# SPRING SCHOOL 2009
# BOROVÁ LADA

Zuzka Safernová, Jan Volec (eds.)

# Preface

Spring school on Combinatorics has been a traditional meeting organized for faculty and students participating in the Combinatorial Seminar at Charles University for nearly 30 years. It is internationally known and regularly visited by students, postdocs and teachers from our cooperating institutions in the DIMATIA network. As it has been the case for several years, this Spring School is generously supported by the Institute of Theoretical Computer Science (ITI) of Charles University and the Department of Applied Mathematics (KAM) of Charles University.

The Spring Schools are entirely organized and arranged by our undegraduate students. The lecture subjects are selected by supervisors from the Department of Applied Mathematics (KAM) and Institute for Theoretical Computer Science (ITI) of Charles University as well as from other participating institutions. In contrast, the lectures themselves are almost exclusively given by students, both undergraduate and graduate. This leads to a unique atmosphere of the meeting which helps the students in further studies and their scientific orientation.

This year the Spring School is organized in Borová Lada, a mountain village in Šumava hills with a great variety of possibilities for outdoor activities like snow-shoe hiking or cross-country skiing.

Dan Král', Jan Kratochvíl, Jaroslav Nešetřil

# Index of Talk Titles

# Index of Speakers

# Linear programming

Pavel Klavík

pavel@klavik.cz

Presented paper by Jiří Matoušek, Bernd Gärtner

Transversals of $d$-Intervals

(*Understanding and Using Linear Programming*)

**Definition 1.** Let $\mathcal{F}$ be a family of sets. Set $T$ is called a *transversal of* $\mathcal{F}$ if it intersects all sets in $\mathcal{F}$.

Natural and interesting question for a family of sets is a *minimal size* of its transversal. What conditions guarantee that it has a small transversal? This is a complex question. In this speak we will prove the upper bound for size of a minimal transversal for a family of pairwise intersecting $d$-intervals.

**Definition 2.** For an integer $d \geq 1$ a *$d$-interval* is defined as the union of $d$ closed intervals on the real line.

Following upper bound is a surprising consequence of the duality theorem of linear programing.

**Theorem 3.** *Let $\mathcal{J}$ be a finite family of $d$-intervals such that $J_1 \cap J_2 \neq \emptyset$ for every $J_1, J_2 \in \mathcal{J}$. Then $\mathcal{J}$ has a transversal of size $2d^2$.*

At first sight it is not obvious that there exists any bound at all that depends only on $d$. The best known bound is $d^2$, and it has been established using algebraic topology. We will prove a bound worse only by a constant factor of 2, but the proof will be much simpler. Note that the known best lower bound is $\frac{d^2}{\log d}$.

Following two lemmas for the above defined family $\mathcal{J}$ imply the theorem:

**Lemma 4.** *There exists an endpoint of some $J_i$ that is contained in at least $\frac{n}{2d}$ of the $d$-intervals.*

**Lemma 5.** *Let $P$ denote the set of endpoints of the $d$-intervals in $\mathcal{J}$. Then there are nonnegative real numbers $x_p$, $p \in P$, such that for every $J \in \mathcal{J}$ is $\sum_{p \in J \cap P} x_p \geq 1$, and $\sum_{p \in P} x_p \leq 2d$.*

In the second part of the speak we will look at the proof in a general context. We define a transversal and a matching number of a family of sets. Sometimes it is hard to compute these numbers. We will introduce fractional transversals and matchings that can be easily computed by linear programming.


Jan Volec

janv@jikos.cz


Presented paper by Jiří Matoušek, Bernd Gärtner

## Sparse Solutions of Linear Systems

(*Understanding and Using Linear Programming*)

**Definition 1.** Let $w \in \mathbb{R}^k$ be a sequence of $k$ real numbers which we would like to encode into some code $z \in \mathbb{R}^n$ such that if at most $\lfloor 0.08n \rfloor$ of numbers become corrupted, we can still recover $z$ (and from them the original sequence). This a little bit corrupted vector we call $z'$.

**Definition 2.** $Q$ is an encoding matrix $n \times k$ of rank $k$ and we encode any $w \in \mathbb{R}^k$ into $z = Qw \in \mathbb{R}^n$.

**Definition 3.** Let $x$ be error vector defined as $z' - z$. The number of nonzero coordinates in $x$ is $r$ and $r \leq \lfloor 0.08n \rfloor$.

Now we try for given $z'$ find its error vector $x$ and if we are lucky then we simply recover $z$ using $z'$ and $x$.

**Definition 4.** Let $m = n - k$ and $A$ is $m \times n$ matrix such that $AQ = 0$.

Now we can reformulate our problem into problem called Sparse solutions of undetermined system of linear equations. The problem is for $m \times n$ matrix $A$ $(m < n)$, vector $b \in \mathbb{R}^m$ and integer $r$ find vector $x \in \mathbb{R}^n$ such that number of nonzero coordinates in $x$ is at most $r$ and holds $Ax = b$.

**Observation 5.** *With fixed $n, m, r$, the following two conditions are equivalent:*

- *For every $b$ the system $Ax = b$ has at most one sparse solution $x$.*

- *Every $2r$ (or fewer) columns of $A$ are lineary independent.*

Now we have exact characterization of $x$, but there is still open question – how fast can we find $x$? Generally it is NP-hard, but in our situation we are lucky because we can reformulate our problem into linear programming and the optimum of program will be our vector $x$.

**Definition 6.** Basis pursuit (BP) is program minimizing $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$ subject to $Ax = b$ and $x \in \mathbb{R}^n$ and this program could be easily reformulated as linear program (BP').

**Theorem 7.** *For $k$ suffciently large integer, let us set $n = 4k, m = 3k$, let $A$ be a random $m \times n$ matrix, $Q$ is defined as ortogonal complement of $A$. Then the following holds with probability close to 1: if $Q$ is used as encoding matrix, then even if any at most 8% of entries of $z$ are corrupted, we can still recover $z$ efficienty by solving (BP).*

Martin Kupec

magon@jkopava.cz

Presented paper by Jiří Matoušek, Bernd Gärtner

## Smallest Ball and Convex Programming

(*Understanding and Using Linear Programming*)

**Definition 1.** Let $p_1, p_2, ..., p_n \in \mathbb{R}^d$ be points. Task is to find a ball of the smallest radius that contains all the points.

**Definition 2.** A convex program is the problem of minimizing a convex function in n variables subject to linear equality and inequality constraints.

**Theorem 3.** *Karush–Kuhn–Tucker conditions*

*Let us consider the convex program*

$$\begin{array}{ll} \textit{minimize} & f(x) \\ \textit{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

*with f convex and differentiable, with continuous partial derivatives. A feasible solution $x^* \in \mathbb{R}^n$ is optimal if and only if there is a vector $\tilde{y} \in \mathbb{R}^m$ such that for all $j \in \{1, ..., n\}$,*

$$\nabla f(x^*)_j + \tilde{y}^T a_j \begin{cases} = 0 & \text{if } x_j^* > 0 \\ \geq 0 & \text{otherwise.} \end{cases}$$

**Lemma 4.** *Let $S = \{s_1, ..., s_k\} \subseteq \mathbb{R}^d$ be a set of points on the boundary of a ball $B$ with center $s^* \in \mathbb{R}^d$. $B$ is the unique smallest enclosing ball of $S$ if and only if for every $u \in \mathbb{R}^d$, there is an index $j \in \{1, 2, ..., k\}$ such that $u^T(s_j - s^*) \leq 0$.*

Using Karush–Kuhn–Tucker condition and the lemma above we can state Smallest enclosing ball problem as convex program as folows.

**Theorem 5.** *Let $p_1, ..., p_n \in \mathbb{R}^d$ be points, and let $Q$ be the $d \times n$ matrix whose j-th column is formed by the $d$ coordinates of the point $p_j$ . Let us consider the optimization problem*

$$\begin{array}{ll} \text{minimize} & x^T Q^T Q x - \sum_{j=1}^n x_j p_j^T p_j \\ \text{subject to} & \sum_{j=1}^n x_j = 1 \\ & x \geq 0 \end{array}$$

*in the variables $x_1, ..., x_n$. The objective function*

$$f(x) = x^T Q^T Q x - \sum_{j=1}^{n} x_j p_j^T p_j$$

*is convex and $x^*$ is the optimal solution. Then there exist point $p^* = Qx^*$ and ball with center $p^*$ and squared radius $-f(x^*)$ which is the unique ball of smallest radius containing $P$.*


Pilsner Urquell
Pilsner Urquell

# Constraint Satisfaction Problem

Tomáš Gavenčiak

gavento@kam.mff.cuni.cz

Constraint Satisfaction Problem tutorial

### Introduction

The *Constraint Satisfaction Problem* (abbreviated as *CSP*) naturally arises as a generalization of many well-studied problems in computational complexity. A CSP problem is given by a finite set of *variables* and a common (finite or countable) *domain* of their values, together with a finite number of constraints. A constraint is a relation on a subsets of variables. The goal is to find a *solution*, that is an assignment of values to variables satisfying all the given constraints.

CSP captures three important aspects of problems at once, as the problems easily described by CSP include:

- *Logical problems:* 3-SAT, NAE-SAT, 2-SAT, ...
- *Graph problems:* $k$-colorability, $\mathcal{L}(2,1)$-labelling, ...
- *Algebraic problems:* Existence of homomorphism between two structures

While some of these problems are in **P** (2-SAT), others are **NP**-complete (3-SAT). One of the big questions in CSP is the complexity of CSP under given restrictions on the constraints.

The algorithms efficiently solving the restricted or the general CSP are widely used in real-world applications, both in industry and research. The examples include: finding efficient layouts, scheduling, resource distribution, fitting phenomena to measurements, solving logical puzzles, and many more.

The design of such algorithms must exploit the structure of the constraints to reduce the (generally exponential) search time and to solve many of the polynomial cases in polynomial time.

The tutorial is divided into three parts. The first part introduces basic concepts of CSP and presents some complexity results. The later parts focus on algebraic properties of CSP and deeper results of algebras corresponding to CSP problem classes.

### Definitions and facts

**Definition 1.**[CSP as a *Value assignment*] Given a finite set of variables $V$, a (finite) domain $D$ and a set of constraints $C$, where a constraint is a pair $(\overline{s_i}, R_i)$ of a *scope* (ordered subset of variables) and a relation, is there an assignment of values $f : V \to D$ such that $f[\overline{s_i}] \in R_i$ for all $i$?

**Definition 2.**[CSP as a *Conjunctive query*] Given a formula $\Phi(x_1, x_2, \ldots, x_k) = R_1(\overline{s_1}) \wedge \cdots \wedge R_q(\overline{s_q})$, where the relations are seen as predicates and $\overline{s_i}$ are variable vectors, is $\Phi$ satisfiable?

**Definition 3.**[CSP as a *Structure homomorphism*] Given two structures $\mathcal{A} = (V; S_1, S_2, \ldots S_k)$ and $\mathcal{B} = (D; R_1, R_2, \ldots R_k)$, is there a homomorphism $h : \mathcal{A} \to \mathcal{B}$?

**Lemma 4.** *The three types of CSP problems are polynomially equivalent.*

**Proof** The transformations between value assignment CSP and structure homomorphism CSP are quite straightforward, the reduction to conjunctive query CSP requires the encoding of the values as $\lceil \log_2 |D| \rceil$-tuples of bits, or some other reduction. □

From this point the tutorial focuses on the homomorphism CSP, although the constraints and most of the results have their counterparts in the two other definitions as well.

We *parametrize* the CSP with limitations on the constraints. The restrictions may affect either the *left* side of the homomorphism (which of the "variables" $V$ appear in common constraints), or the *right* side (which "value" combinations of $D$ are allowed for the constrained "variables").

2-SAT and HORN-SAT are nice examples of right-side restrictions that make the problem polynomial regardless of the structure of the given formula. One example of a left-side restriction for graph CSP problems that makes CSP polynomial is the restriction to trees. Note that finding a graph homomorphism from a tree (input) to a (fixed) graph $G$ is decidable in time linear with the size of the tree.

**Definition 5.** For given structures $\mathcal{A}$ and $\mathcal{B}$, $CSP(\mathcal{A}, \mathcal{B})$ denotes the CSP structure homomorphism problem from $\mathcal{A}$ to $\mathcal{B}$.

For classes of structures $\mathbb{C}$ and $\mathbb{D}$, $CSP(\mathbb{C}, \mathbb{D})$ denotes class of all homomorphism problems $CSP(\mathcal{A}, \mathcal{B})$ for $\mathcal{A} \in \mathbb{C}$ and $\mathcal{B} \in \mathbb{D}$.

We let $CSP(\mathbb{C}, -)$ denote the class of problems with only left-side restrictions and $CSP(\mathbb{D}) = CSP(-, \mathbb{D})$ the class with only right-side restrictions.

**Classification of restrictions**

The theorem below states the border between polynomial and non-polynomial left-side restrictions.

**Definition 6.** Two structures $\mathcal{A}$ and $\mathcal{B}$ are *homomorphically equivalent* if there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ and a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. A class $\mathbb{C}$ has bounded tree-width modulo homomorphic equivalence if there is a $k$ such that every structure in $\mathbb{C}$ is homomorphically equivalent to a structure of tree-width at most $k$.

Please refer to the original paper *Constraint satisfaction, bounded tree-width, and finite-variable logics*, Dalmau et al., LNCS 2002 for further details.

**Theorem 7.** *[Dalmau et al. 2002] Assume that $FPT \neq W[1]$. Then for every recursively enumerable class $\mathbb{C}$ of structures of bounded arity, $CSP(\mathbb{C}, -)$ is in polynomial time if and only if $\mathbb{C}$ has bounded tree-width modulo homomorphic equivalence.*

The right-side restrictions are in general more complex and there are still many open problems in the area.

**Definition 8.** A *constraint language* $\mathcal{L}$ is a finite set of relations on $D$, the problem $CSP(\mathcal{L})$ is the $CSP(-, \mathcal{B})$ where $\mathcal{B} = (D; R_i, \dots)$ is the structure containing the relations from $\mathcal{L}$ in some order.

Two types of classifications of $CSP(\mathcal{L})$:

- Computational complexity – decidability of existence of homomorphism from given structure.
- Descriptive complexity – definability in a given logic on the left-side structures.

**Dichotomy conjecture.** *[Feder, Vardi, SIAM 1998] For each constraint language $\mathcal{L}$, the problem $CSP(\mathcal{L})$ is either in* **P** *or it is* **NP***-complete.*

The motivation for the conjecture is that $\mathbf{P} \neq \mathbf{NP}$ implies that there are many problem classes between **P** and **NP**-complete *[Ladner, J.ACM 1975]*. The conjecture implies that neither of them is a CSP.

Pavel Paták

ppatak@seznam.cz

Presented paper by Andrei Krokhin

# The Complexity of Constraint Satisfaction Problems - Tutorial, Part II

(http://www.karlin.mff.cuni.cz/~ssaos)

### Introduction

In this paper we introduce the idea that stands beyond the algebraic aproach to CSP. We give same basic definitions, e.g relational and operational clones, polymorphisms, constraint languages and some more. Later we will show why the relation clones are so important in deciding whether there exists a homomorphism to a given structure.

In the second part of the talk we will introduce further algebraic notation and discuss some algorithmic reductions of CSP.

### More formally

**Definition 1.** For a set $\Gamma$ on a structure $D$, denote $\langle\Gamma\rangle$ the set of all relations that can be expressed by primitive positive formulas over $\Gamma$, e.g. using only relations from $\Gamma$, conjuctions and existencional quantifier.
We call the set $\langle\Gamma\rangle$ the relational clone generated by $\Gamma$.
For a set of operations $F$ on $D$ let us denote $\langle F\rangle$ the operational clone generated by $F$, i.e. the set of all functions obtained from $F$ via superpositions $f(f_1, \ldots, f_n)$.

**Theorem 2.** *If $\Gamma_1$ and $\Gamma_2$ are constraint languages such that $\langle\Gamma_1\rangle \subseteq \langle\Gamma_2\rangle$, then CSP$(\Gamma_1)$ is logspace reducible to CSP$(\Gamma_2)$.*

**Definition 3.** An $n$-ary operation $f$ is a polymorphism, iff it is a homomorphism from $A^n$ to $A$.

**Definition 4.** Let $\Gamma$ be a set of relations on $D$ and $F$ be a set of operations, then we denote:
Pol$(\Gamma) \equiv$ the set of all polymorphism on $\Gamma$.
Inv$(F) \equiv \{R|R$ is invariant under all operations in F$\}$.

**Theorem 5.** *For any constraint language and for any set $F$ of operation the following is true:*

1. $\langle\Gamma\rangle = Inv(Pol(\Gamma))$
2. $\langle F\rangle = Pol(Inv(\Gamma))$.

*We say that the these operations form a Galois Correspondence.*

**Corollary 6.** *We can decide that a problem $X$ is* ***NP****-complete by constructing a reduction to $X$ from some* ***NP****-complete problem.*
*But all we need to show is $Pol(\Gamma) \subseteq Pol(\Gamma')$, for some $\Gamma'$ with* ***NP****-complete CSP$(\Gamma')$*

For an unary operation $f$ and a relation $R$ on $D$, let

$$f(R) = \{f(a_1), f(a_2), \ldots, f(a_n)|(a_1, a_2, \ldots a_n) \in R\}.$$

For a constraint language $\Gamma$ let $f(\Gamma) = \{f(R)|R \in \Gamma\}$.

**Theorem 7.** *Let $\Gamma$ be finite, and let $f \in Pol(\Gamma)$ be unary with minimal range. Then CSP$(\Gamma)$ and CSP$(f(\Gamma))$ are polynomial time equivalent.*

**Theorem 8.** *If the decision problem CSP$(\Gamma)$ is in P, than the corresponding search problem is in P as well.*

**Definition 9.** A finite algebra is a pair $\mathbf{A} = (D, F)$, where $D$ is a finite set and $F$ is a family of operations on $D$.
The clone $\langle F\rangle$ is called the clone of term operations of $A$.
Two algebras are **term equivalent** if they have the same clone of term operations.

**Definition 10.** Let $\mathbf{A} = (D, F)$ be a finite algebra. Let $\mathrm{CSP}(\mathbf{A}) = \{\mathrm{CSP}(\Gamma) | \Gamma \subseteq \mathrm{Inv}(F), |\Gamma| < \infty\}$.
We say that $\mathbf{A}$ is **tractable** if each problem in $\mathrm{CSP}(\mathbf{A})$ is in P, and $\mathbf{A}$ is **NP**-complete, if some problem in $\mathrm{CSP}(\mathbf{A})$ is in NP.

**Definition 11.** For a class $\mathcal{K}$ of similar algebras, let

- $\mathtt{H}(\mathcal{K})$ be the class of all homomorphic images of algebras from $\mathcal{K}$
- $\mathtt{S}(\mathcal{K})$ be the class of all subalgebras of algebras from $\mathcal{K}$
- $\mathtt{P}(\mathcal{K})$, resp. $\mathtt{P}_{fin}(\mathcal{K})$ be the class of all direct, resp. finite products of algebras from $\mathcal{K}$

A class of similar algebras that is closed under the operators $\mathtt{H}, \mathtt{S}, \mathtt{P}$ is called a variety.
For an algebra $\mathbf{A}$ the class $\mathtt{HSP}(\mathbf{A})$ is the variety generated by $\mathbf{A}$, and is denoted $\mathtt{v}ar(\mathbf{A})$.

**Theorem 12.** *If an algebra* $\mathbf{A}$ *is tractable than every finite algebra in* $\mathtt{var}(\mathbf{A})$ *is tractable. If* $\mathtt{var}(\mathbf{A})$ *contains a finite* ***NP****-complete algebra then* $\mathbf{A}$ *is* ***NP****-complete.*

## References

[1] Andrei Bulatov, Peter Jeavons, Andrei Krokhin: Classifying the complexity of constraints using finite algebras

[2] A. Krokhin: The Complexity of Constraint Satisfaction Problems, Tutorial to Summer School on Algebra

Alexandr Kazda

alexandr.kazda@seznam.cz

Presented paper by Andrei Krokhin

# The Complexity of Constraint Satisfaction Problems, Part III

(http://www.karlin.mff.cuni.cz/~ssaos/CSPTutorialPartIII.pdf)

**Summary**

We finish the overview of algebraic approach to CSP by presenting several starting points for further study.

We introduce Taylor and WNU operations and show links with combinatorial results about graph homomorphism CSP. In particular, we note that a well-known combinatorial result of Bang-Jensen and Hell about CSP($H$) can be derived as a corollary of an algebraic result by Barto, Kozik and Niven.

Taking off in another direction, we present the Datalog language for solving certain kinds of CSP. This kind of reasoning leads to the bounded width theory.

While we do not have time to develop the whole Tame Congruence Theory, we offer a taste of it, describing (in a broad, hand-waving fashion) the complexity of certain kinds of CSP using only algebraic properties of their lattices.

**Major results**

**Definition 1.** An $n$-ary operation $f$ on an algebra $A$ is called a *Taylor operation* if $f$ is idempotent (ie. $f(x, \dots, x) = x$) and for all $1 < i \leq n$, $f$ satisfies some identity of the form

$$f(x_{i1}, \dots, x_{in}) = f(y_{i1}, \dots, y_{in})$$

where $x_i j, y_i j \in \{x, y\}$ (as symbols) and $x_i i \neq y_i i$.

**Definition 2.** An $n$-ary operation $f$ on an algebra $A$ is called *WNU operation* if:

- $f$ is idempotent (ie. $f(x, \ldots, x) = x$.
- $f$ satisfies the identity
$f(y, x, \ldots, x) = f(x, y, \ldots, x) = \cdots = f(x, x, \ldots, y)$.

**Theorem 3.** *(Bang-Jensen, Hell, 1990) If $H$ is a core digraph without sources or sinks that is not a disjoint union of directed cycles then $CSP(H)$ is NP-complete.*

**Theorem 4.** *(Barto, Kozik, Niven, 2008) If $H$ is a core digraph without sources or sinks. If $H$ has a WNU polymorphism then $H$ is a disjoint union of directed cycles.*

**Theorem 5.** *For a core structure $B$ with an associated idempotent algebra $A$ the following is true:*

| $var(A)$ | | $CSP(B)$ | $co - CSP(B)$ |
|---|---|---|---|
| *omits* | *admits* | *complexity* | *definability* |
| | *1* | *NP-complete* | *not Datalog* |
| *1* | *2* | *$mod_p$L-hard* | *not Datalog* |
| *1,2* | *5* | *P-hard* | *not Linear Datalog* |
| *1,2,5* | *4* | *NL-hard* | *not Symmetric Datalog* |

.


Václav Brožek

99081@mail.muni.cz


Presented paper by A. Bulatov, P. Jeavons and A. Krokhin

# Classifying the Complexity of Constraints Using Finite Algebras

( www.dur.ac.uk/andrei.krokhin/papers/SIAMclassifying.pdf )

### Introduction and Motivation

The paper concerns tractability of CSP problems. It starts with a CSP problem given by a set (language) of relations which are used to specify constraints. Then it shows how to focus only on a special kind of such languages, *relational clones*. From relational clones it transfers to sets of operations on the domain set of the relations and finally to algebras. Some of interesting results from the paper include:

- Sufficient and necessary conditions for tractability of a CSP problem in terms of its associated algebra.
- Classification of tractability for CSP with a special kind of associated algebras, generalizing earlier classification results.
- Tractability of decision CSP problem implies tractability of its search variant.

The work in this paper is motivated by earlier results, e.g. Schaefer's dichotomy for CSP over two-point sets. The methods used here connect nicely CSP with universal algebra and lead to promising generalisations. However, some interesting problems remain still open.

### Definitions

It is assumed that $\mathbf{P} \neq \mathbf{NP}$ in this paper. Let $A$ be a set. A finitary relation over $A$ is a subset of $A^n$ for some finite $n$.

**Definition 1.** (CSP) A (finite or infinite) set $\Gamma$ of finitary relations over a set $A$ (*domain*) is called a *constraint language*. By $\mathbf{CSP}(\Gamma)$ we denote the following decision problem:
**Instance:** A triple $(V, A, \mathcal{C})$ where:

- $V$ is a finite set of variables;
- $\mathcal{C}$ is a finite set of constraints $\{C_1, \ldots, C_q\}$ where $C_i = \langle s_i, \varrho_i \rangle$ and
  - $s_i$ is a tuple of variables of length $m_i$ called the *constraint scope*
  - $\varrho_i \in \Gamma$ is a $m_i$-ary relation called the *constraint relation*

**Question:** Is there a *solution*, i.e. a function $\varphi : V \to A$ such that for each $\langle (x_1, \ldots, x_m), \varrho \rangle \in \mathcal{C}$: $(\varphi(x_1), \ldots, \varphi(x_m)) \in \varrho$?
By a *search variant* of $\mathbf{CSP}(\Gamma)$ we denote the problem of finding $\varphi$.

**Definition 2.** (Relational clone) A constraint language $\Gamma$ is called a *relational clone* if it contains every predicate expressible by a FO formula involving

$$\begin{array}{ccccc}
(a_{11}, & a_{21}, & \cdots & a_{n1}) & \in \varrho \\
(a_{12}, & a_{22}, & \cdots & a_{n2}) & \in \varrho \\
\vdots & \vdots & \ddots & \vdots & \\
(a_{1m}, & a_{2m}, & \cdots & a_{nm}) & \in \varrho \\
\downarrow & \downarrow & & \downarrow & \\
(f(a_{11},\ldots,a_{1m}), & f(a_{11},\ldots,a_{1m}), & \cdots & f(a_{n1},\ldots,a_{nm})) & \in \varrho
\end{array}$$

Figure 1: The relation $\varrho$ is preserved by $f$.

1. relations (predicates) from $\Gamma$ and the identity predicate $=_A$ on the domain;
2. conjunction; and
3. existential quantification.

For every constraint language $\Gamma$ the least relational clone containing $\Gamma$ exists and is denoted $\langle\Gamma\rangle$.

A finitary operation on $A$ is a function $A^n \to A$ for some $n$.

**Definition 3.** (Polymorphism) Let $f$ be an $m$-ary operation and $\varrho$ an $n$-ary relation on $A$. Then we say that $f$ preserves $\varrho$ (or that $f$ a polymorphism of $\varrho$ or that $\varrho$ is invariant under $f$) if whenever $(a_{11},\ldots,a_{n1}),\ldots,(a_{1m},\ldots,a_{nm}) \in \varrho$ then $(f(a_{11},\ldots,a_{1m}),\ldots,f(a_{n1},\ldots,a_{nm})) \in \varrho$. See also Figure 1.

Further, we define

$$\begin{aligned}
Pol(\Gamma) &= \{f \mid f \text{ preserves each } \varrho \in \Gamma\} \\
Inv(F) &= \{\varrho \mid \varrho \text{ is invariant under each } f \in F\}
\end{aligned}$$

**Definition 4.** (Essentially unary operation) An operation $f : A^n \to A$ is called *essentially unary* if there are a nonconstant unary operation $g : A \to A$ and an index $i$, $1 \le i \le n$ such that $f(a_1, a_2, \ldots, a_n) = g(a_i)$ for all choices of $a_1, \ldots a_n$. If $g$ is identity then $f$ is called a *projection*.

**Definition 5.** (Algebra)

- A tuple $\mathcal{A} = (A, F)$ where $A$ is a nonempty set and $F$ is a set of some finitary operations on $A$ is called *algebra*.
- Every operation $f$ on $A$ is called *term operation* of $\mathcal{A}$ if $f \in Pol(Inv(F))$.
- $\mathcal{A}$ is *surjective* if all of its term operations are surjective.
- An operation $f$ is called *idempotent* if $f(x, \ldots, x) = x$ for all $x \in A$. The set of all idempotent term operations of $\mathcal{A}$ is denoted $Term_{id}(\mathcal{A})$. The algebra $(A, Term_{id}(\mathcal{A}))$ is then called the *full idempotent reduct of* $\mathcal{A}$.
- A *subalgebra* of $\mathcal{A}$ is a pair $(B, G)$ such that $\emptyset \neq B \subseteq A$, for every $f \in F$ the restriction $f\restriction_B$ is an operation on $B$, and $G = \{f\restriction_B \mid f \in F\}$.
- Assume that there is a surjective map $\varphi : A \to B$ such that the following prescription yields and operation on $B$ for every $f : A^n \to A \in F$:

  $$f_\varphi : (\varphi(a_1), \ldots, \varphi(a_n)) \mapsto (\varphi \circ f)(a_1, \ldots, a_n)$$

  i.e. if $f$ preserves $\ker \varphi$. Then $\varphi$ is a *homomorphism* of $\mathcal{A}$ to $(B, G)$, where $G = \{f_\varphi \mid f \in F\}$ and $(B, G)$ is a *homomorphic image* of $\mathcal{A}$.
- $\mathcal{A}$ is called *simple* if $A$ is finite and every homomorphic image $(B, G)$ of $\mathcal{A}$ satisfies: $|B| < |A| \Rightarrow |B| = 1$. A simple $\mathcal{A}$ is called *strictly* simple if every its subalgebra $(B, G)$ satisfies $B \neq A \Rightarrow |B| = 1$.

**Definition 6.** (Tractability) A finite constraint language $\Gamma$ is *tractable* if the decision problem $\mathbf{CSP}(\Gamma)$ is solvable in polynomial time, i.e. $\mathbf{CSP}(\Gamma) \in \mathbf{P}$. An arbitrary constraint language $\Gamma$ is *tractable* if $\Delta$ is tractable for every finite $\Delta \subseteq \Gamma$. If there is a finite $\Delta \subseteq \Gamma$ such that $\mathbf{CSP}(\Delta)$ is $\mathbf{NP}$-complete then $\Gamma$ is called $\mathbf{NP}$-*complete*.

A set of operations $F$ is called *tractable*, or $\mathbf{NP}$-*complete* if $Inv(F)$ is tractable, or $\mathbf{NP}$-complete, respectively. An algebra $(A, F)$ is called *tractable*, or $\mathbf{NP}$-*complete* if $F$ is tractable, or $\mathbf{NP}$-complete, respectively.

## Results

Some of the results listed do not originate from this paper but they are important for capturing the whole picture.

**Theorem 7.** *(Reduction to clones) A constraint language* $\Gamma$ *is tractable, or* **NP***-complete iff* $\langle\Gamma\rangle$ *is tractable, or* **NP***-complete respectively.*

**Theorem 8.** *(Reduction to algebra) Relational clones are precisely the sets of the form* $Inv(F)$ *for a suitable set* $F$ *of operations. In particular,* $\langle\Gamma\rangle = Inv(Pol(\Gamma))$ *for every* $\Gamma$.

**Theorem 9.** *(Criterion for* **NP***-completeness) If every* $f \in Pol(\Gamma)$ *is esentially unary then* $\Gamma$ *is* **NP***-complete.*

**Theorem 10.** *(Reduction to surective algebra) For every algebra* $(A, F)$ *there is some* $B \subseteq A$ *and a set of operations* $G$ *consisting of all* $f \in Pol(Inv(F))$ *preserving* $B$ *such that* $(B, G)$ *is surjective, and* $(A, F)$ *is tractable, or* **NP***-complete iff* $(B, G)$ *is tractable, or* **NP***-complete respectively.*

**Theorem 11.** *(Reduction to full idempotent reduct) For every algebra* $\mathcal{A}$ *is true that* $\mathcal{A}$ *is tractable, or* **NP***-complete iff its full idempotent reduct is tractable, or* **NP***-complete respectively.*

*As a consequence, a constraint language* $\Gamma$ *is tractable iff the search variant of* $\mathbf{CSP}(\Gamma)$ *can be solved in polynomial time.*

**Theorem 12.** *(Reduction to subalgebra and homomorphic image) For every algebra* $\mathcal{A}$ *is true that*

- *If* $\mathcal{A}$ *is tractable then every subalgebra and homomorphic image of* $\mathcal{A}$ *is tractable.*
- *If some subalgebra or homomorphic image of* $\mathcal{A}$ *is* **NP***-complete then* $\mathcal{A}$ *is* **NP***-complete.*

**Theorem 13.** *(Dichotomy for fsss algebras) A finite strictly simple surjective algebra is* **NP***-complete if all of its term operations are essentially unary. Otherwise it is tractable.*

This result, extending previously known dichotomy for two-point algebras, points towards a more general question:

**Conjecture 14.** *A finite idempotent algebra* $\mathcal{A}$ *is* **NP***-complete if there is* *some algebra $(B, G)$ such that $|B| > 1$, $(B, G)$ is a homomorphic image of some subalgebra of $\mathcal{A}$ and all operations in $G$ are projections. Otherwise $\mathcal{A}$ is tractable.*

Jakub Bulín

jakub.bulin@gmail.com

## CSPs of Bounded Width

### Introduction

Given a finite relational structure $\mathbb{A}$, CSP($\mathbb{A}$) is the following decision problem:

INPUT: A structure $\mathbb{X}$ of the same type as $\mathbb{A}$.

QUESTION: Is there a homomorphism from $\mathbb{X}$ to $\mathbb{A}$?

There are two main polynomial time algorithms (or algorithmical approaches) which solve large classes of CSPs. One of them generalizes the Gaussian elimination and can be used for CSPs with so-called "few subpowers". The other one, the Local consistency checking algorithm, works for CSPs of bounded width. We will concentrate on the latter. It is believed that all tractable CSPs can be solved by a certain combination of these two algorithms.

We will define bounded width via the notion of $(k, l)$-strategy and present the $(k, l)$-consistency checking algorithm. Then we will introduce the recently proved Bounded width conjecture of B. Larose and L. Zádori and explore the class of CSPs of bounded width.

### Bounded width

Let $\mathbb{A} = \langle A, \mathrm{R}_1, \ldots, \mathrm{R}_n \rangle$ be a fixed finite relational structure. Bounded width can be defined in several ways (bounded tree width duality, solvability in Datalog, pebble games). We will introduce the $(k, l)$-strategies:

**Definition 1.** Let $\mathbb{X}$ be a structure of the same type as $\mathbb{A}$ and let $k \leq l$ be positive integers. A family $\mathcal{F} = \bigcup_{L \subseteq X, |L| \leq l} \mathcal{F}_L$ of partial homomorphisms from $\mathbb{X}$ to $\mathbb{A}$ is called a $(k,l)$-*strategy* for $(\mathbb{X},\mathbb{A})$, if it satisfies the following:

(S0) $\mathrm{dom}(f) = L$ for each $f \in \mathcal{F}_L$.

(S1) For any $f \in \mathcal{F}_L$ and $K \subseteq L$ the function $f|_K$ belongs to $\mathcal{F}_K$.

(S2) If $K \subseteq L \subseteq X$ with $|K| \leq k$, $|L| \leq l$ and $f \in \mathcal{F}_K$, then there exists $g \in \mathcal{F}_L$ such that $g|_K = f$.

It is easy to see that if there is a homomorphism from $\mathbb{X}$ to $\mathbb{A}$, then there exists a nonempty $(k,l)$-strategy. $\mathbb{A}$ is said to have *bounded width*, if the converse is true for some $k \leq l$:

**Definition 2.** $\mathbb{A}$ has *relational width* $(k,l)$ if the following is true: For each $\mathbb{X}$, if there exists a nonempty $(k,l)$-strategy for $(\mathbb{X},\mathbb{A})$, then $\mathbb{X}$ is homomorphic to $\mathbb{A}$.

We say that $\mathbb{A}$ has *bounded width* if it has relational width $(k,l)$ for some $k \leq l$.

We will now introduce the Local consistency checking algorithm, which solves CSPs of bounded width in polynomial time:

### Local Consistency Checking

**Lemma 3.** "We can construct the biggest $(k,l)$-strategy in P-time."

**Proof** The idea of the $(k,l)$-consistency checking algorithm is simple: Take all partial homomorphisms from $\mathbb{X}$ to $\mathbb{A}$ with at most $l$-element domain. Then throw away those which falsify conditions (S1) or (S2). We end up with the biggest $(k,l)$-strategy, which is nonempty iff there is a homomorphism from $\mathbb{X}$ to $\mathbb{A}$.

**Input:** A structure $\mathbb{X}$ of the same type as $\mathbb{A}$.
**Initial step:** For each $L \subseteq X, |L| \leq l$ let $\mathcal{F}_L :=$ all partial homomorphisms from $\mathbb{X}$ to $\mathbb{A}$ with domain $L$.
**Iteration step:** If there exist $f \in \mathcal{F}$ falsifying (S1) or (S2), remove $f$ from $\mathcal{F}$.
**Output:** If $\mathcal{F} = \emptyset$, return NO, else return YES. □

### The Larose-Zádori Conjecture

To each relational structure $\mathbb{A}$ is naturally associated an algebra

$$\mathbf{A} = \langle A, \{\text{idempotent polymorphisms of } \mathbb{A}\}\rangle.$$

It is crucial to the algebraic approach to CSP that the complexity of CSP($\mathbb{A}$) depends only on the variety generated by $\mathbf{A}$. It is the same case with the bounded width:

A *reduct of a module* is an algebra whose term operations can be expressed as $\alpha_1 x_1 + \cdots + \alpha_n x_n$, where $\alpha_i \in R$ and $x_i \in M$ for some module $M$ over a ring $R$.

**Theorem 4.**(Larose, Zádori '06). *If HSP($\mathbf{A}$) contains a reduct of a module, then $\mathbb{A}$ doesn't have bounded width.*

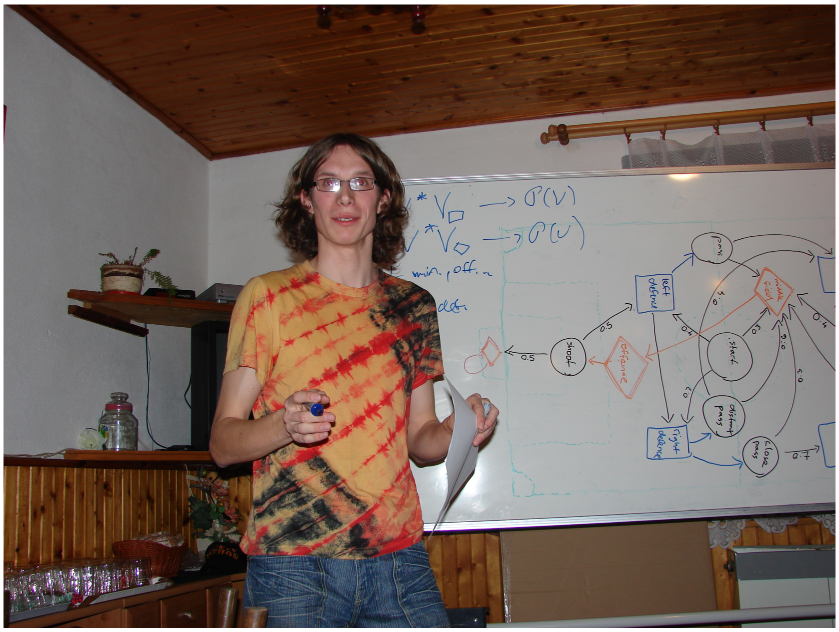
**Conjecture 5.**(Larose, Zádori '06). *$\mathbb{A}$ has bounded width iff HSP($\mathbf{A}$) doesn't contain a reduct of a module ( $\Leftrightarrow$ HSP($\mathbf{A}$) omits types 1 and 2).*

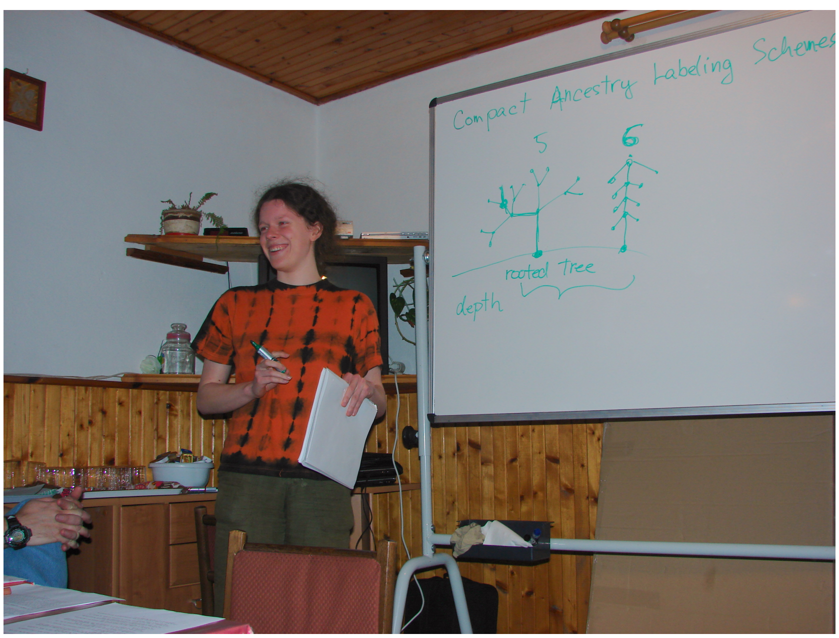
**Theorem 6.**(Barto, Kozik '08). *YES!*
*If HSP($\mathbf{A}$) doesn't contain a reduct of a module, then $\mathbb{A}$ has relational width $(2\lceil \frac{p}{2} \rceil, 3\lceil \frac{p}{2} \rceil)$, where $p$ is the maximal arity of relations in $\mathbb{A}$.*

**References**

[1] L. Barto, M. Kozik: Constraint Satisfaction Problems of Bounded Width, in progress.

[2] B. Larose and L. Zádori: Bounded width problems and algebras. Algebra Universalis, 56(3-4), 439-466, 2007.




Compact Ancestry Labeling Schemes
5
6
rooted tree
depth

# Standalone Talks

Ondřej Bílka

neleai@seznam.cz

Presented paper by Serguei Norine

## On two questions about circular choosability

(http://www.math.princeton.edu/ snorin/papers/oncircchoose.pdf)

### Introduction

In this talk we present first half of article which answers question of Zhu that circular list chromatic number of even cycles is two.

### Circular list choosability

**Definition 1.**

By (p,q)-coloring of G we mean function $f : V- > \{0, 1, \ldots, p-1\}$ such that for each edge $uv$ of $G$ is $q < |f(u) - f(v)| < p - q$

**Definition 2.**

Circular chromatic number is

$$\chi_c(G) = inf\{p/q | \text{G admits (p,q)-coloring}\}$$

**Definition 3.**

For $t \geq 1$ graph $G$ is circulary t-$(p, q)$-choosable if for any list assignment $L(v) \subset \{0, 1, \ldots, p-1\}$ with $L(v) \geq tq$ admits circular $(p, q)$-coloring with these lists.

**Definition 4.**

Circular choosability is defined as

$$\chi_{c,l}(G) = inf\{t | \text{G is t-(p,q)-choosable for any p,q}\}$$

We observe that circular choosability is not less that circular chromatic number.

### Circular list choosability of even cycles

We use combinatorial nullstellensatz:

**Theorem 5.**

*Let $F$ be field and $p$ polynomial in $F[x_1, x_2, \dots, x_n]$. Suppose degree $deg(p)$ is $\sum_{i=1}^{n} d_i$ and suppose coeficient at $\prod_{i=1}^{n} x_i^{d_i}$ is nonzero. Then for any sets $S_1, S_2, \dots, S_n$ with $|S_i| > d_i$ there exists $s_1, s_2, \dots, s_n$ with $s_i \in S_i$ such that $p(s_1, s_2, \dots, s_n) \neq 0$*

**Proof**

First we prove by induction that if $q$ is polynomial in $F[x_1, x_2, \dots, x_n]$and if $d_i$ is maximum degree of $x_i$ in $q$ and $S_i \subset F$ with $|S_i| = d_i + 1$ then if p is zero at $C = S_1 \times S_2 \dots \times S_n$ then $q = 0$. Because at C holds $\prod_{s \in S_i}(x_i - s) = 0$ we can substitute $x_i^{d_i+1}$ to p and get polynomial satisfying previous claim. But coefficient at $\prod_{i=1}^{n} x_i^{d_i}$ is nonzero contradiction. $\square$

**Theorem 6.**

*Every even cycle has circular list choosability two.*

**Proof**

Every graph with edges has obviously $\chi_{c,l}(G) \geq 2$. Let $v_1, v_2, \dots, v_{2n}$ be vertices of $C_2n$ in this order and let $v_{2n+1} = v_1$. Consider complex polynomial $p \in \mathbb{C}[x_1, x_2, \dots, x_2n]$ with convention $x_{2n+1} = x_1$.

$$p = \prod_{j=1}^{2n} \prod_{k=-(q-1)}^{q-1} (x_j - e^{2\pi k/p} x_{j+1})$$

This is analogy of adjacency polynomial. Consider $y_1, y_2, \dots, y_{2n}$ with $y_i \in \{0, 1, \dots, p-1\}$. Then $p(e^{2\pi i y_1/p}, e^{2\pi i y_2/p}, \dots, e^{2\pi i y_{2n}/p}) \neq 0$ if and only if $y_1, y_2, \dots, y_{2n}$ is proper $(p, q)$-coloring of $C_{2n}$

Degree $deg(p) = 2n(2q-1)$ Consider coeficient a at $\prod j = 1^n x_j^( 2q-1)$ It is equal $\sum k = 0^{2q-1} a_k^{2n}$ where $a_k$ is coefficient at $x_j^{2q-1-k} x_{j+1}^k$ in

$\prod_{k=-(q-1)}^{q-1}(x_j - e^{2\pi k/p}x_{j+1}$. Clearly

$$a_k = \sum J \subset \{-(q+1), \ldots, q-1\}, |J| = k \prod_{j \in J} -e^{2\pi ij/p}$$

Observe that $a_k = a_k*$ so $a_k$ are real and because $a_0 = 1$ then a is positive. If we have 2-(p,q)-coloring then size of list $L_j \geq 2p$ so there exist coloring by nullstellensatz. $\square$


Kateřina Böhmová

asi.tak@pikomat.mff.cuni.cz


Presented paper by V. Anuradha, Chinmay Jain, Jack Snoeyink, Tibor Szabó

## How long can a graph be kept planar?

(emis.kaist.ac.kr/journals/EJC/Volume_15/PDF/v15i1n14.pdf)

### Introduction

The graph (non-)planarity game is played on the complete graph $K_n$ between an Enforcer and an Avoider, each of whom take one edge per round. The game ends when the edges chosen by Avoider form a non-planar subgraph.

The graph (non-)planarity game is an instance of a graph game that Enforcer "easy" wins and we are interested in how fast he wins. Enforcer inevitably succeeds by round $3n - 5$, when Avoider's graph has simly too many edges to be planar. The paper shows that Avoider can play for $3n-26$ turns, improving the previous bound of $3n - 28\sqrt{n}$.

The paper details Avoider's strategy, which can be summarized as follows: Avoider designates four *anchor* vertices in two pairs, $a_0$, $a_1$ and $b_0$, $b_1$. In each turn, Avoider adds an edge either to connect a remaining vertex to one anchor from each pair, or to play a *permutation game* that builds path joining vertices that are connected to the same pair of anchors.

**More formally**

In the talk we prove the following theorem.

**Theorem 1.** *In the graph (non-)planarity game on n vertices, Avoider can play at least* $3n - 26$ *edges while keeping his graph planar.*

We define some terminology.

**Definition 2.** Let $V$ be the set of $n - 4$ non-anchor vertices. Avoider will connect each $v \in V$ to exactly two anchors $a_i$ and $b_i$ for some $i, j \in \{0, 1\}$ – in this case we say that $w$ is *pinned to ij.*

**Definition 3.** After $v$ is pinned to $ij$, Avoider may choose edges that join $v$ to at most two other vertices that have also been pinned to $ij$. We say that a vertex is *active in ij* if it is pinned to $ij$, but not yet connected to two other neighbours.

For the proof the following lemmas will be useful.

**Lemma 4.** *Avoider chooses at least three of the six edges between anchors, and guarantees that every non-anchor vertex* $v \in V$ *becomes pinned to some* $ij$*, with* $i, j \in \{0, 1\}$

**Lemma 5.** *Let* $P$ *be the set of all vertices pinned to* $ij$*. Avoider will be able to choose at least* $|P| - 6 - i \cdot j + 2(i + j)$ *edges in the subgraph induced by* $P$*.*

Martin Chmelík
207772@mail.muni.cz
Presented paper by Avraham Trahtman

# The Road Coloring Problem

(http://arxiv.org/pdf/0709.0099)

### Introduction

A synchronizing word of a deterministic automaton is a word in the alphabet of colors (considered as letters) of its edges that maps the automaton to a single state. A coloring of edges of a directed graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

The road coloring problem is the problem of synchronizing coloring of a directed finite strongly connected graph with constant outdegree of all its vertices if the greatest common divisor of lengths of all its cycles is one. The problem was posed by Adler, Goodwyn and Weiss over 30 years ago and evoked noticeable interest among the specialists in the theory of graphs, deterministic automata and symbolic dynamics.

### More formally

**Definition 1.** An *AGW* graph (Adler, Goodwyn and Weiss) is a finite directed strongly connected graph with constant outdegree of all its vertices where the *gcd* of lengths of all its cycles is one.

**Definition 2.** A word $s \in \Sigma^+$ is called a synchronizing word of an automaton with transition graph $\Gamma$ if $|\Gamma s| = 1$.

**Definition 3.** A coloring of a directed finite graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

**Definition 4.** A pair of distinct states $\mathbf{p}, \mathbf{q}$ of an automaton will be called *synchronizing* if $\mathbf{p}s = \mathbf{q}s$ for some $s \in \Sigma^+$. If for any $s$ $\mathbf{p}s \neq \mathbf{q}s$, we call the pair *deadlock*. A synchronizing pair of states $\mathbf{p}$, $\mathbf{q}$ of an automaton is called *stable* if for any word $u$ the pair $\mathbf{p}u, \mathbf{q}u$ is also synchronizing.

**Definition 5.** We call the set of all outgoing edges of a vertex a bunch if all these edges are incoming edges of only one vertex.

**Definition 6.** Let $u$ be a left eigenvector with positive components having no common divisor of adjacency matrix of a graph with vertices $\mathbf{p}_1, \ldots, \mathbf{p}_n$. The $i$-th component $u_i$ of the vector $u$ is called the weight of the vertex $\mathbf{p}_i$ and denoted by $\omega(\mathbf{p}_i)$. The sum of the weights of the vertices from a set $D$ is denoted by $\omega(D)$ and is called the weight of $D$. The subset $D$ of states of an automaton such that $\omega(D)$ is maximal and $|Ds| = 1$ for some word $s \in \Sigma^+$ let us call *$F$-maximal*.

The subset $\Gamma s$ of states for some word $s$ such that every pair of states from the set is deadlock will be called an *$F$-clique*.

**Theorem 7.** *[1] There exists a partition of $\Gamma$ on $F$-maximal sets (of the same weight)*

**Theorem 8.** *[2] Let us consider a coloring of $AGW$ graph $\Gamma$. Stability of the states is a binary relation on the set of states of the obtained automaton; denote this relation by $\rho$. Then $\rho$ is a congruence relation, $\Gamma/\rho$ presents an $AGW$ graph and synchronizing coloring of $\Gamma/\rho$ implies synchronizing recoloring of $\Gamma$.*

**Lemma 9.** *Let $\omega$ be the weight of $F$-maximal set of the $AGW$ graph $\Gamma$ via some coloring. Then the size of every $F$-clique of the coloring is the same and equal to $\omega(\Gamma)/\omega$ (the size of partition of $\Gamma$ on $F$-maximal sets).*

**Lemma 10.** *Let $F$ be $F$-clique via some coloring of $AGW$ graph $\Gamma$. For any word $s$ the set $Fs$ is also an $F$-clique and any state [vertex] $p$ belongs to some $F$-clique.*

**Lemma 11.** *Let $A$ and $B$ ($|A| > 1$) be distinct $F$-cliques via some coloring without stable pairs of the $AGW$ graph $\Gamma$. Then $|A| - |A \cap B| = |B| - |A \cap B| > 1$.*

**Lemma 12.** *Let some vertex of $AGW$ graph $\Gamma$ have two incoming bunches. Then any coloring of $\Gamma$ has a stable couple.*

**Definition 13.** Let us call a subgraph $S$ of the AGW graph $\Gamma$ a spanning subgraph of $\Gamma$ if to $S$ belong all vertices of $\Gamma$ and exactly one outgoing edge of every vertex.

**Lemma 14.** *Let $N$ be a set of vertices of level $n$ from some tree of the spanning subgraph $S$ of $AGW$ graph $\Gamma$. Then in a coloring of $\Gamma$ where all* *edges of $S$ have the same color $\alpha$, any $F$-clique $F$ satisfies $|F \cap N| \leq 1$.*

**Lemma 15.** *Let $AGW$ graph $\Gamma$ have a spanning subgraph $R$ of only disjoint cycles (without trees).Then $\Gamma$ also has another spanning subgraph with exactly one vertex of maximal positive level.*

**Lemma 16.** *Let any vertex of an $AGW$ graph $\Gamma$ have no two incoming bunches. Then $\Gamma$ has a spanning subgraph such that all its vertices of maximal positive level belong to one non-trivial tree.*

**Theorem 17.** *Any $AGW$ graph $\Gamma$ has a coloring with stable couples.*

**Theorem 18.** *Every $AGW$ graph $\Gamma$ has synchronizing coloring.*

### References

[1] J. Friedman: On the road coloring problem, *Proc. of the Amer. Math. Soc.110(1990), 1133-1135.*

[2] J. Kari: Synchronizing finite automata on Eulerian digraphs, *Springer, Lect. Notes in Comp. Sci., 2136(2001), 432-438*

## Július Czap

julius.czap@upjs.sk

Presented paper by J. Czap, S. Jendroľ, F. Kardoš and J. Miškuf

## Looseness of Plane Graphs

(http://umv.science.upjs.sk/preprints/dokumenty/A3-2009.pdf)

### Introduction

All considered graphs are finite, loops and multiple edges are allowed.

Let $G = (V, E, F)$ be a connected plane graph with the vertex set $V$, the edge set $E$ and the face set $F$. For a face $f$, the *size* of $f$, $deg(f)$, is defined to be the length of the shortest closed walk containing all edges from the boundary of $f$.

We write $v \in f$ if a vertex $v$ is incident with a face $f$. The vertices $u$ and $v$ are *face independent* if there is no face $f$ such that $v \in f$ and $u \in f$. A set $S$ of vertices is face independent if any two vertices from $S$ are face independent. The *face independence number* of a graph $G$, $\alpha_2(G)$, is defined to be the number of vertices in the maximum face independent set of vertices.

A $k$*-colouring* of a graph $G$ is a mapping $\varphi : V(G) \to \{1, \ldots, k\}$. For a set $X \subseteq V$ we define $\varphi(X)$ to be the set of colours $\{\varphi(v); v \in X\}$. Particularly, if $f$ is a face of $G$ then $\varphi(f)$ denotes the set of colours used on the vertices incident with the face $f$. A face $f \in F$ is called *loose* if $|\varphi(f)| \geq 3$, otherwise it is called nonloose. A $k$-colouring of a graph $G$ is called the *nonloose* $k$*-colouring* if it does not involve any loose face, otherwise it is a *loose* $k$*-colouring.*

We are interested in finding an answer to the following

**Question:** What is the minimum number of colours $ls(G)$ that any surjective vertex colouring of a connected plane graph $G$ with $ls(G)$ colours enforces a loose face?

The invariant $ls(G)$ of a plane graph $G$ is called the *looseness* of $G$ and it has been introduced by Negami and Midorikawa. Negami proved that for plane triangulation $G$

$$\alpha_0(G) + 2 \leq ls(G) \leq 2\alpha_0(G) + 1 \ ,$$

where $\alpha_0(G)$ is the vertex independence number of $G$.

## General properties

**Observation 1.** *Let $G$ be a plane graph. Then $ls(G) \geq \alpha_2(G) + 2$.*

**Theorem 2.** *Let $G$ be a connected plane graph such that the dual $G^*$ of $G$ has $t$ vertex disjoint cycles. Then*

$$ls(G) \geq t + 2 \ .$$

**Theorem 3.** *Let $G$ be a connected plane graph and let $G^*$ be its dual. Then there are $t_0$ vertex disjoint cycles in $G^*$ such that*

$$ls(G) = t_0 + 2 \ .$$

The problem of determining the maximum number of vertex disjoint cycles in plane graphs is known to be NP-complete, so good estimations for this parameter are welcome.

The girth of a graph $G$ is the length of its shortest cycle.

**Theorem 4.** *Let $G = (V, E, F)$ be a connected plane graph, let $g$ be the girth of the dual graph $G^*$ of $G$. Then*

$$ls(G) \leq \frac{1}{g}|F(G)| + 2 \ .$$

*Moreover, the bound is sharp.*

The edge connectivity of a plane graph $G$ plays an important role in the concept of the looseness of $G$. Observe, that each minimum edge-cut of size $g$ in $G$ corresponds to a cycle in $G^*$ and vice versa, therefore, the edge connectivity of a graph $G$ is equal to the girth of the dual graph $G^*$. Hence, we can obtain an upper bound for the looseness of plane graphs in terms of the edge connectivity of $G$.

**Theorem 5.** *Let $G = (V, E, F)$ be a connected plane graph with the edge connectivity $\kappa'$. Then*

$$ls(G) \leq \frac{1}{\kappa'}|F(G)| + 2 \ .$$

*Moreover, the bound is sharp.*

**Corollary 6.** *Let $G$ be an $n$-vertex 3-connected cubic plane graph. Then*

$$ls(G) \leq \frac{1}{6}n + \frac{8}{3} \ .$$

**Theorem 7.** *For any integer $t \geq 4$ there exists a 3-connected cubic plane graph $G$ on $n$ vertices such that*

$$t = ls(G) = \frac{1}{6}n + \frac{8}{3} \ .$$

**Observation 8.** *Let $G$ be a plane graph on $n$ vertices which contains a face incident with at least three vertices. Then $ls(G) \leq n$.*

If $G$ is a simple plane graph then this trivial upper bound is tight if and only if $G$ is a triangle. If multiple edges or loops are allowed, suitable graphs could be constructed easily.

**Theorem 9.** *Let $G$ be a connected simple plane graph on $n$ vertices. Then*

$$ls(G) \leq \frac{2n+2}{3} \quad .$$

**Theorem 10.** *For any integer $t \geq 1$ and any $k \in \{1,2,3\}$ there exists a simple $k$-connected plane graph $G$ on $n$ vertices, $n \geq t$, such that*

$$ls(G) = \frac{2n+2}{3} \quad .$$


VANDERBILT

Martin Derka

mderka@mail.muni.cz

# Planar Finite Graph Emulators and Fellows' Conjencture

(http://arxiv.org/abs/0812.3700v1)

## Introduction

In this short paper, we deal with finite planar emulation of undirected graphs. We define basic terms related to emulators and have a look at Fellows' conjencture. Considering recent research, we displace this 20 years old conjecture. We also demonstrate using simple prove technique called discharging to prove absence of emulator for particular graph.

## More formally

**Definition 1.** A graph $H$ is a cover of a graph $G$ if there exists a pair of onto mappings $(\varphi, \psi), \varphi : V(H) \to V(G), \psi : E(H) \to E(G)$, called a (cover) projection, such that $\psi$ maps the edges incident with each vertex $v$ in $H$ bijectively onto the edges incident with $\varphi(v)$ in $G$.

**Definition 2.** A graph $H$ is an emualtor of a graph $G$ if there exists a pair of onto mappings $(\varphi, \psi), \varphi : V(H) \to V(G), \psi : E(H) \to E(G)$, called a (cover) projection, such that $\psi$ maps the edges incident with each vertex $v$ in $H$ surjectively onto the edges incident with $\varphi(v)$ in $G$.

**Definition 3.** Graph $G$ is called minor of graph $H$ if it can be obtained by 0 or more edge contraction from a subgraph of $H$.

**Conjecture 4.** *Graph $G$ has a finite cover if and only if it embeds in the projective plane.*

**Conjecture 5.** *Graph $G$ has a finite emulator if and only if there is a finite cover of this graph.*

**Lemma 6.** *Graph $G$ embeds in the projective plane if and only if it does not contain any one of forbidden minors for the projective plane.*

**Lemma 7.** *Let's consider $H$ to be a finite emulator of $G$ and $v \in V(G)$ to be a vertex degree 3. Then $H$ can be modified so that all vertices in $\varphi^{-1}(v)$*

*are degree 3.*

### References

[1] P. Hlinn: 20 Years of Negamis Planar Cover Conjecture, *http://www.fi.muni.cz/˜hlineny/Research/papers/plcover20-gc.pdf*, 2008.

[2] Y. Rieck, Y. Yamashita: Finite planar emulators for $K_{4,5} - 4K_2$ and $K_{1,2,2,2}$ and Fellows' Conjecture, *http://arxiv.org/abs/0812.3700v1*, 2008

Codrut Grosu

grosu.codrut@gmail.com

Presented paper by Jacob Fox, Csaba D. Tóth

## On the decay of crossing numbers

(*Journal of Combinatorial Theory, Series B 98 (2008), 33-42*)

### Introduction

The crossing number $\mathrm{cr}(G)$ of a graph $G$ is the minimum possible number of crossings over all drawings of $G$ in the plane. In 1993, Richter and Thomassen conjectured that in any graph $G$ with crossing number $k$ there is an edge $e$ such that $\mathrm{cr}(G - e) \geq k - c\sqrt{k}$, for some constant $c$. However this was proved only for dense graphs, that is, graphs with $n$ vertices and $\Omega(n^2)$ edges.

In this talk I shall present a proof of the following theorem:

**Theorem 1.** *For every $\epsilon > 0$, there is a constant $n_0$ depending on $\epsilon$ such that if $G$ is a graph with $n > n_0$ vertices and $m > n^{1+\epsilon}$ edges, then there is a subgraph $G'$ of $G$ formed by deleting at least $\frac{\epsilon}{24}m$ edges, and with crossing number*

$$\mathrm{cr}(G') \geq \left(\frac{1}{28} - o(1)\right)\mathrm{cr}(G)$$

The proof of Theorem 1 uses a technique called the embedding method. This technique helps proving upper bounds for the crossing number of a graph $G$ in terms of the crossing number of one of its subgraphs.

We shall first apply the embedding method to prove the theorem in the special case when the crossing number is bounded from below by the sum of the squares of the degrees (Theorem 2).

While in general this restriction does not hold, it can be shown that any graph $G$ which is not too sparse has the crossing number bounded from below by a sum of the degree squares of a well chosen subset of vertices. These vertices induce a subgraph $G^*$ of $G$ to which the strategy of Theorem 2 may be applied, thus finding the subgraph $G'$ required by Theorem 1.

**More formally**

In the special case when $G$ is a graph with $n$ vertices, degree sequence $d_1, d_2, \ldots, d_n$ and $\mathrm{cr}(G) \geq \frac{7}{16}\sum_{i=1}^{n} d_i^2$, the following theorem holds:

**Theorem 2.** *For every $\epsilon, 0 < \epsilon < 1$, there is a positive constant $n_0(\epsilon)$ such that for every graph G with $n > n_0(\epsilon)$ vertices, degree sequence $d_1, d_2, \ldots, d_n$, and $m > n^{1+\epsilon}$ edges, there is a subgraph $G'$ of $G$ with at most $(1 - \frac{\epsilon}{8})m$ edges such that*

$$4\mathrm{cr}(G') \geq \mathrm{cr}(G) - \frac{3}{8}\sum_{i=1}^{n} d_i^2$$

Clearly in the special case above, Theorem 2 implies Theorem 1.

In order to prove Theorem 1 in general, we shall need the following:

**Lemma 3.** *Let $G$ be a graph with $n$ vertices of degrees $d_1, d_2, \ldots, d_n$ and $m$ edges. For any $\delta$, $0 < \delta < 1$, let $\Delta = \Delta(\delta)$ be the integer such that $\sum_{i=1}^{n} \min(d_i, \Delta) < 2\delta m$ but $\sum_{i=1}^{n} min(d_i, \Delta + 1) \geq 2\delta m$.*
*If $m \geq 45(1-\delta)^{-2} n log^2 n$, then*

$$\mathrm{cr}(G) \geq \frac{1}{16}\sum_{i=1}^{n}(\min(d_i, \Delta))^2$$

The proof of lemma 3 relies on a relation between the bisection width $b(G)$ and the crossing number $\mathrm{cr}(G)$ of a graph $G$. Namely, it uses the following theorem by Pach, Shahrokhi, and Szegedy:

**Theorem 4.** *Let $G$ be a graph with $n$ vertices of degrees $d_1, d_2, \ldots, d_n$. Then*

$$40\,\mathrm{cr}(G) \geq b^2(G) - \frac{5}{2}\sum_{i=1}^{n} d_i^2$$

Using lemma 3 one can show that any dense enough graph has the crossing number bounded from below by a prefix sum of the degree squares. More precisely, we can show the following lemma:

**Lemma 5.** *Let $G$ be a graph with $n$ vertices and $m$ edges, and let $d_1 \leq d_2 \leq \ldots \leq d_n$ denote the degree sequence sorted in monotone increasing order. Let $l$ be the integer such that $\sum_{i=1}^{l-1} d_i < 4m/3$ but $\sum_{i=1}^{l} d_i \geq 4m/3$. If $m = \Omega(nlog^2 n)$, then*

$$\mathrm{cr}(G) \geq \left(\frac{1}{64} - o(1)\right)\sum_{i=1}^{l} d_i^2$$

Using lemma 5 we may now prove Theorem 1 in the general case. The proof will be along the same lines as the proof of Theorem 2.


The reduction to the T-join problem


Edge-Coloring Bipartite Multigraphs in
E – cardinality of edge set (edge set)
V – vertex set (vertex set)
R. COLE
S. SCHIRRA
PRELIMINARIES
THE ALGORITHM
THEOREMS
teacher time table

Tereza Klimošová

matfyzacka@gmail.com


Presented paper by Pierre Fraigniaud, Amos Korman

# Compact Ancestry Labeling Schemes for Trees of Small Depth

(arxiv.org/PS_cache/arxiv/pdf/0902/0902.3081v1.pdf)

## Introduction

An *ancestry labeling scheme* labels the nodes of any tree in such a way that ancestry queries between any two nodes (whether the first node is an ancestor of the second) in a tree can be answered just by looking at their corresponding labels. The common measure to evaluate the quality of an ancestry labeling scheme is by its *label size*, that is the maximal number of bits stored in a label, taken over all $n$-node trees. Known bounds on the label size are $\log n + \mathcal{O}(\sqrt{\log n})$ for upper bound and $\log n + \mathcal{O}(\log \log n)$ for lower bound.

The design of ancestry labeling schemes finds applications in XML search engines. Article presents an ancestry labeling scheme of size $\log n + 2\log d + \mathcal{O}(1)$, for the family of trees with at most $n$ nodes and depth at most $d$. It is motivated by the observation that the depth of a typical XML tree is bounded from above by a small costant.

## More formally

**Definition 1.** A *rooted forest* $F$ is a collection of rooted trees. The depth of $F$ is the maximum depth of tree in $F$. For integers $n$ and $d$, let $\mathcal{F}(n, d)$ denote the family of all rooted forests with at most $n$ nodes and depth bounded from above by $d$.

**Definition 2.** An *ancestry labeling scheme* $(\mathcal{M}, \mathcal{D})$ for a family of rooted forest $\mathcal{F}$ is composed of

- a *marker* algorithm $\mathcal{M}$ that assigns labels to nodes of a given forest $F \in \mathcal{F}$

- a polynomial time *decoder* algorithm $\mathcal{M}$ that given two labels $l_1$ and $l_2$ in the output domain of $\mathcal{M}$, returns a boolean in $\{0, 1\}$.

These algorithms has to satisfy that if $L(u)$ and $L(v)$ denote labels assigned by marker algorithm to two nodes $u$ and $v$ in some rooted forest $F \in \mathcal{F}$, then

$$\mathcal{D}(L(u), L(v)) = 1 \Leftrightarrow u \text{ is an ancestor of } v \text{ in } F.$$

Note that the decoder $\mathcal{D}$ is independent of the forest $F$.

**Theorem 3.** *There exists an ancestry labeling scheme for the family of rooted forests in $\mathcal{F}$ whose label size is* $\log n + 2 \log d + \mathcal{O}(1)$.

Informally the scheme performs as follows. We construct a set of intervals $U$ such that the nodes of any forest in $\mathcal{F}$ can be mapped to $U$, in a way that ancestry relation can be answered using a simple interval containment test, i.e. we make sure that $u$ is an ancestor of $v$ in some forest $F$ if and only if the interval associated with $u$ contains the interval associated with $v$. A label of a node in $F$ is a pointer to an element in $U$ and thus can be encoded using $\log |U|$ bits. The construction of $U$ is done by induction on the number of nodes in the forest.


Ľuboš Korenčiak

208317@mail.muni.cz


Presented paper by Richard Cole, Kirstin Ost, Stefan Schirra

## Edge-Coloring Bipartite Multigraphs in $O(E \log D)$ Time

(http://www.springerlink.com/content/xj94c5n8406tykac/)

The aim of the paper is to show that a minimal edge-coloring of bipartite multigraph can be computed in $O(E \log D)$ time. The result is based on an algorithm for finding a matching in a regular bipartite graph in $O(E)$ time.

The technique presented in the paper uses a divide and conquer approach: They partition regular graph $G$ into subgraphs that are recursively colored with different color sets using Euler splits and finding a matching.

An edge-coloring of a multigraph $G = (V, E)$ with a vertex set $V$ and edge set $E$ is a map $c : E \longrightarrow N$ such such that no two adjacent edges have the same color.

The number of colors used in a minimal edge-coloring for the class of bipartite multigraphs is the maximum degree.

A *matching* in $G$ is a subset $M$ of $E$ with the property that no two edges have a common endpoint. $M$ is said do be $full$ if it contains, as an endpoint, every vertex of $G$.

*Euler partition* of a graph $G$ is a partition of its edges into open and closed paths, such that each vertex of odd degree is at the end of exatly one open path, and each vertex of even degree is at the end of no open path.

An *Euler split* of bipartite graph $G$ splits $G$ into two bipartite graphs $G_1 = (V, E_1)$ and $G_2(V, E_2)$ where $E_1$ and $E_2$ are formed by scanning the paths of an Euler partition of $G$ and alternatively placing one edge into $E_1$ and one edge into $E_2$.

$Chains$ are edges formed into paths of length at most $s$ edges. Chains have implicit direction and are vertex disjoint.

**Theorem 1.** *The edge-coloring algorithm runs in time $O(E \log D)$.*

Jan Krčál

xkrcal4@fi.muni.cz

Presented paper by Tomáš Brázdil, Václav Brožek, Antonín Kučera, Jan Obdržálek

# The Determinacy of Stochastic Games with Reachability Objectives

(http://www.fi.muni.cz/usr/kucera/papers/stacs09.pdf)

## Introduction

A stochastic game is basically moving a token in a graph. In a 2 1/2-player game, there are 3 players involved: player □ , player ◊ and the random player ◯ (the 1/2 player). The vertices in the graph are divided among the players, each player decides the next move if the token gets into his vertex. Players □ and ◊ decide according to their strategies, player ◯ decides randomly.

Every game needs a goal. For a game with a reachability objective, the goal of the player □ is to move the token to any vertex in a specified target set, the goal of the player ◊ is to avoid it.

What is the highest probability of reaching the target, which the player □ can achieve using a fixed strategy against arbitrary strategy of the adversary? And vice versa, what is the lowest probability of reaching the target, which the player ◊ can force using a fixed strategy against arbitrary strategy of player □ ? If the answer is the same for both questions, then the game is determined, it has a value. The player □ cannot guarantee a higher chance of hitting the target whereas the player ◊ cannot guarantee a lower chance.

Another question is, whether there are optimal strategies for both players for a given game, whether a player gets the value of the game by one fixed strategy or whether there is only an infinite sequence of strategies approaching the value.

Stochastic games with reachability objectives are determined, we will give simple proofs of the determinacy and discuss some subtle details of the problem.

## Definitions

**Definition 1.** A *stochastic game* is a tuple $G = (V, \mapsto, (V_\Box, V_\Diamond, V_\bigcirc), Prob)$ where $V$ is a finite or countably infinite set of vertices, $\mapsto \subseteq V \times V$ is a total edge relation, $(V_\Box, V_\Diamond, V_\bigcirc)$ is a partition of $V$, and $Prob$ is a probability assignment which to each $v \in V$ assigns a positive probability distribution on the set of its outgoing transitions. We say that $G$ is finitely branching if for each $v \in V$ there are only finitely many $u \in V$ such that $v \mapsto u$.

**Definition 2.** Let $\odot \in \{\Box, \Diamond\}$. A *strategy* for player $\odot$ is a function which to each history $wv \in V^* V_\odot$ assigns a probability distribution on the set of outgoing edges of $v$.

The set of all strategies for player $\Box$ and player $\Diamond$ is denoted $\Sigma$ and $\Pi$, respectively.

We say that a strategy $\tau$ is *memoryless* (M) if $\tau(wv)$ depends just on the last vertex v, and *deterministic* (D) if $\tau(wv)$ is a Dirac distribution for all $wv$ (i.e. assigns probability 1 to one outgoing vertex and 0 to all other outgoing vertices). Strategies that are not necessarily memoryless are called *history-dependent* (H), and strategies that are not necessarily deterministic are called *randomized* (R).

**Definition 3.** A $\sigma$-field over a set $X$ is a set $\mathcal{F} \subseteq 2^X$ that includes $X$ and is closed under complement and countable union.

A *probability space* is a triple $(X, \mathcal{F}, \mathcal{P})$ where $X$ is a set called sample space, $\mathcal{F}$ is a $\sigma - field$ over $X$, so that $(X, \mathcal{F})$ is a measurable space and $\mathcal{P}$ is a probability measure over $(X, \mathcal{F})$, i.e. a function $\mathcal{P} : \mathcal{F} \to \mathbb{R}^{\geq 0}$ such that, for each countable collection $\{X_i\}_{i \in I}$ of pairwise disjoint elements of $\mathcal{F}$, $\mathcal{P}(\bigcup_{i \in I} X_i) = \sum_{i \in I} \mathcal{P}(X_i)$, and moreover $\mathcal{P}(X) = 1$.

**Definition 4.** A *Markov chain* is a triple $\mathcal{M} = (M, \longrightarrow, Prob)$ where $M$ is a finite or countably infinite set of states, $\longrightarrow \subseteq M \times M$ is a total transition relation, and $Prob$ is a function which to each $s \in M$ assigns a positive probability distribution over the set of its outgoing transitions.

Each finite path $w$ in $\mathcal{M}$ determines a *basic cylinder* $Run(\mathcal{M}, w)$ which consists of all runs that start with $w$.

To every $s \in M$ we associate the probability space $(Run(\mathcal{M}, s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the $\sigma$-field generated by all basic cylinders $Run(\mathcal{M}, w)$ where $w$ starts with $s$, and $\mathcal{P} : \mathcal{F} \to \mathbb{R}^{\geq 0}$ is the unique probability measure such that $P(Run(\mathcal{M}, w)) = \prod_{i=0}^{m1} x_i$ where $w = s_0, \ldots, s_m$ and $s_i \xrightarrow{x_i} s_{i+1}$ for every

$0 \leq i < m$ (if $m = 0$, we put $\mathcal{P}(Run(\mathcal{M}, w)) = 1$).

**Definition 5.** Each pair of strategies $(\sigma, \pi) \in \Sigma$ determines a unique play of the game $G$, which is a Markov chain $G(\sigma, \pi)$ where $V^+$ is the set of states, and $wu \xrightarrow{x} wuu$ iff $u \mapsto u'$ and one of the following conditions holds:

- $u \in V_\Box$ and $\sigma(wu)$ assigns $x$ to $u \mapsto u'$, where $x > 0$;
- $u \in V_\Diamond$ and $\pi(wu)$ assigns $x$ to $u \mapsto u'$, where $x > 0$;
- $u \in V_\bigcirc$ and $Prob(u)$ assigns $x$ to $u \mapsto u'$, where $x > 0$

Let $T \subseteq V$ be a set of target vertices. By $P_v^{\sigma,\pi}(Reach(T))$ we denote the probability of all runs $w \in Run(G(\sigma, \pi), v)$ such that $w$ visits some $u \in T$. By $P_v^{\sigma,\pi}(Reach_n(T))$ we denote the probability of all runs that visit some $u \in T$ in first $n$ steps.

**Definition 6.** A reachability objective is a pair $(T, \trianglerighteq \rho)$ where $T \subseteq V$, $\trianglerighteq \in \{>, \geq\}$, and $\rho \in [0, 1]$. We define $\triangleleft \in \{<, \leq\}$ as $\triangleleft\ =\ \ntrianglerighteq$. We say that:

- $\sigma \in \Sigma$ is $(\trianglerighteq \rho) - winning$ if $P_v^{\sigma,\pi}(Reach(T)) \trianglerighteq \rho$ for all $\pi \in \Pi$;
- $\pi \in \Pi$ is $(\triangleleft \rho) - winning$ if $P_v^{\sigma,\pi}(Reach(T)) \triangleleft \rho$ for all $\sigma \in \Sigma$.

**Definition 7.** For a reachability game $G$:

- *lower value* of the game in vertex $v$ is
  $h_v^- = \sup_{\sigma\in\Sigma} \inf_{\pi\in\Pi} P_v^{\sigma,\pi}(Reach(T))$,
- *upper value* of the game in vertex $v$ is
  $h_v^+ = \inf_{\pi\in\Pi} \sup_{\sigma\in\Sigma} P_v^{\sigma,\pi}(Reach(T))$.
- We say that vertex $v$ has a *value* $val(v)$ iff $h_v^- = h_v^+$, i.e.:
  $val(v) = \sup_{\sigma\in\Sigma} \inf_{\pi\in\Pi} P_v^{\sigma,\pi}(Reach(T)) =$
  $\inf_{\pi\in\Pi} \sup_{\sigma\in\Sigma} P_v^{\sigma,\pi}(Reach(T))$.
- We say that $G$ is *determined* if all vertices have a value.

## Claims

**Theorem 8.** *Let $G$ be an finite game and let $(T, \trianglerighteq \rho)$ be a reachability objective. Every $v \in V$ has a value and both player have optimal MD (memoryless deterministic) strategy.*

**Theorem 9.** *Let $G$ be a finite branching game and let $(T, \trianglerighteq \rho)$ be a reachability objective. Every $v \in V$ has a value and player $\Diamond$ has an optimal strategy.*

**Lemma 10.** *For every fixed vertex $v \in V$, we have that*

$$\forall \epsilon > 0 \; \exists \sigma \in \Sigma \; \exists n \in \mathbb{N} \; \forall \pi \in \Pi : P_v^{\sigma,\pi}(Reach_n(T)) > val(v) - \epsilon$$

**Theorem 11.** *Let $G$ be a finite branching game and let $(T, \trianglerighteq \rho)$ be a reachability objective. For each $v \in V$, either the player $\Box$ has a $(\trianglerighteq \rho)$-winning strategy in $v$ or the player $\Diamond$ has a $(\lhd \rho)$-winning strategy in $v$.*

**Observation 12.** *In infinite branching games with reachability objective, player $\Diamond$ does not have optimal strategies in general.*


Martin Křivánek

set@mail.muni.cz

Presented paper by Omid Amini, Frédéric Mazoit, Nicolas Nisse, Stéphan Thomassé


## Submodular Partition Functions

(www-sop.inria.fr/members/Nicolas.Nisse/publications/bramble.pdf)

### Introduction

Article shows a new proof of the duality between the bramble-number of a graph and its tree-width. Their approach is based on a new definition of submodularity on partition function. The proof does not rely on Menger's theorem, and thus greatly generalises the original one. One can also derive all known dual notions of other classical width-parameters from it.

### More formally

Let $E$ be a finite set with at least two elements and $\mathcal{F}$ be a set of admissible partitions of $E$.

**Definition 1.** A *partitioning-tree* on $E$ is a tree $T$ in which the leaves are identified with elements of $E$ in a one-to-one way. $T$ defines a set of partitions of $E$, every edge $e \in E(T)$ defines a bipartition $T_e$ of $E$, every vertex $v \in V(T)$ defines a partition $T_v$ of $E$. Tree $T$ is an *admissible partitioning-tree* for $\mathcal{F}$ if $\{T partitions\} \subseteq \mathcal{F}$.

**Definition 2.** A *$\mathcal{F}$-bramble* is a set $\mathcal{B}$ of subsets of $E$ such that for any $X_i, X_j \in \mathcal{B}, X_i \cap X_j \neq \emptyset$ and for any $\{E_1, ..., E_k\} \in \mathcal{F}$, there is $E_i \in \mathcal{B}$. $\mathcal{B}$ is *principal* if $\bigcap_{X_i \in \mathcal{B}} \neq \emptyset$.

**Lemma 3.** *If there is a non-principal $\mathcal{F}$-bramble, then there is no admissible partitioning-tree for $\mathcal{F}$.*

The question is how to characterize the families $\mathcal{F}$ of partitions of $E$ such that it is an equivalence.

**Definition 4.** A *partition function* is a function $\Phi : \{\text{partitions of } E\} \to \mathbb{N}$. Let $\Phi$ be a partition function and let $k \geq 1$. Let $\mathcal{F}_{\Phi,k}$ be the family of the partitions $P$, with $\Phi(P) \leq k$.

**Definition 5.** A *$k$-partitioning-tree for* $\Phi$ is an admissible partitioning-tree for $\mathcal{F}_{\Phi,k}$. A *$k$-bramble for* $\Phi$ is a $\mathcal{F}_{\Phi,k}$-bramble.

Let $Y$ be a subset of $E$. The partition

$$\mathcal{X}_{X_i \to Y^c} := \{X_1 \cap Y, ..., X_{i-1} \cap Y, X_i \cup Y^c, X_{i+1} \cap Y, ..., X_n \cap Y\}$$

is the partition obtained from $\mathcal{X}$ by *pushing $X_i$ to $Y^c$*.

**Definition 6.** A partition function $\Phi$ is *submodular* if, for any partitions $\mathcal{X} = \{X_1, X_2, ..., X_n\}, \mathcal{Y} = \{Y_1, Y_2, ..., Y_m\}$ and for every $1 \leq i \leq n$ and $1 \leq j \leq m$:

$$\Phi(\mathcal{X}) + \Phi(\mathcal{Y}) \geq \Phi(\mathcal{X}_{X_i \to Y_j}) + \Phi(\mathcal{Y}_{Y_j \to X_i})$$

**Definition 7.** A partition function $\Phi$ is *weakly submodular* if, for any partitions $\mathcal{X} = \{X_1, X_2, ..., X_n\}, \mathcal{Y} = \{Y_1, Y_2, ..., Y_m\}$ and for every $1 \leq i \leq n$ and $1 \leq j \leq m$:

- either there exists $F$ such that $X_i \subseteq F \subseteq (Y_j \setminus X_i)^c$ and $\Phi(\mathcal{X}) > \Phi(\mathcal{X}_{X_i \to F})$
- or $\Phi(\mathcal{Y}) \geq \Phi(\mathcal{Y}_{Y_j \to X_i^c})$.

**Theorem 8.** *Let $\Phi$ be a weakly submodular partition function on a set $E$, and let $k \geq 1$.*

- *either there is a non-principal $k$-bramble for $\Phi$,*
- *or there is a $k$-partitioning-tree for $\Phi$.*

Now we can use this theorem to prove many dual notions, for example:

**Definition 9.** The *border* function $\delta$ is defined by: $\delta(\mathcal{X})$ is the set of vertices incident to an edge in $X_i$ and in $X_j$.

**Lemma 10.** *$|\delta|$ is a submodular partition function.*

If $T$ is a $k$-partitioning-tree for $|\delta|$, than $(T, (\delta(T_t))_{t \in V(T)})$ is a tree-decomposition of width at most $k-1$. We can compute a bramble (in usual sense) of order at least $k$ from any non-principal $k$-bramble for $|\delta|$.


Robert Lukot'ka

lukotka@dcs.fmph.uniba.sk


Presented paper by Alexandr Kostochka, Dhruv Mubayi

## When is an almost monochromatic $K_4$ guaranteed

(http://www.math.uic.edu/ mubayi/papers/K4final.pdf)

The Ramsey number $R_k(p)$ denotes the minimum $n$ such that every $k$-coloring of $K_n$ results in a monochromatic $K_p$. We will examine a generalization of this parameter. A $(p,q)$-coloring of $K_n$ is an edge-coloring such that every copy of $K_p$ is colored with at least $q$ distinct colors. The minimum $n$ such that every $(p,q)$-coloring of $K_n$ with $k$ colors does not exist will be denoted as $R_k(p,q)$.

Moreover, we will examine the related parameter $f(n,p,q)$, which denotes the minimum number of colors in a $(p,q)$-coloring of $K_n$.

It is well-known that $R_k(4) < k^{O(k)}$. This implies that for some constant $c$

$$f(n,4,2) > \frac{c \log n}{\log \log n}.$$

These facts also give the bounds for $R_k(4,3)$ and $f(n,4,3)$. We will improve these bounds through examining the special case $R_k(2a,a+1)$.

**Definition 1.** A $k$-edge-colouring $\xi$ of $K_n$ is a $(\gamma_1,\ldots,\gamma_k)$-coloring if for each $i$, the color $i$ does not appear in any subgraph $K_{2\gamma_i+2}$ whose edges are colored with at most $\gamma_i+1$ colors.

We show that if $\xi$ is a $(\gamma_1,\ldots,\gamma_k)$-coloring of $K_n$ and $\gamma = \sum_i \gamma_i$ then $n < c(\log\gamma)^{1000\gamma}$. We will show this with induction on $\gamma$ and $k$.

Since a $(2a,a+1)$-coloring si an $(a-1,a-1,\ldots,a-1)$-coloring we get:

**Theorem 2.** *Let $c'$ be a positive constant depending only on $a$. Then $R_k(2a,a+1) \leq c'(\log k)^{c'k}$.*

This gives a better bound for $f(n,2a,a+1)$:

**Theorem 3.** *Let $a \geq 1$ be a fixed integer. There is a constant depending only on $a$ such that for all $n \geq 2a$*

$$f(n,2a,a+1) > \frac{c\log n}{\log\log\log n}.$$

Ján Mazák

mazak@dcs.fmph.uniba.sk

Presented paper by Serguei Norine, Tsai-Lien Wong, and Xuding Zhu

# Circular Choosability Via Combinatorial Nullstellensatz

(http://www.math.princeton.edu/~snorin/papers/circularnull.pdf)

This paper studies circular choosability of graphs, which is a combination of two variations of graph colourings: circular colouring and list colouring.

A $p$-list assignment $L$ of a graph $G$ assigns to each vertex $v$ of $G$ a set $L(v) \subseteq \{0, 1, \ldots, p-1\}$ of permissible colours. We say that $G$ is $L$-$(p,q)$-colourable if $G$ has a $(p,q)$-colouring $h$ such that $h(v) \in L(v)$ for each vertex $v$ of $G$. The circular list chromatic number $\chi_{c,l}(G)$ of a graph $G$ is the infimum of those real numbers $t$ for which the following holds: for any $p$, $q$ and for any $p$-list assignment $L$ with $|L(v)| \geq tq$ for each vertex $v \in G$, the graph $G$ is $L$-$(p,q)$-colourable.

Let $D$ be an orientation of a graph $G$. A subgraph $D'$ of $D$ is *eulerian* if $d^+_{D'}(v) = d^-_{D'}(v)$ for each vertex $v$ of $G$. An eulerian subgraph is called odd (even) if it has odd (even) number of edges. Let $EE(D)$ and $EO(D)$, respectively, denote the number of even and odd subgraphs of $D$.

We generalize a result of Alon and Tarsi and sketch a proof of the following theorem.

**Theorem 1.** *Let a graph $G$ have an orientation $D$ which has no odd directed cycles. Let $L$ be a $p$-list assignment of $G$ such that for each vertex $v$ of $G$, $|L(v)| = d^+_D(v)(2q-1)+1$. Then $G$ is $L$-$(p,q)$-colourable.*

The original result of Alon and Tarsi was proved for ordinary list colourings under the assumption that $G$ has an orientation $D$ such that $EE(D) \neq EO(D)$ instead of having an orientation $D$ without directed cycles. In the proof of Theorem 1 we first generalize the concept of odd and even eulerian subgraphs and then prove a straightforward generalization of the result of Alon and Tarsi. In this proof, we use the Combinatorial Nullstellensatz stated below.

**Theorem 2.** *Let $F$ be a field and let $f(x_1, x_2, \ldots, x_n)$ be a polynomial from the ring $F[x_1, x_2, \ldots, x_n]$. Assume that the degree of $f$ is equal to $\sum_{j=1}^{n} t_j$ and the coefficient of $\prod_{j=1}^{n} x_j^{t_j}$ in $f$ is nonzero. Then for any subsetsets $S_1, S_2, \ldots, S_n$ of $F$ satisfying $|S_j| > t_j$ there exist elements $s_1 \in S_1, s_2 \in S_2, \ldots, s_n \in S_n$ such that*

$$f(s_1, s_2, \ldots, s_n) \neq 0.$$

We illustrate usefulness of this theorem in an easy proof of the following non-trivial result: Every 4-regular graph with one added edge contains a cubic subgraph (not necessarily spanning).

Theorem 1 implies that if $G$ is a bipartite graph, then $\chi_{c,l}(G) \leq 2\lceil \mathrm{mad}(G)/2 \rceil$, where $\mathrm{mad}(G)$ is the maximum average degree of a subgraph of $G$. Using very similar techniques this result can be extended to the following theorem.

**Theorem 3.** *Let $G$ be a connected bipartite graph which is not a tree. Then $\chi_{c,l}(G) \leq \mathrm{mad}(G)$.*


Ondrej Moriš

xmoris@fi.muni.cz


Presented paper by F. V. Fomin, P. A. Golovach, D. Lokshtanov, S. Saurabh

## Clique-width: On the Price of Generality

(http://www.ii.uib.no/~daniello/papers/cliquewidth.pdf)

### Introduction

By the celebrated result of Courcelle, every decision problem expressible in monadic second order logic ($MS_2$-logic) is fixed parameter tractable when parameterized by the tree-width of the input graph. In particular, basic problems like Graph Coloring, Edge Dominating Set and Hamiltonian Cycle are solvable in linear time on graphs of bounded tree-width.

The analogue of Courcelle's result holds for graphs of bounded clique-width when the logical formulas do not use edge set quantifications ($MS_1$-logic). Despite of its generality, there are many problems expressible in $MS_2$-logic that cannot be expressed in $MS_1$-logic. The most natural are perhaps EDGE DOMINATING SET, GRAPH COLORING, and HAMILTONIAN CYCLE. There are various algorithms solving these problems on an $n$-vertex graphs of clique-width at most $k$ in time $\mathcal{O}(n^{f(k)})$, where $f$ is some function of $k$.

We show that the problems mentioned above are $\mathsf{W}[1]$-hard parameterized by clique-width – the first results distinguishing between tree-width and clique-width parameterizations.

### Definitions

A problem is called *fixed parameter tractable* (FPT) if it can be solved in time $f(k) \cdot n^c$, where $f$ is a computable function depending only on $k$ and $c$ is some constant. The basic complexity class for fixed parameter intractability is $\mathsf{W}[1]$. To show a problem is $\mathsf{W}[1]$-hard, one needs to exhibit a parameterized reduction from a known $\mathsf{W}[1]$-hard problem.

We consider finite undirected graphs without loops or multiple edges. Let $G$ be a graph, and $k$ be a positive number. A *$k$-graph* is a graph whose vertices are labeled by integers from $\{1, 2, \ldots, k\}$. The $k$-graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \ldots, k\}$ is called an *initial $k$-graph.*

**Definition 1.** The *clique-width*, $\mathbf{cwd(G)}$, of graph $G$ is the smallest integer $k$ such that $G$ can be constructed by means of repeated application of the following four operations: (i) *introduce* – construction of an initial $k$-graph labeled by $i$, denoted $i(v)$, (ii) *disjoint union* – denoted by $\oplus$, (iii) *relabel* – changing all labels $i$ to $j$, denoted by $\rho_{i\mapsto j}$, (iv) *join* – connecting all vertices labeled by $i$ with all vertices labeled with $j$ by edges, denoted by $\eta_{i,j}$.

It is a well-known that the bounded tree-width implies bounded clique-width.

**Theorem 2.** *[Corneil & Rotics, 2005] If graph $G$ has tree-width at most $t$ then $\boldsymbol{cwd}(G)$ is at most $k = 3 \cdot 2^{t-1}$.*

### Graph Coloring

In this section, we rigorously prove that GRAPH COLORING is $\mathsf{W}[1]$-hard parameterized by clique width.

Graph Coloring: The chromatic number of a graph $G$ is the smallest number of colors $\chi(G)$ needed to color the vertices of $G$ so that no two adjacent vertices share the same color.

Our reduction is from the exact version of the Equitable Coloring problem parameterized by the number $r$ of colors used, and the tree-width of the input graph. In this problem one is given a graph $G$ and integer $r$ and asked whether $G$ can be properly $r$-colored (with exactly $r$ colors) in such a way that the number of vertices in any two color classes differs by at most 1.

**Theorem 3.** Equitable Coloring *is W[1]-hard parameterized by the tree-width $t$ of the input graph and the number of colors $r$.*

On input $(G, r)$ to Equitable Coloring, we construct an instance $(G', r')$ to Graph Coloring, where $r' = r + nr$, such that the following lemmas hold.

**Lemma 4.** *If $G$ has an equitable $r$-coloring $\psi$ then $G'$ has an $r'$-coloring $\phi$.*

**Lemma 5.** *If $G'$ has an $r'$-coloring $\phi$ then $G$ has an equitable $r$-coloring $\psi$.*

**Lemma 6.** *If the tree-width of $G$ is $t$, then the clique-width of $G'$ is at most $k = 3 \cdot 2^{t-1} + 7r + 3$. Furthermore, an expression tree of width $k$ for $G'$ can be computed in FPT time.*

Lemmas 1, 2 and 3 together imply the following theorem.

**Theorem 7.** *The* Graph Coloring *problem is W[1]-hard when parameterized by clique-width. Moreover, this problem remains W[1]-hard even if the expression tree is given.*

### Edge Dominating Set and Hamiltonian Cycle Problem

In this section, we state (without proof) that Edge Dominating Set and Hamiltonian Cycle problems are also W[1]-hard parameterized by clique-width.

Edge Dominating Set: Given a graph $G$, find a minimum set of edges $X \subseteq E(G)$ such that every edge of $G$ is either included in $X$ or it is adjacent to at least one edge of $X$. The set $X$ is called an *edge dominating set* of $G$.

HAMILTONIAN CYCLE: Given a graph $G$, check whether there exists a cycle passing through every vertex of $G$.

**Theorem 8.** *The* EDGE DOMINATING SET *problem and the* HAMILTONIAN CYCLE *problem are W[1]-hard when parameterized by clique-width. Moreover, these problems remain W[1]-hard even if the expression tree is given.*

# List of All Participants

**Libor Barto**
Charles University, Prague
Czech Republic
`libor.barto@gmail.com`

**Dana Bartošová**
Charles University, Prague
Czech Republic
`dadik@email.cz`

**Ondřej Bílka**
Charles University, Prague
Czech Republic
`neleai@seznam.cz`

**Martin Böhm**
Charles University, Prague
Czech Republic
`bohm@atrey.karlin.mff.cuni.cz`

**Kateřina Böhmová**
Charles University, Prague
Czech Republic
`asi.tak@pikomat.mff.cuni.cz`

**Václav Brožek**
Masaryk University, Brno
Czech Republic
`xbrozek@fi.muni.cz`

**Jakub Bulín**
Charles University, Prague
Czech Republic
`jakub.bulin@gmail.com`

**Július Czap**
Pavol Jozef Šafárik University, Košice
Slovakia
`julius.czap@upjs.sk`

**Martin Derka**
Masaryk University, Brno
Czech Republic
mderka@mail.muni.cz

**Tomáš Gavenčiak**
Charles University, Prague
Czech Republic
gavento@kam.mff.cuni.cz

**Codrut Grosu**
Politehnica University of Bucharest
Romania
grosu.codrut@gmail.com

**Petr Hliněný**
Masaryk University, Brno
Czech Republic
hlineny@fi.muni.cz

**Martin Chmelík**
Masaryk University, Brno
Czech Republic
207772@mail.muni.cz

**František Kardoš**
Pavol Jozef Šafárik University, Košice
Slovakia
frantisek.kardos@upjs.sk

**Alexandr Kazda**
Charles University, Prague
Czech Republic
alexandr.kazda@seznam.cz

**Pavel Klavík**
Charles University, Prague
Czech Republic
pavel@klavik.cz

**Martin Klazar**
Charles University, Prague
Czech Republic
klazar@kam.mff.cuni.cz

**Tereza Klimošová**
Charles University, Prague
Czech Republic
matfyzacka@gmail.com

**Ľuboš Korenčiak**
Masaryk University, Brno
Czech Republic
lkorenciak@gmail.com

**Dan Král'**
Charles University, Prague
Czech Republic
kral@kam.mff.cuni.cz

**Jan Kratochvíl**
Charles University, Prague
Czech Republic
honza@kam.mff.cuni.cz

**Jan Krčál**
Masaryk University, Brno
Czech Republic
xkrcal4@fi.muni.cz

**Marek Krčál**
Charles University, Prague
Czech Republic
krcal@kam.mff.cuni.cz

**Martin Kupec**
Charles University, Prague
Czech Republic
magon@jkopava.cz

**Martin Křivánek**
Masaryk University, Brno
Czech Republic
set@mail.muni.cz

**Bernard Lidický**
Charles University, Prague
Czech Republic
bernard@kam.mff.cuni.cz

**Robert Lukot'ka**
Comenius University, Bratislava
Slovakia
lukotka@dcs.fmph.uniba.sk

**Jiří Matoušek**
Charles University, Prague
Czech Republic
matousek@kam.mff.cuni.cz

**Ján Mazák**
Comenius University, Bratislava
Slovakia
mazak@dcs.fmph.uniba.sk

**Ondřej Moriš**
Masaryk University, Brno
Czech Republic
xmoris@fi.muni.cz

**Pavel Paták**
Charles University, Prague
Czech Republic
ppatak@seznam.cz

**Zuzka Safernová**
Charles University, Prague
Czech Republic
zuzka@saferna.net

**Petr Škovroň**
Charles University, Prague
Czech Republic
xofon@pikomat.mff.cuni.cz

**Marek Sterzik**
Charles University, Prague
Czech Republic
marek@milimetr.org

**Martin Tancer**
Charles University, Prague
Czech Republic
tancer@kam.mff.cuni.cz

**Tomáš Valla**
Charles University, Prague
Czech Republic
valla@kam.mff.cuni.cz

**Jan Volec**
Charles University, Prague
Czech Republic
janvolec@jikos.cz