes s}$.

Vectors $|\psi\rangle \in (\mathscr{H}^2)^{\otimes s}$ are called states (state vectors) of Q, $|\psi_0\rangle \in (\mathscr{H}^2)^{\otimes s}$ is the initial state of Q.

We define a computation of Q on an input $\sigma = \sigma_1 \ldots \sigma_n \in \{0,1\}^n$ as follows:

1. A computation of Q starts from the initial state $|\psi_0\rangle$;
2. The j-th instruction of Q reads the input symbol σ_{i_j} (the value of x_{i_j}) and applies the transition matrix $U_j = U_j(\sigma_{i_j})$ to the current state $|\psi\rangle$ to obtain the state $|\psi'\rangle = U_j(\sigma_{i_j})|\psi\rangle$;
3. The final state is
$$|\psi(\sigma)\rangle = \left(\prod_{j=l}^{1} U_j(\sigma_{i_j})\right) |\psi_0\rangle.$$

Theorem 2. *Quantum (δ, ε)-hash function (15)*

$$\psi_{H_S} : \mathbb{F}_q \to (\mathscr{H}^2)^{\otimes s}$$

can be computed by quantum branching program Q composed from $s = O(\log\log q)$ qubits in $\log q$ steps.

Proof. Quantum function ψ_{H_S} (15) for an input $x \in \mathbb{F}_q$ determines quantum states (16)

$$\left|\psi_{H_S}(x)\right\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} \omega^{a_j x} |j\rangle$$

which is a result of quantum Fourier transformation (QFT) of the initial state

$$|\psi_0\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} |j\rangle.$$

Such a QFT is controlled by the input x. QBP Q for for computing quantum hash $\left|\psi_{H_S}(x)\right\rangle$ determined as follows. We represent an integer $x \in \{0, \ldots, q-1\}$ as the

bit-string $x = x_0 \dots x_{\log q-1}$ that is, $x = x_0 + 2^1 x_1 + \dots + 2^{\log q-1} x_{\log q-1}$. For a binary string $x = x_0 \dots x_{\log q-1}$ a Quantum Branching Program Q over the space $(\mathscr{H}^2)^{\otimes s}$ for computing $|\psi_S(x)\rangle$ (composed of $s = \log T$ qubits) is defined as

$$Q = \langle \mathbb{T}, |\psi_0\rangle \rangle,$$

where $|\psi_0\rangle$ is the initial state and $\mathbb{T}$ is a sequence of $\log q$ instructions:

$$\mathbb{T}_j = (x_j, U_j(0), U_j(1))$$

is determined by the variable x_j tested on the step j, and $U_j(0)$, $U_j(1)$ are unitary transformations in $(\mathscr{H}^2)^{\otimes s}$. More precise $U_j(0)$ is $T \times T$ identity matrix. $U_j(1)$ is the $T \times T$ diagonal matrix whose diagonal entries are $\omega^{a_0 2^j}, \omega^{a_1 2^j}, \dots, \omega^{a_{T-1} 2^j}$ and the off-diagonal elements are all zero. That is,

$$U_j(1) = \begin{bmatrix} \omega^{a_0 2^j} & & & \\ & \omega^{a_1 2^j} & & \\ & & \ddots & \\ & & & \omega^{a_{T-1} 2^j} \end{bmatrix}.$$

We define a computation of Q on an input $x = x_0 \dots x_{\log q-1} \in \{0,1\}^{\log q}$ as follows:

1. A computation of Q starts from the initial state $|\psi_0\rangle$;
2. The j-th instruction of Q reads the input symbol x_j (the value of x) and applies the transition matrix $U_j(x_j)$ to the current state $|\psi\rangle$ to obtain the state $|\psi'\rangle = U_j(x_j)|\psi\rangle$;
3. The final state is
$$|\psi_S(x)\rangle = \left(\prod_{j=0}^{\log q-1} U_j(x_j) \right) |\psi_0\rangle.$$

□

Consider the following notations. For the QBP Q from Theorem 2 we let $Width(Q) = s$ and $Time(Q) = |\mathbb{T}|$. Next for quantum hash function ψ_{H_S} (15) we let

$$Width(\psi_{H_S}) = \min Width(Q), \qquad Time(\psi_{H_S}) = \min Time(Q)$$

where minimum is taken over all QBPs that compute ψ_{H_S}. Then from Theorem 2 we have

$$Width(\psi_{H_S}) = O(\log \log q), \tag{17}$$

$$Time(\psi_{H_S}) = O(\log q). \tag{18}$$

Lower bounds.

We present here the following

Theorem 3.

$$Width(\psi_{H_S}) = \Omega(\log\log q), \tag{19}$$
$$Time(\psi_{H_S}) = \Omega(\log q). \tag{20}$$

QBP Q is a procedure for the function ψ_{H_S} computation. ψ_{H_S} can be presented as follows

$$\psi_{H_S} : \{|\psi_0\rangle\} \times \{0,1\}^{\log q} \to (\mathscr{H}^2)^{\otimes s}.$$

The proof of the lower bound (19) is the immediate corollary from the following statement [1]. We present its proof for completeness.

Lemma 4. *Let $\psi : \mathbb{X} \to (\mathscr{H}^2)^{\otimes s}$ be a collision ε-resistant function. Then*

$$s \geq \log\log|\mathbb{X}| - \log\log\left(1+\sqrt{2/(1-\varepsilon)}\right) - 1.$$

Proof. First we observe, that from the definition $|||\psi\rangle|| = \sqrt{\langle\psi\,|\,\psi\rangle}$ of the norm it follows that

$$|||\psi\rangle - |\psi'\rangle||^2 = |||\psi\rangle||^2 + |||\psi'\rangle||^2 - 2\langle\psi\,|\,\psi'\rangle.$$

Hence for arbitrary pair w, w' of different elements from $\mathbb{X}$ we have that

$$|||\psi(w)\rangle - |\psi(w')\rangle|| \geq \sqrt{2(1-\varepsilon)} \tag{21}$$

We let $\Delta = \sqrt{2(1-\varepsilon)}$. For short we let $(\mathscr{H}^2)^{\otimes s} = V$ in this proof. Consider a set $\Phi = \{|\psi(w)\rangle : w \in \mathbb{X}\}$. If we draw a sphere of the radius $\Delta/2$ with the center $|\psi\rangle \in \Phi$ then all such spheres do not intersect pairwise. All these K ($K = |\mathbb{X}|$) spheres are in large sphere of radius $1+\Delta/2$. The volume of a sphere of a radius r in V is $cr^{2^{s+1}}$ for the complex space V. Constant c depends on the metric of V. From this we have, that the number K is bonded by the number of "small spheres" in the "large sphere"

$$K \leq \frac{c(1+\Delta/2)^{2^{s+1}}}{c(\Delta/2)^{2^{s+1}}}.$$

Hence

$$s \geq \log\log K - \log\log\left(1+\sqrt{2/(1-\varepsilon)}\right) - 1.$$

□

The proof of the lower bound (20) for $Time(\psi_{H_S})$ follows from the proof of Lemma 4. The assumption that QBP Q for ψ_{H_S} can test less than $\log q$ (not all $\log q$) variables of inputs $x \in \mathbb{F}_q$ means existence of (at least) two different inputs

$w, w' \in \mathbb{F}_q$ such that Q produces the same quantum hashes for w and w', that is, $|\psi(w)\rangle = |\psi(w')\rangle = |\psi\rangle$. The last contradicts (21).

References

1. F. Ablayev and M. Ablayev, *Quantum hashing via ε-universal hashing constructions and Freivalds' fingerprinting schemas*. 16th DCFS 2014, Turku. Lecture Notes in Computer Science **8614**, pp. 42–52, (2014).
2. F. Ablayev and A. Vasiliev, *Cryptographic quantum hashing*, Laser Phys. Lett, **11**(2):025202, (2013).
3. F. Ablayev and A. Vasiliev, *Computing Boolean functions via quantum hashing*, Computing with New Resources, Lecture Notes in Computer Science **8808**, pp. 149–160 (2014).
4. D. Gottesman and I. Chuang, *Quantum digital signatures*, arXiv:quantph/0105032, (2001).
5. A. Holevo, *Some estimates of the information transmitted by quantum communication channel* (russian), Probl. Pered. Inform. [Probl. Inf. Transm.], 9(3):311, (1973).
6. A. Korol'kov, *About some applied aspects of quantum cryptography in the context of development of quantum computations and emergense of quantum computations and emergence of quantum computers (russian)*. Voprosy kiberbezopasnosty, 1(9), (2015).
7. R. Amiri and E. Andersson, *Unconditionally Secure Quantum Signature*, Entropy, **17**: pp. 5635–5659, (2015).
8. A. Ben-Aroya and A. Ta-Shma, *Constructing Small-Bias Sets from Algebraic-Geometric Codes*, Theory of Computing **9**: pp. 253-272 (2013).
9. J. Naor and M. Naor, *Small-bias probability spaces: Efficient constructions and applications*, Proceedings of the twenty-second annual ACM symposium on Theory of computing, pp. 213–223 (1990).
10. A. Nayak, *Optimal Lower Bounds for Quantum Automata and Random Access Codes*, arXiv:quant-ph/9904093v3, (1999).
11. A. Vasiliev, *Quantum Hashing for Finite Abelian Groups*, arXiv:1603.02209 [quant-ph], (2016).