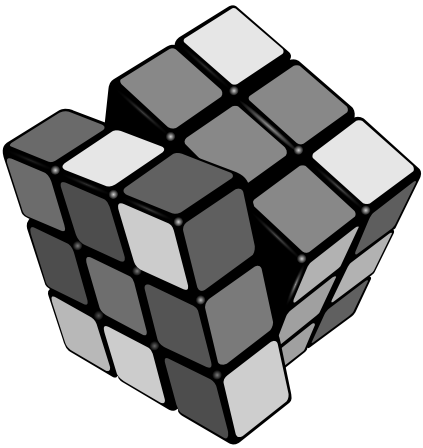
Autumn School of the Department of Algebra

Roztoky u Křivoklátu, November 19–22, 2015

ABSTRACTS



Cover image courtesy of Wikipedia.

Contents

Another Introduction to Modern Cryptography

Miloslav Homer

1. Introduction

In this talk, an introduction to cryptography will be presented. Firstly, basic notions such as split to symmetric and asymmetric cryptography, goals of cryptography and types of security will be shown. Secondly, we shall look at Impagliazzo's Five Worlds – based on his article A Personal View of Average-Case Complexity describing few implications based on truthfullness of $P = NP$ and other such problems. And finally, we will focus on cryptographic models and present few key definitions and examples.

2. Basic notions

Encryption, Decryption, Key – Symmetric/Asymmetric Cryptography. Goals of Cryptography.

Definition 1 (One-way function)**.** Let $f\colon A \to B$ be a function. We say that f is *one-way* if and only if there exists polynomial time algorithm computing f, but any polynomial randomized algorithm computing f^{-1} succeeds with negilible probability.

3. Impagliazzo's Five Worlds

(1) *Algorithmica* – $P = NP$ or at least some fast probabilistic algorithms solving NP.
(2) *Heuristica* – NP problems are hard in the worst case, but easy on average.
(3) *Pessiland* – NP problems are hard on average and we cannot create hard NP problems with known solutions, ie no one-way functions exists.
(4) *Minicrypt* – One-way functions exist, but public cryptography does not.
(5) *Cryptomania* – It is possible for two parties to agree on a secret message using only publicly accessible channels.

4. Cryptographic Models

Definition 2 (Zero knowledge proof)**.** We say that a protocol is *zero-knowledge* if there exists a simulator (that does not have access to a prover) that can simulate a malicious verifier's output after interaction with a prover.

Definition 3 (Random Oracle)**.** We call $f\colon A \to B$ a *random oracle* if and only if f responds to every unique $a \in A$ with a uniformly chosen random $b \in B$. If $a \in A$ is repeated, f responds with the same b.

Definition 4 (Common Reference String)**.** By *common reference string* we mean a public (ie all parties have access to it) uniformely randomly selected string chosen before any protocol interaction starts.

Zeta- and L-functions and prime numbers

Adam Stejskal

In this lecture we will introduce the Riemann zeta function and Dirichlet's L-functions. Then we will present the sketch of proof of Dirichlet's theorem on arithmetic progressions. Finally, we will discuss the requirements for a general L-function.

1. Riemann zeta function

Definition 1. The *Riemann zeta function* is defined by

$$\zeta(s) = \sum_{n\in\mathbb{N}} \frac{1}{n^s},$$

where $s \in \mathbb{C}$, such that $\mathbf{Re}(s) > 1$.

Lemma 2 (Euler product)**.** *For each $s \in \mathbb{R}$, $s > 1$, the following holds:*

$$\zeta(s) = \sum_{n\in\mathbb{N}} \frac{1}{n^s} = \prod_{p\in\mathbb{P}} \frac{1}{1-p^{-s}}.$$

Definition 3. Let f be a holomorphic function defined on an open set $U \neq \emptyset \subset \mathbb{C}$ and let V is an open set, such that $U \subset V$, and F is a holomorphic function on V, such that

$$F(z) = f(z), \quad \forall z \in U.$$

Then F is called *analytic continuation* of f to V.

Definition 4. We define the *Gamma function* for $t \in \mathbb{C}, \mathbf{Re}(t) > 0$ by integral

$$\Gamma(t) = \int_0^\infty x^{t-1}e^{-t}dt.$$

The integral converges absolutely and satisfies the *functional equation*

$$\Gamma(t+1) = t\Gamma(t).$$

Moreover, we can uniquely extend the *Gamma function* by putting $\Gamma(t) = t^{-1}\Gamma(t+1)$ to whole complex plane except integers $t \leq 0$, where the *Gamma function* has simple poles.

Theorem 5 (Riemann 1859)**.** *The completed zeta function*

$$\xi(s) = \pi^{-\frac{s}{2}}\Gamma(\frac{s}{2})\zeta(s)$$

has analytic continuation to $s \in \mathbb{C}$, except points $s = 0, 1$, where it has simple poles. Moreover $\xi(s)$ satisfies the functional equation

$$\xi(s) = \xi(1-s).$$

2. Dirichlet L-functions

Definition 6. Let $N \in \mathbb{N}$. We say that $\chi : \mathbb{Z} \to \mathbb{C}$ is a *Dirichlet character* modulo N iff the following holds:

- $\chi(n) = \chi(n+N)$ for all $n \in \mathbb{Z}$;
- $\chi(n) = 0$ if and only if $gcd(n, N) > 1$;
- $\chi(1) = 1$;
- $\chi(mn) = \chi(m)\chi(n)$ for all $m, n \in \mathbb{Z}$.

Definition 7. Let χ be a Dirichlet character. Then we define a *Dirichlet L-function* to be

$$L(s, \chi) = \sum_{n \in \mathbb{N}} \frac{\chi(n)}{n^s}$$

for $s \in \mathbb{C}$, $\mathbf{Re}(s) > 1$.

Note. The Dirichlet L-function can be, like the Riemann zeta function, written as *Euler product*. It is

$$L(s, \chi) = \sum_{n \in \mathbb{N}} \frac{\chi(n)}{n^s} = \prod_{p \in \mathbb{P}} \frac{1}{1 - \chi(p)p^{-s}}.$$

Moreover, one can also find the analytic continuation of $L(s, \chi)$ to whole complex plane (now it is actually whole complex plane, whenever χ is a nontrivial character).

The choice of a Dirichlet character in $L(s, \chi)$ allow us to distinguish between numbers of different residue classes. It will be useful in the proof of the following theorem.

Theorem 8 (Dirichlet theorem on aritmethic progressions)**.** *Let $n \in \mathbb{N}$, $a \in \mathbb{Z}$ such that $gcd(a, n) = 1$. Then there exist infinitely many prime numbers p satisfying the congruence*

$$p \equiv a (mod\ n).$$

3. General L-functions

Let $X = (a_n)_{n \in \mathbb{N}}$ be a sequence of complex numbers and put

$$L(s, X) = \sum_{n \in \mathbb{N}} \frac{a_n}{n^s}.$$

Furthermore suppose that following holds:

- $L(s, X)$ is absolutely convergent for $\mathbf{Re}(s) > k, k \in \mathbb{N}$
- $L(s, X)$ has analytic continuation to $\mathbb{C}$
- $L(s, X) = \gamma(s, X)L(k - s, X')$ for some "elementary" function γ, $k \in \mathbb{N}$
- $L(s, X) = \prod_{p \in \mathbb{P}} F(p^{-s})^{-1}$ where F is a polynomial.

It is conjectured, that analysis of such $L(s, X)$ should yield some information about an object associated with X, as it does in case of Dirichlet L-functions. This $L(s, X)$ is called a *(general) L-function*.

Modular forms

Josef Svoboda

In this lecture we will introduce special complex functions called modular forms and discuss their basic properties. Then we will suggest a connection with theory of L-functions and elliptic curves.

1. Basic definitions

Definition 1. The *upper half-plane* $\mathcal{H}$ is set of complex numbers with positive imaginary part. Let $SL_2(\mathbb{Z})$ be a group of 2×2-matrices over $\mathbb{Z}$. This group has an action on $\mathcal{H}$ defined by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} z = \frac{az+b}{cz+d}$$

Lemma 2. *Group G of all fractional transformations is isomorphic to $SL_2(\mathbb{Z})/\{-1,1\}$. It is generated by elements $T(z) = z+1$ and $S(z) = -\frac{1}{z}$.*

Definition 3. The set D of complex numbers z with $|z| \geq 1$ and $-1/2 \leq Re(z) \leq 1/2$ is called *fundamental domain* for action of $SL_2(\mathbb{Z})$ on the half plane $\mathcal{H}$.

Definition 4. Function f on $\mathcal{H}$ is called *modular form of weight $2k$* if it satisfies following conditions:

(1) $f(z) = (cz+d)^{-2k} f\left(\frac{az+b}{cz+d}\right)$ for $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z})$.

(2) f is holomorphic on $\mathcal{H}$.

(3) f is holomorphic at ∞.

Modular form is called *cusp form* if it satisfies also

(4) $f(\infty) = 0$.

Lemma 5. *Let $q = e^{2\pi i z}$. Every modular form f can be written as a series $f(z) = \sum_{n=0}^{\infty} a_n q^n$ which converges for $|q| < 1$ (i.e. for $Im(z) > 0$).*

Definition 6. *Eisenstein series* is defined by

$$E_k(z) = \sum_{(m,n) \neq (0,0) \in \mathbb{Z}^2} \frac{1}{(mz+n)^{2k}}$$

Proposition 7. *Let k be an integer ≥ 2. The Eisenstein $E_k(z)$ is a modular form of weight $2k$.*

2. Spaces of modular forms

Definition 8. Define Δ function as

$$\Delta(z) = (60E_2(z))^3 - 27(140E_3(z))^2$$

It is a cusp form of weight 12.

Definition 9. Modular forms (resp cusp forms) of weight $2k$ form a $\mathbb{C}-$vector space, which we denote by M_k (resp. S_k).

Theorem 10. *(1) We have $M_k = 0$ for $k < 0$ and $k = 1$.*

(2) For $k = 0, 2, 3, 4, 5$, M_k is a vector space of dimension 1 with basis 1, E_2, E_3, E_4, E_5; we have $S_k = 0$.

(3) Multiplication by Δ defines an isomorphism of M_{k-6} onto S_k.

Corollary. *We have*

$$\dim M_k = \begin{cases} \lfloor \frac{k}{6} \rfloor & \text{if } k \equiv 1 \pmod 6, k \geq 0 \\ \lfloor \frac{k}{6} \rfloor & \text{if } k \not\equiv 1 \pmod 6, k \geq 0 \end{cases}$$

Corollary. *Space M_k has for basis the family of monomials $E_2^\alpha E_3^\beta$ with $\alpha, \beta \in \mathbb{N}_0$ and $2\alpha + 3\beta = k$.*

3. From modular forms to L-functions

Definition 11. Let $f(z) = \sum_{n=0}^{\infty} a_n q^n$ is a modular form. L-function for f is defined by

$$L(s, f) = \sum_n \frac{a_n}{n^s}$$

Theorem 12. *Function $L(s, f)$ satisfies the following conditions:*

(1) $L(s, f)$ is absolutely convergent for $\mathbf{Re}(s) > 2k, k \in \mathbb{N}$

(2) $L(s, f)$ has analytic continuation to $\mathbb{C}$

(3)

$$L(s, f) = \prod_{p \in \mathbb{P}} \frac{1}{1 - a_p p^{-s} + p^{2k+1-2s}}$$

(4) Let $\widetilde{L}(s, f) = (2\pi)^{-s}\Gamma(s)L(s, f)$. Then

$$\widetilde{L}(s, f) = (-1)^k \widetilde{L}(2k - s, f)$$

Note. To an elliptic curve E we can associate an L-function $L(s, E)$ whose coefficients are the numbers of solutions mod p. The importance of modular forms comes from the fact that for each E there is a modular form f such that $L(s, E) = L(s, f)$.

Partitions and modular forms

Kristýna Zemková

At first we will introduce the partition function and its arithmetic properties. After that we will repeat some basic facts about modular forms and will define the Ramanujan function. At the end we will briefly look at the connection between modular forms and the partition function.

1. Partitions

Definition 1. A *partition* of a positive integer n is any nonincreasing sequence $\lambda_1, \ldots, \lambda_r$ of positive integers such that $n = \lambda_1 + \cdots + \lambda_r$. The number of partitions of n is denoted by $p(n)$ (by convention is $p(0) = 1$ and $p(-n)$=0 for each $n > 0$).

Lemma 2 (Generating function)**.**

$$\sum_{n=0}^{\infty} p(n)x^n = \prod_{n=1}^{\infty} \frac{1}{1-x^n}$$

Theorem 3 (Euler's Pentagonal Number Theorem)**.**

$$\prod_{n=1}^{\infty}(1-x^n) = \sum_{k-\infty}^{\infty}(-1)^k x^{(3k^2-k)/2} = 1 - x - x^2 + x^5 + x^7 - x^{12} - \ldots$$

Corollary.

$$p(n) = p(n-1) + p(n-2) - p(n-5) - p(n-7) + p(n-12) + \cdots$$

Theorem 4 (Ramanujan's congruences)**.**

$$p(5n+4) \equiv 0 \pmod 5$$

$$p(7n+5) \equiv 0 \pmod 7$$

$$p(11n+6) \equiv 0 \pmod{11}$$

2. Modular forms

Definition 5. Let z be a complex number with strictly positive imaginary part. For an integer $k \geq 2$ we define the *Eisenstein series* of index k by the following series:

$$E_k(z) = \sum_{(m,n)\in\mathbb{Z}^2\setminus(0,0)} \frac{1}{(mz+n)^{2k}}.$$

Then we put

$$\Delta = 216\,000\, E_2^3 - 529\,200\, E_3^2.$$

Definition 6. The *Ramanujan function* is the function $\tau : \mathbb{N} \to \mathbb{Z}$ defined as the nth coefficient of the cusp form $F(z) = (2\pi)^{-12}\Delta(z)$. Thus

$$\sum_{n=1}^{\infty} \tau(n)q^n = q\prod_{n=1}^{\infty}(1-q^n)^{24},$$

where $q = e^{2\pi iz}$.

Theorem 7 (Properties of $\tau(n)$)**.**

(i) $\tau(n) = O(n^{11/2+\epsilon})$ *for every* $\epsilon > 0$,
(ii) $\tau(mn) = \tau(m)\tau(n)$ *if* $GCD(m,n) = 1$,
(iii) $\tau(p^{n+1}) = \tau(p)\tau(p^n) - p^{11}\tau(p^{n-1})$ *for* p *prime,* $n > 1$.

3. A modular form for the partitition function

Theorem 8.

$$\sum_{n=1}^{\infty} \tau(n)q^{n-1} = \frac{1}{\left(\sum_{n=1}^{\infty} p(n)q^n\right)^{24}},$$

where again $q = e^{2\pi iz}$ *and* $p(n)$ *denotes the partition function.*

Theorem 9. *For any prime* $l \geq 5$*, there exist infinitely many congruences of the form*

$$p(An + B) \equiv 0 \pmod{l}.$$

An application of algebraic geometry to a combinatorial problem

Michal Szabados

1. Motivation

We are interested in the study of colorings of the infinite two-dimensional square grid by finitely many colors. Such a coloring is *periodic* if there exists a non-zero vector such that any two positions which differ by the vector have the same color.

We can impose local restrictions on the coloring: we can say that only a few patterns of certain size are allowed. Intuitively, the smaller number of patterns allowed, the more restricted the coloring is. The question is, what restrictions imply that the coloring is necessarily periodic?

Morse-Hedlund theorem answers such a question in one dimension. We are interested in its natural generalization to two dimensions, called Nivat's conjecture.

Theorem 1 (Morse and Hedlund, 1938)**.** *Let A be a finite set and $c : \mathbb{Z} \to A$ a bi-infinite sequence of "symbols" from A. Denote $P(n)$ the number of distinct subwords of length n occuring in c. Then c is periodic iff $\exists n \colon P(n) \leq n$.*

Conjecture 2 (Nivat, 1997)**.** *Let A be a finite set and $c : \mathbb{Z}^2 \to A$ a two-dimensional array of "symbols" from A. Denote $P(m, n)$ the number of distinct rectangular $m \times n$ patterns occuring in c. If $\exists m, n \colon P(m, n) \leq mn$ then c is periodic.*

New partial results to Nivat's conjecture have been obtained recently:

Theorem 3 (Cyr and Kra, 2013)**.** *If $\exists m, n \colon P(m, n) \leq mn/2$ then c is periodic.*

Theorem 4 (Kari and S., 2015)**.** *If there exist infinitely many pairs m, n such that $P(m, n) \leq mn$ then c is periodic.*

We will present an algebraic method introduced in the proof of Theorem 4.

2. Colorings as Formal Power Series

Definition 5. A *configuration* c is a formal power series in variables x, y with complex coefficients:

$$c(x, y) = \sum_{(i,j)\in\mathbb{Z}^d} c_{ij} x^i y^j \qquad \text{where } c_{ij} \in \mathbb{C}.$$

A configuration is *integral* if $\forall i, j \colon c_{ij} \in \mathbb{Z}$ and it is *finitary* if there are only finitely many distinct coefficients c_{ij}.

A configurations can be multiplied by a polynomial or a Laurent polynomial to obtain another configuration. For the next definition denote by 0 the all-zero configuration.

Definition 6. An *annihilator polynomial* is a polynomial $f \in \mathbb{C}[x, y]$ such that

$$f(x, y)c(x, y) = 0.$$

An *annihilator ideal* is

$$\mathrm{Ann}(c) = \left\{ f \in \mathbb{C}[x, y] \;\middle|\; f(x, y)c(x, y) = 0 \right\}.$$

It is clear that by choosing an integer representation of colors we can represent a coloring as a finitary integral configuration. Notice that a configuration is periodic iff $x^a y^b - 1$ is an annihilator polynomial for some non-zero $(a, b) \in \mathbb{Z}^2$. Let us show the connection to Nivat's conjecture. Define $P_c(m, n)$ to be the number of rectangular $m \times n$ patterns in the coloring represented by a configuration c.

Lemma 7. *If* $\exists m, n\colon P_c(m, n) \leq mn$ *then* $\mathrm{Ann}(c) \neq \{0\}$.

3. Let's Do Some Algebra

In what follows we fix c to be a finitary integral configuration with a non-trivial annihilator ideal.

Lemma 8. *Configuration c has an annihilator polynomial $f \in \mathbb{Z}[x, y]$.*

Lemma 9. *There exists $r \in \mathbb{N}$ such that if $f \in \mathbb{Z}[x, y]$ is an annihilator polynomial of configuration c, then also $f(x^{kr+1}, y^{kr+1})$ is for any $k \in \mathbb{N}_0$.*

Theorem 10. *There is a Laurent polynomial of the form $(x^{a_1} y^{b_1} - 1) \cdots (x^{a_n} y^{b_n} - 1)$ for some non-zero $(a_i, b_i) \in \mathbb{Z}^2$ which annihilates configuration c.*

Following tools are needed to prove our goal theorem.

Definition 11. The *support* of a polynomial $f = \sum_{(i,j)\in\mathbb{Z}^2} a_{ij} x^i y^j \in \mathbb{C}[x, y]$ is

$$\mathrm{supp}(f) = \{\, (i, j) \in \mathbb{Z}^2 \mid a_{ij} \neq 0 \,\}.$$

A polynomial f is a *line polynomial* if $|\mathrm{supp}(f)| \geq 2$ and the points from $\mathrm{supp}(f)$ are collinear.

Definition 12. Recall some algebraic notions about ideals in a commutative ring R.

- An ideal A is *prime* if $ab \in A \Rightarrow a \in A \vee b \in A$.
- The *radical* of an ideal is $\sqrt{A} = \{\, a \in R \mid \exists m\colon a^m \in A \,\}$. An ideal is *radical* if $A = \sqrt{A}$.
- Ideals A and B are *comaximal* if $A + B = R$, or equivalently if $1 \in A + B$.

Theorem 13 (Commutative algebra course)**.**

(i) If $A_1, \ldots, A_n$ are pairwise comaximal then $A_1 \cap \cdots \cap A_n = A_1 \cdots A_n$.

(ii) Proper prime ideals in $\mathbb{C}[x, y]$ are principal ideals generated by irreducible polynomials and maximal ideals.

(iii) Let $A \in \mathbb{C}[x, y]$ be proper. Then $\sqrt{A}$ can be uniquely written as a finite intersection of prime ideals $P_1 \cap \cdots \cap P_n$ where $P_i \not\subset P_j$ for $i \neq j$.

The goal of the talk is to prove the following:

Theorem 14. $\mathrm{Ann}(c)$ *is radical.*

Gabriel's theorem – Part I

Ondřej Draganov

In this part we will introduce basic definitions, lemmas and theorems needed for the proof of Gabriel's theorem.

1. Motivation

Let V be a vector space (of a finite dimension) and $U, W \leq V$ be subspaces of V as depicted on the figure 1.

Figure 1

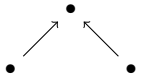


Figure 2

It is an interesting problem from linear algebra to classify how can these two subspaces be included in V. It is known that there are up to an isomorphism only finitely many indecomposable cases how can two subspaces be included in a vector space of a finite dimension. At this point we will not discuss in what sense "up to an isomorphism" and "indecomposable". This also holds for three subspaces. However for four subspaces there are infinitely many indecomposable cases.

We will generalize this idea and present the Gabriel's theorem, which tells us when do we have "only finitely many indecomposable cases up to an isomorphism".

2. Basic definitions

Let K be a field. We fix that field throughout the text. When we talk about a vector space, we mean vector space over K.

Definition 1 (Quiver)**.** A *quiver* is a directed graph with multiple edges (arrows) and loops. We assume quivers to be finite. More precisely it is a quadruple $Q = (Q_0, Q_1, s, t)$, where Q_0 is a finite set of *vertices*, Q_1 is a finite set of *arrows* and $s, t : Q_0 \to Q_1$ are two maps which denotes where does the arrow *start* and *terminate*. For an arrow $\alpha \in Q_1$ we sometimes write $\alpha : s(\alpha) \to t(\alpha)$.

An example of a very simple quiver is the figure 2, which is in a sense a scheme for the situation on a figure 1. For fixed vector spaces V, U, W, the situation on figure 1 is a *representation* of the quiver on figure 2.

Definition 2 (Representation)**.** Let Q be a quiver. A *representation* of Q is a collection

$$X = (X_i, X_\alpha)_{i \in Q_0, \alpha \in Q_1}$$

consisting of a vector space X_i for each vertex i and a linear map $X_\alpha : X_{s(\alpha)} \to X_{t(\alpha)}$ for each arrow α.

If we have two representations of the same quiver, we would like to say for example when those two are "similar", i.e. isomorphic. We would therefore like to define some "maps" between representations.

Definition 3 (Morphism)**.** Let Q be a quiver and let X, Y be its representations. A *morphism* $\phi : X \to Y$ of these representations is a collection of linear maps $\phi_i : X_i \to Y_i$ for each vertex i, such that $Y_\alpha \phi_{s(\alpha)} = \phi_{t(\alpha)} X_\alpha$ for each arrow α, that is such that the following diagram commutes:

$$\begin{array}{ccc} X_{s(\alpha)} & \xrightarrow{\phi_{s(\alpha)}} & Y_{s(\alpha)} \\ \downarrow X_\alpha & & \downarrow Y_\alpha \\ X_{t(\alpha)} & \xrightarrow{\phi_{t(\alpha)}} & Y_{t(\alpha)} \end{array}$$

We can now carry various concepts defined for vector spaces and linear maps over to representations and morphisms by applying the vector space definitions *point-wise*, that is, for each vertex i. For example a morphism ϕ of representations is an *isomorphism* if ϕ_i is an isomorphism for each vertex i. Or a representation X is *finite dimensional* if each X_i is finite dimensional.

Similarly we can define a *direct sum* of representations X and Y as a representation $Z = X \oplus Y$, where $Z_i = X_i \oplus Y_i$ for each vertex i and the maps Z_α are defined accordingly. A representation is called *indecomposable* if it cannot be written as a direct sum of representations. By Krull-Schmidt theorem, every finite dimensional representation can be decomposed into a direct sum of finitely many indecomposable representations. This decomposition is unique (in a natural sense).

Definition 4 (Finite representation type)**.** We say a quiver Q is of *finite representation type* if there exist (up to an isomorphism) only finitely many indecomposable finite dimensional representations of Q.

For a quiver Q we denote Γ an *underlying graph*, which is Q without an orientation. The main goal of both parts of this presentation is to prove the Gabriel's theorem. The following statement is actually a corollary of the theorem.

Theorem 5 (Gabriel)**.** *Let Q be a connected quiver. Then there are up to an isomorphism only finitely many indecomposable finite dimensional representations of Q if and only if the underlying graph Γ of Q is Dynkin diagram (see figure 3).*

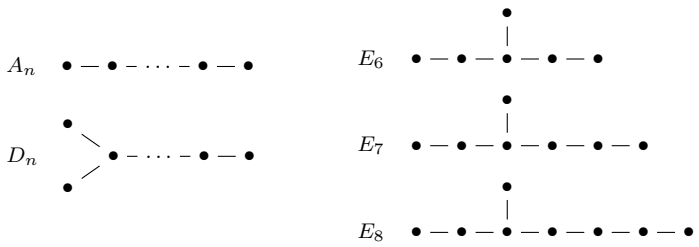


FIGURE 3. Dynkin diagrams

Gabriel's Theorem – Part II

Peter Kálnai

Following [Assem, Simson, Skowronski: "Elements of the Representation Theory of Associative Algebras", Volume 1 Techniques of Representation Theory, 2006, Chapter VII], we try to shed some light on both combinatorial and categorical techniques behind the following theorem:

Theorem 1 (Gabriel's Theorem, 1972)**.** *Let Q be a finite, connected, and acyclic quiver; let K be an algebraically closed field; and $rep_K(Q)$ be the category of all finite-dimensional K-linear representations of Q.*

Then $rep_K(Q)$ is of finite-representation type if and only if the underlying graph $\overline{Q}$ of Q is one of the Dynkin graphs $\mathbb{A}_n$, $\mathbb{D}_n$, with $n \geq 4$, $\mathbb{E}_6$, $\mathbb{E}_7$ or $\mathbb{E}_8$.

The *underlying graph* $\overline{Q}$ of a quiver Q is obtained from Q by forgetting the orientation of the arrows. The index in the Dynkin graphs always refers to the number of points in the graph, whereas in the so-called *Euclidean graphs* and denoted as the Dynkin ones but with added$\tilde{}$, it refers to the number of points minus one. In fact, a Euclidean graph can be constructed from the corresponding Dynkin graph by adding one point and could be considered in a sense as minimal non-Dynkin graphs.

1. Necessity for the representation-finiteness of $rep_K(Q)$

We solve the implication partially. As the next two lemmas show, if we want to prove that non-Dynkin graphs escape the representation-finiteness of the corresponding category, it is enough to prove that Euclidean graphs do so:

Lemma 2. *Let Q be a finite, connected and acyclic quiver. If the underlying graph $\overline{Q}$ of Q is not a Dynkin graph, then $\overline{Q}$ contains a Euclidean graph as a subgraph.*

Lemma 3. *Let Q be a finite, connected and acyclic quiver. If Q' is a subquiver of Q such that $rep_K(Q')$ is representation-infinite, $rep_K(Q)$ is representation-infinite.*

If the underlying graph $\overline{Q}$ of a finite, connected and acyclic quiver Q is one of $\widetilde{\mathbb{A}}_m$, $m \geq 1$, then $rep_K(Q)$ is representation-infinite. While the latter fact handles completely one scheme of Euclidean graphs, we prove a similar result for just *some* quivers of the remaining graph types, namely $\widetilde{\mathbb{D}}_n$, $n \geq 4$ and $\widetilde{\mathbb{E}}_6$, $\widetilde{\mathbb{E}}_7$, $\widetilde{\mathbb{E}}_8$.

2. Sufficiency for the representation-finiteness of $rep_K(Q)$

Let Q be a quiver whose underlying graph is a Dynkin graph. We show that the assignment $X \mapsto \dim X$ induces a bijection between the isomorphism classes of indecomposable representations of Q and the positive roots corresponding to the graph of Q. In particular, there are only finitely many isomorphism classes of indecomposable representations.

Lemma 4. *The number of roots of a Dynkin graph over n vertices (i.e. the number of n-tuples of integers for which the (positive definite) quadratic form is less or equal one) is finite.*

For every vertex a of a quiver Q, we define a new quiver $\sigma_a(Q)$ as the quiver Q with all the arrows having a as source or as target are reversed and all other arrows remain unchanged.

- A vertex i of a quiver Q is called a *sink* (resp. *source*) if there is no arrow in Q starting (resp. ending) at i.
- Given any vertex i, the quiver $\sigma_i(Q)$ is obtained from Q by reversing all arrows which start or end at i.
- A sequence $(i_1, i_2, \cdots, i_n)$ of the vertices of Q is called *admissible sequence of sinks* if the vertex i_p is a sink in the quiver $\sigma_{i_{p-1}}(\cdots(\sigma_{i_1}(Q))\cdots)$ for each $p = 2, \cdots, n$.

Lemma 5. *There exists an admissible sequence of sinks of a quiver Q if and only if there are no oriented cycles in Q.*

Definition 6 (Reflection functor $\mathcal{S}_i^+$ resp. $\mathcal{S}_i^-$ of a sink resp. a source i)**.** (This concept will be explained in the presentation.)

Lemma 7. *Let i be a sink in a finite, connected and acyclic quiver Q with at least two points; let i be a sink in Q;and let X be an indecomposable representation of Q. Then the reflection functors $\mathcal{S}_i^+ : rep_K(Q) \to rep_K(\sigma_i(Q))$ and $\mathcal{S}_i^- : rep_K(\sigma_i(Q)) \to rep_K(Q)$ satisfy*

(i) the functor $\mathcal{S}_i^-$ is left adjoint to $\mathcal{S}_i^+$

(ii) if X is indecomposable representation of Q then the following is equivalent:

(1) $\mathcal{S}_i^+(X) \neq 0$

(2) X is not isomorphic to the simple representation of the vertex i

Moreover, if this is the case, then $\mathcal{S}_i^-(\mathcal{S}_i^+(X)) \simeq X$.

(iii) if Y is indecomposable representation of $\sigma_i(Q)$ then the following is equivalent:

(1') $\mathcal{S}_i^-(Y) \neq 0$

(2') Y is not isomorphic to the simple representation of the vertex i

Moreover, if this is the case, then $\mathcal{S}_i^+(\mathcal{S}_i^-(Y)) \simeq Y$.

Combinatorics on words and automated proving I – Basic definitions and examples

Jan Fišer

Definition 1. An *alphabet* is a nonempty finite set, its elements are called *letters*, and sequences of letters are called *words*. As usual, we define the *length* $|w|$ of a word w as the number of letters of w, the *empty word* ε as the word of length 0, and *concatenation* uv of two words u and v. We denote A^* the set of all (finite) words. We say that v is a *factor* of w if $w = u_1vu_2$ for some $u_1, u_2, v, w \in A^*$.

Definition 2. A *square* is a word of the form ww where w is a nonempty word. An *overlap* is a word of the form $awawa$ where $a \in A$ is a letter and $w \in A^*$ a (possibly empty) word.
We say that a word is *square-free* if it contains no square factor. Similarly, a word is *overlap-free* if it has no overlapping factor.

Definition 3. Considering a mapping $a\colon \mathbb{N} \to A$, we define an *infinite word* as the infinite sequence $a(0)a(1)a(2)\dots$ of letters of the alphabet A, usually written as $a_0a_1a_2\dots$.
The set of all infinite words over A is denoted by A^ω. We say that an infinite word has a property P if all its factors do.

From now on, let $A = \{a, b\}$ be the alphabet. Over such alphabet, we obtain the complement $\overline{w}$ of a word w by exchanging a's and b's.

An interesting example of an overlap-free infinite word is the Thue-Morse sequence (word).

Definition 4. We define the *Thue-Morse word* as the limit

$$\mathbf{t} = \lim_{n\to\infty} U_n$$

where

$$U_0 = a, \quad U_n = U_{n-1}\overline{U}_{n-1} \text{ for } n \geq 1.$$

Explicitly, $\mathbf{t} = abbabaabbaababba\dots$.

Proposition 5. *The Thue-Morse word is overlap-free.*

Proposition 6. *Squares of the Thue-Morse word have length either* 2^n *or* $3\cdot 2^n$ *for some* $n \in \mathbb{N}$.

Another well-known infinite word is the (infinite) Fibonacci word.

Definition 7. The *Fibonacci word* is the infinite word obtained as the limit

$$S_\infty = \lim_{n\to\infty} S_n$$

where

$$S_0 = a, \quad S_1 = ab, \quad S_n = S_{n-1}S_{n-2} \text{ for } n \geq 2.$$

Explicitly, $S_\infty = abaababaabaab\dots$.

Combinatorics on words and automated proving II – Theory behind the prover

Jan Butora

At first we'll focus on automatic sequences, their representation by finite automata and some of their properties. In the second part we take a brief introduction into theory that allows us to purely mechanically extract information about automatic sequences.

1. Automatic sequences

Definition 1 (Finite automaton)**.** A *finite automaton* is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

(i) Q is finite set called the *states*
(ii) Σ is a finite set called the *alphabet*
(iii) $\delta : Q \times \Sigma \to Q$ is the *transition function*
(iv) $q_0 \in Q$ is the *start state*, and
(v) $F \subseteq Q$ is the *set of accept states*.

Definition 2. Let B be finite alphabet. We denote by B^* the set of all words written with letters of B, including the empty word ϵ. We say that $L \subseteq B^*$ is *recognizable* if it is the set of words accepted by some finite automaton.

Definition 3 (k-automatic sequence)**.** An infinite sequence $\mathbf{a} = (a_n)_{n\geq 0}$ over a finite alphabet is said to be *k-automatic* if there exists a deterministic finite automaton (with output associated with the states) such that after completely processing the input n expressed in base k, the automaton reaches some state q with output a_n.

2. Logic

Definition 4 (Presburger arithmetic)**.** Let $\omega = \mathbb{N} \cup \{0\}$. Than by $Th(\omega, +)$ we mean set of all true first-order sentences in the logical theory of the natural numbers with addition. We call this set *Presburger arithmetic*.

Example. Any element of ω can be defined in the language $(\omega, +)$. For $x = 1$, it is the following formula

$$(\neg(x = 0)) \wedge ((\forall y)(\neg(y = 0)) \to (x \leq y)) .$$

Presburger arithmetic was proved to be *decidable* (that is, there exists an algorithm that, given a sentence in the theory, will decide its truth). However, even more is true. If we add the function $V_k : \omega \to \omega$ to Presburger arithmetic, where $V_k(x) = k^n$, and k^n is the largest power of k dividing x, it is still decidable. That gives us the ability to decide many questions about automatic sequences. Thus we have

Theorem 5. *There is an algorithm that, given a predicate phrased using only the universal and existential quantifiers, indexing into a given automatic sequence* $\mathbf{a}$*, addition, subtraction, logical operations, and comparisons, will decide the truth of that proposition.*

Combinatorics on words and automated proving III – Walnut

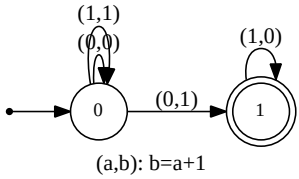
Dominik Lachman

In this talk, we present Walnut prover, programmed by Hamoon Mousavi. We learn its commands and then we let Walnut to prove some properties of Thue-Morse and Fibonacci words. Walnut, together with a manual and many examples, is available on the website of Jeffrey O. Shallit `https://cs.uwaterloo.ca/~shallit/papers.html`.

1. Example of using Walnut

input: command *eval example "b=a+1"*

output: 2 files describing the resultant automaton: *example.txt* (table-form description) and *example.gv* (graph-form description)



The automaton expects two binary words a and b. It stars at state 0 and at step n it moves to a new state, following an arrow labeled $(a[n], b[n])$.

In Walnut, binary base is not the only one we can work with. All bases of orders $n \in \mathbb{N}, n > 1$ are possible. We can even use Fibonacci base.

2. Some theorem that can be proved by Walnut

Definition 1. Let $\Sigma = \{0, 1\}$ and $\mu, \varphi : \Sigma^* \to \Sigma^*$ be morphisms such that μ maps $0 \to 01$, $1 \to 10$ and φ maps $0 \to 01$, $1 \to 0$. Then the fixed point of μ:

$$\mathbf{t} = \mu^\omega(0) = 0110100110010110\cdots$$

is called the *Thue-Morse infinite word*. And the fixed point of φ:

$$\mathbf{f} = \varphi^\omega(0) = 01001010010010100 1010\cdots$$

is called the *infinite Fibonacci word*.

Using Walnut, we can purely mechanically prove these theorems:

Theorem 2. *The Thue-Morse infinite word is overlap-free.*

Theorem 3. *Infinite Fibonacci word is not ultimately periodic, that is there are no words $u, v \in \{0, 1\}^*$, such that $\boldsymbol{f} = uv^\omega$.*

Theorem 4. *There exist palindromes of every lenght ≥ 0 in the infinite Fibonacci word.*

Pairing-based cryptography I – Pairing basics

Radka Luňáčková

Definition 1 (Discrete logarithm problem (DLP), additive notation)**.** DLP in group $G = \langle P \rangle$ of order n is the problem, given P and Q, of finding the integer $x \in \{0, 1, ..., n-1\}$ such that $Q = xP$.

Definition 2 (Diffie-Hellman problem (DHP), additive notation)**.** DHP in group $G = \langle P \rangle$ of order n is the problem, given P, aP and bP, of finding abP, where $a, b \in \{0, 1, ..., n-1\}$.

Definition 3 (Bilinear pairing)**.** Let n be a prime number. Let $G_1 = \langle P \rangle$ be an additively-written group of order n with identity $\mathcal{O}$ and let G_T be a multiplicatively-written group of order n with identity 1. A bilinear pairing on (G_1, G_T) is a mapping

$$\hat{e} : G_1 \times G_1 \to G_T$$

that satisfies the following conditions:

(i) bilinearity:

$$\hat{e}(R + S, T) = \hat{e}(R, T)\hat{e}(S, T)$$

$$\hat{e}(R, S + T) = \hat{e}(R, S)\hat{e}(R, T)$$

for all $R, S, T \in G_1$,

(ii) non-degeneracy: $\hat{e}(P, P) \neq 1$,

(iii) computability: $\hat{e}$ can be efficiently computed.

Definition 4 (Bilinear Diffie-Hellman problem (BDHP))**.** Let $\hat{e}$ be a bilinear pairing on (G_1, G_T). The BDHP is the problem of computing $\hat{e}(P, P)^{abc}$, given P, aP, bP and cP.

Definition 5 (Elliptic curve)**.** Let K be an algebraically closed field and assume that K has characteristic different from 2 and 3. Then we can define elliptic curve E over K by equation

$$y^2 = x^3 + ax + b,$$

where $a, b \in K$ and $4a^3 + 27b^2 \neq 0$. The set of points of E over K is denoted $E(K)$ and defined by

$$E(K) = \left\{(x, y) \in K \times K : y^2 = x^3 + ax + b\right\} \cup \{\mathcal{O}\}.$$

Note: $E(K)$ forms a group, $(E(K), +, -, \mathcal{O})$.

Definition 6 (Weil pairing)**.** Let E be an elliptic curve over K and let $m > 0$ be an integer prime to characteristic of K. The Weil pairing is a mapping $w : E[m] \times E[m] \to K$ defined by

$$w(P, Q) = (-1)^m \frac{f_P(Q)}{f_Q(P)}.$$

Note: $E[m] = \{P \in E(K) : mP = \mathcal{O}\}$ and $f_P, f_Q \in K(x, y)$ and the definition will be in presentation.

Pairing-based cryptography II – Applications

David Kubát

1. Identity-Based Encryption

The idea of Identity-Based Encryption (IBE) is to allow users of Public-Key Cryptosystems to use any string as a public key.

Definition 1. The four algorithms defining any IBE scheme are as follows:

- Setup algorithm: $S(\lambda) \to (MK, PP)$
  given a security parameter λ it outputs a master key MK and public parameters PP
- Extract key: $K(MK, ID) \to sk_{ID}$
  given a user's public key ID it outputs user's private key sk_{ID}
- Encrypt: $E(PP, ID, M) \to C$
  encrypts a message M using public key ID (and PP)
- Decrypt: $D(sk_{ID}, C) \to M$

2. Attribute-Based Encryption

The goal of Attribute-Based Encrytion (ABE) is to encrypt data in such a way, that those able to decrypt it are exactly the users matching a set of attributes specified while encrypting.

We distinguish between Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE messages are encrypted with respect to subsets of attributes and an acces policy is encoded into the users secret key.

In CP-ABE, the roles of attributes sets and acces formulas are flipped, (i.e. users keys are associated with sets of attributes and messages are encrypted with respect to formulas). In this talk, we will only be concerned with KP-ABE.

Definition 2 (KP-ABE specification)**.** There are four algorithms defining a KP-ABE scheme:

- $Setup(\lambda, U)$:
  Given a security parameter λ and an attribute universe U, it generates public parameters PP and a master key MK
- $KeyGen(Policy, MK)$:
  Generates a user key SK for a given policy
- $Encrypt(PP, M, S \subseteq U)$:
  Encrypts message M under attribute set S
- $Decrypt(C, SK)$:
  Decrypts ciphertext using a key

Visual Cryptography

Tereza Hrubešová

Visual Cryptography is a technique that allows information (images, text, diagrams, etc.) to be encrypted using an encoding system that can be decrypted by the human vision.

The technique was proposed by Moni Naor and Adi Shamir in 1994. The basic model consists of a printed page of ciphertext and a printed transparency (which serves as a secret key). The original cleartext is revealed by placing the transparency with the key over the page with the ciphertext, even though each one of them is indistinguishable from random noise. The system is similar to a one time pad in the sense that each page of ciphertext is decrypted with a different transparency. Due to its simplicity, the system can be used by anyone without any knowledge of cryptography and without performing any cryptographic computations.

This basic model can be extended into a visual variant of the k out of n secret sharing problem: Given a written message, we would like to generate n transparencies so that the original message is visible if any k (or more) of them are stacked together, but totally invisible if fewer than k transparencies are stacked together (or analysed by any other method). The original encryption problem can be considered as a 2 out of 2 secret sharing problem.

Definition 1. A solution to the k out of n visual secret sharing scheme consists of two collections of $n \times m$ Boolean matrices $\mathbf{C_0}$ and $\mathbf{C_1}$. To share a white pixel, the dealer randomly chooses one of the matrices in $\mathbf{C_0}$, and to share a black pixel, the dealer randomly chooses one of the matrices in $\mathbf{C_1}$. The chosen matrix defines the colour of the m subpixels in each one of the n transparencies. The solution is considered valid if the following three conditions are met:

(i) For any $\mathbf{S}$ in $\mathbf{C_0}$, the "OR" operation of any k out of n rows of $\mathbf{S}$ is a vector v, that satisfies $w_H(v) \leq d - \alpha m$ (for some fixed threshold $1 \leq d \leq m$) where $w_H(v)$ is the Hamming weight of the vector v, m is the pixel expansion of the scheme and α is the "contrast" of the scheme.

(ii) For any $\mathbf{S}$ in $\mathbf{C_1}$, the "OR" operation of any k out of n rows of $\mathbf{S}$ is a vector v, that satisfies $w_H(v) \geq d$.

(iii) For any subset $\{i_1, i_2, \ldots, i_q\}$ of $\{1, 2, \ldots, n\}$ with $q < k$, the two collections of $q \times m$ matrices $\mathbf{D_t}$ for $t \in \{0, 1\}$ obtained by restricting each $n \times m$ matrix in $\mathbf{C_t}$ ($t \in \{0, 1\}$) to rows $i_1, i_2, \ldots, i_q$ are indistinguishable in the sense that they contain the same matrices with the same frequencies.

The parameters of a scheme:

- m – the number of pixels in a share
- α – the relative difference in weight between combined shares that come from a white pixel and a black pixel in the original picture
- r – the size of the collections $\mathbf{C_0}$ and $\mathbf{C_1}$, $\log r$ represents the number of random bits needed to generate the shares

Cryptography is not just encryption – Obfuscation

Martin Mach

Definition 1 (Obfuscator of circuits under Virtual Black Box security)**.** $\mathcal{O}$ is an *obfuscator of circuits* if

(i) *Correctness:* $\forall c$ circuit , $\mathcal{O}(c) \equiv c$.

(ii) *Efficiency:* $\forall c$ circuit, $|\mathcal{O}(c)| \leq \text{poly}(|c|)$.

(iii) *VBB:* $\forall A, A$ is bounded, $\exists S$ PPT simulator s.t. $\forall c$ circuit, $\exists \mu$ negligible function.:

$$\left|\Pr\left[A(\mathcal{O}(c)) = 1\right] - \Pr\left[S^c\left(1^{|c|} = 1\right)\right]\right| \leq \mu\left(|c|\right).$$

Theorem 2. *Obfuscators of circuits under VBB security do not exists.*

Some of the weaker variants of Obfuscation

Definition 3 (Indistinguishability Obfuscator for circuits)**.** We call $i\mathcal{O}$ an *indistinguishability obfuscator* for a circuit class $\{\mathcal{C}_\lambda\}$ if

(i) *Corretness:* $\forall \lambda \in \mathbb{N}$ security parametr, $\forall C \in \mathcal{C}_\lambda$, $\forall x$ input, we have that

$$\Pr\left[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)\right] = 1.$$

(ii) *Polynomial slowdown:* $\exists p$ polynom s.t. $\forall C \in \mathcal{C}_\lambda$, we have $|C'| \leq p(|C|)$, where $C' = i\mathcal{O}(\lambda, C)$.

(iii) *Computational indistinguishability:* For any PPT adversaries $Samp, D$, $\exists \mu$ a negligible function s.t.: if $\Pr\left[\forall x, C_0(x) = C_1(x) : (C_0, C_1, \sigma) \leftarrow Samp(1^\lambda)\right] > 1 - \mu(\lambda)$ then we have:

$$\Big|\Pr\left[D(\sigma, i\mathcal{O}(\lambda, C_0)) = 1 : (C_0, C_1, \sigma) \leftarrow Samp(1^\lambda)\right] - \\ - \Pr\left[D(\sigma, i\mathcal{O}(\lambda, C_1)) = 1 : (C_0, C_1, \sigma) \leftarrow Samp(1^\lambda)\right]\Big| \leq \mu(\lambda).$$

Definition 4 (Differing-inputs circuit family)**.** A circuit family $\mathcal{C}$ together with PPT Sampler is called *differing-inputs circuit family* if for every PPT adversary D, $\exists \mu$ a negligible function s.t.:

$$\Pr\left[C_0(x) \neq C_1(x) : (C_0, C_1, \sigma) \leftarrow Samp(1^\lambda), x \leftarrow D(1^\lambda, C_0, C_1, \sigma)\right] \leq \mu(\lambda).$$

Definition 5 (Differing-inputs Obfuscator for circuits)**.** We call $di\mathcal{O}$ a *Differing-inputs obfuscator* for a differing-inputs circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}$ if

(i) *Corretness:* $\forall \lambda \in \mathbb{N}$ security parametr, $\forall C \in \mathcal{C}$, $\forall x$ input, we have that

$$\Pr\left[C'(x) = C(x) : C' \leftarrow di\mathcal{O}(\lambda, C)\right] = 1.$$

(ii) *Polynomial slowdown:* $\exists p$ polynom s.t. $\forall C \in \mathcal{C}$ circuit, we have $|C'| \leq p(|C|)$, where $C' = di\mathcal{O}(\lambda, C)$.

(iii) *Differing-inputs:* For any PPT distinguisher D, $\exists\mu$ a negligible function s.t.: $\forall\lambda \in \mathbb{N}$ security parametr, for $(C_0, C_1, \sigma) \leftarrow Samp(1^\lambda)$, holds:

$$|\Pr[D(di\mathcal{O}(\lambda, C_0), \sigma) = 1] - \Pr[D(di\mathcal{O}(\lambda, C_1), \sigma) = 1]| \leq \mu(\lambda).$$

Usage of Obfuscation

Functional encryption. The task is to learn a function of encrypted data if we have decryption key.

Definition 6 (Functionality)**.** A *functionality* F defined over (K, X) is a function $F\colon K \times X \to \{0,1\}^*$. The set K is called the *key space*, the set X is called the *plaintext space*. The space key K contains a special key called the *empty key* denoted ϵ.

Definition 7 (Functional encryption scheme)**.** A *functional encryption scheme* FE for a functionality F defined over (K, X) is a tuple of four PPT algorithms (Setup, Key, Encrypt, Decrypt) satysfying:

(i) *Correctness:* $\forall k \in K, \forall x \in X$:

- *generate a public and master key pair:* $(pk, mk) \leftarrow \text{Setup}(1^\lambda)$,
- *generate secret key for* k*:* $sk \leftarrow \text{Key}(mk, k)$,
- *encrypt message* x*:* $c \leftarrow \text{Encrypt}(pk, x)$,
- *compute functionality:* $y \leftarrow \text{Decrypt}(sk, c)$,

Then we require: $\Pr[y = F(k, x)] = 1$.

Punctured programs.

Definition 8 (Puncturable family of pseudo-random functions)**.** A *puncturable* family of pseudo-random functions (PRFs) F mapping is given by a triple of programs (Key_F, Puncture_F, Eval_F), and a pair of computable functions $n(\cdot)$ and $m(\cdot)$ satysfying:

(i) *Functionality preserved under puncturing:* for every PPT adversary A such that $A(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$, then $\forall x \in \{0,1\}^{n(\lambda)}$ where $x \notin S$, we have

$$\Pr\left[\text{Eval}_F(K, x) = \text{Eval}_F(K_S, x) : K \leftarrow \text{Key}_F(1^\lambda), K_S = \text{Puncture}_F(K, S)\right] = 1.$$

(ii) *Pseudorandomness at punctured points:* For every PPT adversary (A_1, A_2) s.t. $A_1(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$ and state σ, let us consider an experiment, where

- $K \leftarrow \text{Key}_F(1^\lambda)$,
- $K_S = \text{Puncture}_F(K, S)$.

Then we have for μ a negligible function:

$$\left|\Pr[A_2(\sigma, K_S, S, \text{Eval}_F(K, S)) = 1] - \Pr\left[A_2(\sigma, K_S, S, U_{m(\lambda)\cdot|S|}) = 1\right]\right| \leq \mu(\lambda).$$

Theorem 9. *If one-way functions exists, then for all efficiently computable functions $n(\lambda)$, $m(\lambda)$, there exists a puncturable PRF family that maps $n(\lambda)$ bits to $m(\lambda)$ bits.*

Fully Homomorphic Encryption: A Holy Grail of Cryptography

Jakub Klemsa

In 1977, RSA was publicly announced. A year later, Rivest, Adleman and Dertouzos proposed the basic concept of so called Fully Homomorphic Encryption which had been inspired by the following property of RSA encryption:

$$E(a \cdot b) = E(a) \cdot E(b),$$

i.e. RSA encryption is a multiplicative homomorphism. However, the same does not hold for addition.

The main point was that if some encryption scheme was homomorphic with respect to both, it would be possible to perform arbitrary computation with ciphertexts only! Nowadays you can imagine that a tiny device sends some encrypted data to an untrusted cloud, it performs some demanding computation and sends the result back while the cloud does not learn *anything* about your data. A formal definition of Fully Homomorphic Encryption follows.

Definition 1. Fully Homomorphic Encryption (FHE) is a public key encryption scheme which consists of 4 polynomial-time algorithms (K, E, D, V) where

- given a security parameter λ, K outputs a keypair (pk, sk),
- given pk and a message $m \in \mathcal{M}$, E outputs randomized encryption of m,
- given sk and $c \in \mathcal{C}$, an encryption of m, D outputs m,
- given pk, a polynomial-time evaluable function $f\colon \mathcal{M}^t \to \mathcal{M}$ and $c_1, \ldots, c_t$, encryptions of $m_1, \ldots, m_t$, respectively, V outputs ciphertext c which is an encryption of $f(m_1, \ldots, m_t)$.

Over 30 years FHE was not even known to exist, a positive answer was given by Gentry in 2009. A plenty of new schemes has emerged since then but there is still not any practically usable. We will mainly focus on a specific FHE scheme framework by Nuida and study its practical usability.

Introduction to ECC implementations on embedded devices

Lukáš Pohanka

With the Internet of Things era we are witnessing an outbreak of plethora of new devices and platforms. Most of these devices are small-sized, being limited both computationally and storage-wise. For applications that are not required to run in real-time, it is often more important to optimize for size than performance. This is more than desirable for cryptographic libraries, as they should bring reasonable security and occupy as little space as possible, so it could be more effectively utilized by other applications. In such case, it is obvious to switch to asymmetric cryptosystems based on elliptic curves over fields of large characteristic. They achieve standard levels of security with much smaller primes than RSA.

There are already well-known and standardized elliptic curves usable with ECDSA signature scheme or ECIES. However, in case of ECC there is a long way from a scheme description to the actual implementation – for most of the embedded devices the implementation has to be done from scratch. Another concern is the side-channel security of such implementation, as physical access to an IoT device is usually easier compared to a server in a data warehouse. The NIST standardized elliptic curves are very tricky to implement properly with respect to side-channel security. One of the reasons is that their group law is not unified, nor complete, which leads to additional code branches and implementation pit-falls. This results to a somewhat large code size when implemented properly, wasting precious memory resources of the target device. Another problem arises when going to the higher level – e.g. the ECDSA signature scheme requires a per-message randomness for every signature. On an embedded device, entropy is typically a scarcer resource than memory.

In this presentation, we will take a look at twisted Edwards curves, which are an equivalent representation of a large portion of elliptic curves. Twisted Edwards curves can solve most of the drawbacks of ordinary elliptic curves and make implementation a lot easier and more compact. However, not all coordinate systems of twisted Edwards curves are suitable for memory optimized implementations. We will analyze a signature scheme based on a particular twisted Edwards curve: from the lowest level of modular arithmentic, all the way to the final memory optimized implementation and discuss all the threats waiting for the implementors.

Concatenation hierarchies of star-free languages

Jana Bartoňová

A star-free language is a regular language which can be constructed by means of a regular expression without any use of iteration and with a possible use of complementation instead. More precisely, a star-free language is a language which can be constructed from languages consisting of one one-letter word by means of finitely many applications of the following operations: union, concatenation and complementation.

A concatenation hierarchy of star-free languages arises from sequential applications of concatenation and Boolean operations (i.e. union, intersection, complementation) to a basic class of languages, which constitutes level 0 of this hierarchy. Other levels of the hierarchy are obtained by alternating the so-called *polynomial closure* and the *Boolean closure*.

The *polynomial closure* of the class of languages $\mathcal{V}$ consists of all finite unions of languages of the form $L_0a_1L_1a_2\dots a_kL_k$ where $L_0, L_1, \dots L_k \in \mathcal{V}$ and $a_1, \dots, a_k \in A$ for any finite alphabet A. The *Boolean closure* of a class of languages is a closure under Boolean operations.

For every natural number n, level $n+\frac{1}{2}$ of the hierarchy is obtained by the polynomial closure of level n and level $n+1$ is obtained by the Booolean closure of level $n+\frac{1}{2}$. In this manner one constructs an infinite hierarchy of star-free languages.

If level 0 of a concatenation hierarchy creates the so-called *variety of regular languages* (i.e. is closed under certain "reasonable" operations) then the whole hierarchy can be built in terms of so-called *pseudovarieties of finite monoids*, which correspond to varieties of regular languages due to *Eilenberg's theorem*.

The *pseudovariety of monoids* is an analogy to the variety of monoids for the case when only finite monoids are considered. It's a class of *finite* monoids closed under submonoids, homomorphic images and *finite* direct products.

Another way to investigate such a hierarchy is in terms of the first-order logic over words. The levels of the hierarchy are obtained by an alternating use of an existential quantifier and Boolean combinations (i.e. disjunction, conjuction, negation) to formulas.

The main problem concerning concatenation hierarchies is for a given level of a given hierarchy to find an algorithm which enables to decide whether a given language belongs to this level. Such a level of a hierarchy is called *decidable*.

The simplest basis, level 0, is in the so-called *Straubing–Thérien hierarchy*. The level 0 of this hierarchy consists only of the empty language and the language A^* (i.e. the language of all words over A) for every finite alphabet A.

Up to now levels $0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, \frac{7}{2}$ of the Straubing–Thérien hierarchy have been known to be decidable. Some of these results are very recent. The decidability of levels 2 and $\frac{5}{2}$ was proven in 2014, the decidability of level $\frac{7}{2}$ was proven in 2015. All proofs of these latest results were done in terms of the first-order logic over words.

Coextensions of totally ordered monoids

Jiří Janda, Thomas Vetterlein

1. Introduction

A *totally ordered monoid* $(M; \cdot, \leq, 1)$ (or shortly a *tomonoid*) is an algebraic structure such that $(M; \cdot, 1)$ is a monoid and $\leq$ is a total order compatible with the operation $\cdot$. A homomorphism $f : M_1 \to M_2$ between tomonoids M_1, M_2 is an isotone homomorphism $f : M_1 \to M_2$ between underlining monoids M_1, M_2. We say that tomonoid $(M; \cdot, \leq, 1)$ is negative if 1 is the top element w.r.t. $\leq$. By a *coextension* of an algebraic structure A we mean the same type structure C such that A is a factorization of C, i.e. there exists a surjective homomorphism $f : C \to A$ (in some literature, C is denoted as an *extension*).

There exist classical approaches for studying particular classes of coextensions of semigroups (see e.g. [Gri], [CP], [Pet]). Our aim is to apply some of these approaches also on the ordered case, resp. the case of tomonoids. A motivation for studying tomonoids comes from the field of fuzzy logic, particularly from describing t-norms and finding new ways of their constructions.

2. Coextensions by a system of tomonoids

The type of coextension considered in this part has been motivated by so called *coextension by a direct system of groups* (G, γ) of a monoid S [Gri]. For the unordered case it encapsulates an idea of attaching a group G_a to every element $a \in S$ of the monoid S in a way that groups G_a form congruence classes of a coextension E of S. Using negative tomonoids M_a instead of groups G_a, we present an analogous construction for commutative tomonoids.

Theorem 1. *Let $(S; \cdot, \leq, 1)$ be a negative commutative tomonoid. Let (M, φ) be a direct system of commutative tomonoids over $(S, \leq_{\mathcal{H}})$, where $\leq_{\mathcal{H}}$ is the Green's order. Moreover, let $\sigma = (\sigma_{a,b})_{a,b \in S}$ where $\sigma_{a,b} \in M_{ab}$ for every $a, b \in S$.*

We define $E[S, M, \sigma]$ as a set of all ordered pairs (a, x), $x \in M_a, a \in S$ with operation given for every $(a, x), (b, y) \in E[S, M, \sigma]$ by

$$(a, x)(b, y) := (ab, \varphi^a_{ab}(x) + \varphi^b_{ab}(y) + \sigma_{a,b}) \tag{2.1}$$

and a relation $\leq_E$ lexicographically i.e.

$$(a, x) \leq_E (b, y) \text{ if } a \leq b \text{ or } a = b \text{ and } x \leq_a y. \tag{2.2}$$

Then $E[S, M, \sigma]$ is a negative tomonoid coextension of S if and only if the following conditions are satisfied for all $a, b, c \in S$

(M1) $\sigma_{a,b} = \sigma_{b,a}$,

(M2) $\varphi^{ab}_{abc}\sigma_{a,b} + \sigma_{ab,c} = \sigma_{a,bc} + \varphi^{bc}_{abc}\sigma_{b,c}$,

(M3) $\sigma_{1,a} = 0$,

(M4) *M_1 is negative tomonoid,*

(M5) *if $a < b$ and $ca = cb$, then $\varphi^a_{ca}(x) + \sigma_{a,c} \leq_{ca} \varphi^b_{cb}(y) + \sigma_{a,b}$.*

3. Ideal coextensions

An *ideal coextension* E of semigroup T is a coextension of T such that there exists an ideal $S \subseteq E$, $E/S \cong T$. This type of coextension has been constructively described for an arbitrary semigroup (e.g in [Pet]). More precisely, if we have a semigroup with zero T and a semigroup S, $S \cap T = \emptyset$, there has been given necessary and sufficient conditions for existence of an ideal coextension E of T such that $E = S \cup (T - \{0\})$ and $E/S \cong T$.

This result has been extended for the case of partially ordered semigroups by Kehayopulu and Tsingelis in [KT]. However, in their case the theorem has aside from the definition of operation 13 conditions.

A different approach has been chosen by Petrík and Vetterlein in [PV]. They gave the constructive description for all one element ideal extension E of a given finite negative tomonoid T.

Our aim is to use this two approaches together for finding a description of arbitrary finite coextensions of finite negative tomonoids.

References

[CP] A. H. Clifford, G. B. Preston *"The Algebraic Theory of Semigroups, Part 1"*, AMS, Providence, 1977.

[Gri] P. A. Grillet *"Semigroups: An Introduction to the Structure Theory"*, Marcel Dekker Inc, New York, 1995.

[KT] N. Kehayopulu, M. Tsingelis *Ideal Extensions of Ordered Semigroups*, Communications in Algebra, Vol. 31, No. 10, 4939–4969, 2003.

[Pet] M. Petrich *"Introduction to Semigroups"*, Bell & Howell Company, Columbus, 1973.

[PV] M. Petrík, T. Vetterlein *Rees Coextensions of Finite, Negative Tomonoids*, Journal of Logic and Computation, doi: 10.1093/logcom/exv047.