Institut für Informationssysteme Universität Hannover

Diplomarbeit

Enumeration Algorithms for Constraint Satisfaction Problems

Ilka Johannsen

15. Februar 2005

Prüfer:
Prof. Dr. Heribert Vollmer
PD. Dr. Matthias Kriesell

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbst verfasst und dabei keine außer den angegebenen Quellen und Hilfsmitteln verwendet habe.

Ilka Johannsen, Hannover den 15. Februar 2005

Contents

1 Introduction

A constraint satisfaction problem (CSP) is the question, if there is an assignment of values over some domain to a set of variables that fulfills specified constraints. In many areas of computer science and mathematics there are problems that can be expressed as CSPs, for example graph colorability and problems in database theory.

In general a CSP is NP-complete, but if we restrict the form of the constraints in a certain way we get CSPs that are tractable, that is solvable in polynomial time. In 1978 Schaefer classified the complexity of all Boolean CSPs, that means CSPs over a 2-element domain ([Sch78]). He identified tractable CSPs and proved that all others are NP-complete. Later it was shown that the complexity of CSPs is determined by certain algebraic closure properties which are represented by *polymorphisms*. So it was possible to reprove Schaefer's classification by using only polymorphisms.

Bulatov proved in [Bul02c, Bul02a] a dichotomy theorem for CSPs with a 3-element domain. Such CSPs are (like Boolean CSPs) NP-complete or tractable. Bulatov's result gives a classification of the 3-element case which had remained open since 1978. In [BJK00] the conjecture was stated that a dichotomy holds for arbitrary CSPs over finite domains. In Section 3 we describe Bulatov's classification and in Section 3.3 we restate the classification using only polymorphisms.

Another problem related to CSPs often examined is finding all assignments of values over some domain to a set of variables that fulfill specified constraints. Algorithms that enumerate all such assignments are called enumeration algorithms and the question is which sets of constraints allow an *efficient* enumeration algorithm. In [CH97] Creignou and Hebrard solved this question for the Boolean case by giving an efficient enumeration algorithm for some sets of constraints and proving all others have none unless P=NP. We generalize this algorithm in Section 4 for the 3-element domain.

2 Preliminaries

An n-ary relation on a set A is a subset of A^n. A set of finitary relations on A is said to be a constraint language on A.

Definition *Let* Γ *be a constraint language on* A*. The* constraint satisfaction problem *over* Γ*, denoted* $\mathrm{CSP}(\Gamma)$*, is defined to be the decision problem with instances* $(V, A, \mathcal{C})$*, where*

V *is a set of* variables*;*

A *is called* domain *and its elements* values*; and*

$\mathcal{C} = \{C_1, \ldots, C_q\}$ *is a set of* constraints*, in which a constraint* $C_i \in \mathcal{C}$ *is a pair* $\langle s_i, R_i \rangle$ *with* $s_i \in V^{m_i}$*, called the* constraint scope*, and* R_i *an* m_i*-ary relation from* Γ*, called the* constraint relation*.*

A solution *to* $(V, A, \mathcal{C})$ *is a function* $\varphi : V \to A$*, such that, for each constraint* $\langle s, R \rangle \in \mathcal{C}$ *it holds that* $\varphi(s) \in R$*. The question is whether there exists a solution.*

Definition *A constraint language* Γ *is said to be* tractable*, if* $\mathrm{CSP}(\Gamma')$ *is tractable, i.e. there exists an algorithm that decides in time bounded by a polynomial in the input size if the input is an instance from* $\mathrm{CSP}(\Gamma')$ *or not, for each finite subset* $\Gamma' \subseteq \Gamma$*. It is said to be* NP-complete*, if* $\mathrm{CSP}(\Gamma')$ *is NP-complete for some finite subset* $\Gamma' \subseteq \Gamma$*.*

We want to know which constraint languages are tractable, which are NP-complete and which lie in between. In [FV98] it was conjectured that there are no constraint languages on finite domains with intermediate complexity.

2.1 Algebras

To characterize tractable constraint language on a finite domains we use algebraic closure properties of relations.

Definition *Let* f *be an* n*-ary operation and* R *an* m*-ary relation. We say* f *is a* polymorphism *of* R *(or* f preserves R*,or* R *is* invariant *under* f*) if the following holds:*

$$(a_{11}, \ldots, a_{m1}), \ldots, (a_{1n}, \ldots, a_{mn}) \in R \Rightarrow (f(a_{11}, \ldots, a_{1n}), \ldots, f(a_{m1}, \ldots, a_{mn})) \in R$$

With $\mathrm{Pol}\,\Gamma := \{f \mid$ *every* $R \in \Gamma$ *is invariant under* $f\}$ *we denote the set of all polymorphisms of all relations from* Γ *and with* $\mathrm{Inv}\,F := \{R \mid R$ *is invariant under every* $f \in F\}$ *we denote the set of all relations invariant under all operations from* F*.*

We say a set F of operations on A is closed under *superposition* if F contains the identity and for any n-ary operation $f \in F$ and m-ary operation $g \in F$ the following holds:

- h defined by $h(x_1, \ldots, x_{n-1}, y_1, \ldots, y_m) = f(x_1 \ldots, x_{n-1}, g(y_1, \ldots, y_m))$ is in F.
- h defined by $h(x_1, \ldots, x_n) = f(x_{\Pi(1)}, \ldots, x_{\Pi(n)})$, where Π is a permutation on $\{1, \ldots, n\}$, belongs to F.
- h defined by $h(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, x_{n-1})$ is in F.
- h defined by $h(x_1, \ldots, x_{n+1}) = f(x_1, \ldots, x_n)$ is in F.

Such closed sets of operations on A are called *clones* and they form a lattice. It holds that Pol Inv F is the smallest clone containing F. Post identified all Boolean clones and their inclusion structure, moreover he gave finite bases for each clone. The lattice of Boolean clones is now known as *Post's lattice*. While the number of clones in Post's lattice is countable, it has been proven that the lattices of clones over a domain containing more than 2 elements contain uncountably many clones and thus not every clone has a finite base. However, only the lattice of Boolean clones is fully specified.

Corresponding to superposition we say a constraint language Γ on A is closed if for any n-ary relation $R_1 \in \Gamma$ and for any $R_2 \in \Gamma$ holds:

- $A^k \in \Gamma$ for $k \in \mathbb{N}$.
- $R_1 \times R_2 \in \Gamma$.
- $\{(x_{\Pi(1)}, \ldots, x_{\Pi(n)}) \mid (x_1, \ldots, x_n) \in R_1\} \in \Gamma$ for every permutation Π on $\{1, \ldots, n\}$.
- $\{(x_1, \ldots, x_{n-1}) \mid$ there is an $x \in A$ such that $(x_1, \ldots, x_{n-1}, x) \in R_1\} \in \Gamma$.
- $\{(x_1, \ldots, x_{n-1}, x_{n-1}) \mid (x_1, \ldots, x_{n-1}, x_{n-1}) \in R_1\} \in \Gamma$.

Such closed constraint languages are called *coclones* and the smallest coclone which contains a constraint language Γ is Inv Pol Γ.

It is easy to see that Pol Γ always is a clone and Inv F is a coclone for any set of operations F. The following theorem shows that there is a direct correspondence between clones and coclones.

Theorem 1 ([Dal00]) Pol *and* Inv *form a* Galois connection *between the lattice of clones on a domain A and the lattice of coclones on A, that means:*

1. *For all clones F_1, F_2 it holds that: if $F_1 \subseteq F_2$ then* Inv $F_2 \subseteq$ Inv F_1.
2. *For all coclones Γ_1, Γ_2 it holds that: if $\Gamma_1 \subseteq \Gamma_2$ then* Pol $\Gamma_2 \subseteq$ Pol Γ_1.
3. *For all clones F and coclones Γ it holds that: $F \subseteq$* Pol Inv F *and* $\Gamma \subseteq$ Inv Pol Γ.

Because of this connection we can transfer results about clones on coclones and vice versa. With the following theorem from [Jea98] we see that the polymorphisms of a constraint language fully determine its complexity. Therefore its algebraic structure is very interesting.

Theorem 2 ([Jea98]) *Let Γ be a constraint language on a finite set. Γ is tractable (NP-complete) if and only if* Inv Pol (Γ) *is tractable (NP-complete).*

Definition *A pair* $\mathbb{A} = (A, F)$ *such that* A *is a nonempty set and* F *is a family of finitary operations on* A *is an* algebra*.* A *is called the* universe *of* $\mathbb{A}$ *and the operations from* F *are called* basic operations*. We say* $\mathbb{A}$ *is a* finite algebra *if* A *is finite.*

Given a constraint language Γ on A, we call the algebra $\mathbb{A}_\Gamma := (A, \text{Pol}\,\Gamma)$ the *algebra associated with* Γ. In this way any algebra is assigned to a constraint language.

So the complexity of a constraint language Γ depends on $\mathbb{A}_\Gamma$ only and constraint languages that give raise to an algebra $\mathbb{A} = (A, F)$ have the same complexity as Inv F. We say $\mathbb{A}$ is tractable (NP-complete) if Inv F is tractable (NP-complete). Operations from Pol Inv F are called term operations of $\mathbb{A}$.

We now define some properties of operations:

Definition *Let* f *be an* n*-ary operation on a finite set* A*.* f *is called*

- *a* projection *if there is* $i \in \{1, \ldots, n\}$ *such that for any* $x_1, \ldots, x_n \in A$ *it holds that* $f(x_1, \ldots, x_n) = x_i$*;*
- essentially unary *if there is a unary operation* g *such that for any* $x_1, \ldots, x_n \in A$ *it holds that* $f(x_1, \ldots, x_n) = g(x_i)$*;*
- *a* constant *operation if there is* $c \in A$ *such that for any* $x_1, \ldots, x_n \in A$ *it holds that* $f(x_1, \ldots, x_n) = c$ *;*
- idempotent *if* $f(x, \ldots, x) = x$ *for any* $x \in A$*;*
- semilattice *if it is binary and satisfies the following conditions for any* $x, y, z \in A$*:*
  - *(a)* $f(x, (y, z)) = f(f(x, y), z)$ *(Associativity),*
  - *(b)* $f(x, y) = f(y, x)$ *(Commutativity),*
  - *(c)* $f(x, x) = x$ *(Idempotency);*
- conservative commutative *if it is binary,* $f(x, y) = f(y, x)$ *and* $f(x, y) \in \{x, y\}$ *for any* $x, y \in A$*;*
- *a* majority *operation if it is ternary and* $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$ *for any* $x, y \in A$*;*
- affine *if it is ternary and* $f(x, y, z) = x - y + z$ *for any* $x, y, z \in A$*, where* $+, -$ *are the operations of an abelian group;*
- Malt'sev *if it is ternary and* $f(x, y, y) = f(y, y, x) = x$ *for any* $x, y \in A$*.*

Note that every affine operation is also Malt'sev.

Proposition 3 ([Bul02b, BJ00, JCC98, JCG97]) *Let Γ be a constraint language. If $\mathbb{A}_\Gamma$ has a term operation that is constant, semilattice, conservative commutative, Malt'sev, or majority, then it is tractable.*

2.2 The Boolean Case

We say a CSP(Γ) is a Boolean constraint satisfaction problem if Γ is a constraint language over the Boolean domain $\{0, 1\}$. The following famous result classifies the complexity of the Boolean constraint languages.

Theorem 4 (Schaefers Dichotomy Theorem) *Let Γ be a constraint language over the Boolean domain. If Γ satisfies one of the following conditions then it is tractable.*

1. *$\mathbb{A}_\Gamma$ has the constant term operation 0.*
2. *$\mathbb{A}_\Gamma$ has the constant term operation 1.*
3. *$\wedge$ is a term operation of $\mathbb{A}_\Gamma$.*
4. *$\vee$ is a term operation of $\mathbb{A}_\Gamma$.*
5. *f with $f(x, y, z) = (x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$ is a term operation of $\mathbb{A}_\Gamma$.*
6. *f with $f(x, y, z) = x - y + z (mod\ 2)$ is a term operation of $\mathbb{A}_\Gamma$.*

In any other case Γ is NP-complete.

If Γ satisfies the condition 1 resp. 2 it is called 0-valid resp. 1-valid and if it satisfies one of the conditions 3-4 it is called *Schaefer*.

It holds that $\wedge$ and $\vee$ are semilattice operations. The operation from condition 5 is a majority operation and that one from condition 6 is Malt'sev. So in all polynomial cases from Theorem 4 there is a term operation with a property mentioned in Proposition 3.

2.3 G-Sets

Definition *We say an algebra is a* G-Set *if every of its term operations is essentially unary and the corresponding unary operation is a permutation.*

G-Sets are are important for finding NP-complete algebras. The following proposition was proved in [JCG97].

Proposition 5 *A finite G-Set with at least two elements is NP-complete.*

In Post's Lattice there are only two clones, N_2 and I_2 (notation from [BCRV03]) such that the corresponding algebras $(\{0,1\},N_2)$ and $(\{0,1\},I_2)$ are G-Sets. It holds that N_2 and I_2 are the only clones with Inv N_2 and Inv I_2 are neither 0-valid nor 1-valid nor Schaefer, and therefore NP-complete. So we can restate Schaefer's Dichotomy Theorem.

Theorem 6 ([BJK00]) *A Boolean constraint language* Γ *is tractable if* $\mathbb{A}_\Gamma$ *is not a G-Set. Otherwise* Γ *is NP-complete.*

Definition *Let* $\mathbb{A} = (A, F)$ *be an algebra and* $B \subseteq A$ *such that, for any* $f \in F$ *and any* $b_1, \ldots, b_n \in B$*, it holds that* $f(b_1, \ldots, b_n) \in B$*. Let* $F|_B := \{f|_B \mid f \in F\}$ *be the set of basic operations restricted to* B*. Then we call the algebra* $\mathbb{B} = (B, F|_B)$ *a* subalgebra *of* $\mathbb{A}$ *and* B *a* subuniverse *of* $\mathbb{A}$*.*

Definition *Let* $\mathbb{A} = (A, F)$ *be an algebra. An equivalence relation* $\theta \in \mathrm{Inv}\ F$ *is said to be a congruence of* $\mathbb{A}$*. The equivalence class containing* $a \in A$ *is denoted by* a^θ *and we call* $A/_\theta := \{a^\theta \mid a \in A\}$ *a* factor-set *of* $\mathbb{A}$*.* $\mathbb{A}/_\theta = (A/_\theta, F^\theta)$ *with* $F^\theta = \{f^\theta | f \in F\}$*, where* $f^\theta(a_1^\theta, \ldots, a_n^\theta) = (f(a_1, \ldots, a_n))^\theta$*, is said to be a* factor-algebra *of* $\mathbb{A}$*. Observe that the operations are well-defined because* $\theta \in \mathrm{Inv}\ F$*.*

We call every factor-algebra of a subalgebra of $\mathbb{A}$ *a* factor *of* $\mathbb{A}$*.*

Proposition 7 ([BJK00]) *Let* Γ *be a tractable constraint language,* B *a subuniverse of* $\mathbb{A}_\Gamma$ *and* θ *a congruence of* $\mathbb{A}_\Gamma$*. Then*

1. $\mathbb{B} = (B, (\mathrm{Pol}\ \Gamma)|_B)$ *is a tractable subalgebra of* $\mathbb{A}_\Gamma$*;*
2. $\mathbb{A}_\Gamma/_\theta$ *is tractable.*

As a conclusion from the last proposition we get that every factor of a tractable algebra is tractable. Hence, every tractable algebra satisfies the following condition: none of its factors with at least two elements is a G-Set. We write (No-G-Set) for this condition.

In the special case of 3-element algebras factors have a simple structure.

Proposition 8 *Let* $\mathbb{A}$ *be a 3-element algebra and let* $\mathbb{B}$ *be a factor of* $\mathbb{A}$ *with at least two elements. Then one of the following conditions holds.*

1. $\mathbb{B} = \mathbb{A}$*;*
2. $\mathbb{B}$ *is a 2-element subalgebra of* $\mathbb{A}$*;*
3. $\mathbb{B}$ *is a 2-element factor-algebra of* $\mathbb{A}$*.*

Proof Let $\mathbb{A}$ be a 3-element algebra and $\mathbb{B}$ a factor of $\mathbb{A}$. That means there is an algebra $\mathbb{C}$ such that $\mathbb{B}$ is a factor-algebra of $\mathbb{C}$ and $\mathbb{C}$ is a subalgebra of $\mathbb{A}$. Let θ be a congruence of $\mathbb{C}$ such that $\mathbb{B} = \mathbb{C}/_{\theta}$.

Case 1: $\mathbb{C}$ is a 3-element algebra. Then it holds that $\mathbb{C} = \mathbb{A}$. If θ has three equivalence classes we have $\mathbb{B} = \mathbb{C}/_{\theta} = \mathbb{C} = \mathbb{A}$. If θ has two equivalence classes then $\mathbb{B}$ is a 2-element factor-algebra of $\mathbb{C} = \mathbb{A}$. For fewer equivalence classes $\mathbb{B}$ has less than two elements.

Case 2: $\mathbb{C}$ is a 2-element algebra. If θ has two equivalence classes then it holds that $\mathbb{B} = \mathbb{C}/_{\theta} = \mathbb{C}$ is a 2-element subalgebra of $\mathbb{A}$. Otherwise $\mathbb{B}$ has less than two elements.

Other cases lead to factors with only one element. □

Definition *Let Γ be a constraint language and $f \in \mathrm{Pol}\ \Gamma$ of minimal range such that f is unary and $f \circ f = f$. Then we say $f(\Gamma) = \{f(R) | R \in \Gamma\}$ where $f(R) = \{(f(x_1), \ldots, f(x_n)) | (x_1, \ldots, x_n) \in R\}$ is a* core *of Γ. We denote the constraint language $\Gamma \cup \{\{(a)\} \mid a \in A\}$ by Γ^+ and $f(\Gamma)^+$ by Γ_f^{id}.*

Example The following table contains all unary operations f on $\{0, 1, 2\}$ with $f \circ f = f$.

f	$f(0)$	$f(1)$	$f(2)$
c_0	0	0	0
c_1	1	1	1
c_2	2	2	2
$g_{0,1}$	1	1	2
$g_{0,2}$	2	1	2
$g_{1,0}$	0	0	2
$g_{1,2}$	0	2	2
$g_{2,0}$	0	1	0
$g_{2,1}$	0	1	1
h	0	1	2

The operations with a range of 1 are trivially the constants and the only operation in the table with a range of 3 is the identity. We see that the operations in the table with a range of 2 are of the form $g_{a,b}(x) = x$ if $x \in \{0, 1, 2\} \setminus \{a\}$ and $g_{a,b}(a) = b$ for $a, b \in \{0, 1, 2\}$, $a \neq b$.

Let Γ be a constraint language, f a polymorphism of Γ which satisfies the conditions in the previous definition, and $\mathcal{P} = (V, A, \mathcal{C})$ a problem instance with constraint relations from Γ. If φ is a solution of $\mathcal{P}$, then $\psi = f \circ \varphi$ also is a solution of $\mathcal{P}$. Since every solution of $\mathcal{P}' = (V, f(A), \mathcal{C}')$ where $\mathcal{C}' = \{\langle s, f(R)\rangle \mid \langle s, R\rangle \in \mathcal{C}\}$ is a solution of $\mathcal{P}$, we have that $\mathcal{P}$ has a solution if and only if $\mathcal{P}'$ has one. In [BJK00] the following proposition was shown.

Proposition 9 *Let Γ be a constraint language on A and let $f \in \mathrm{Pol}\,\Gamma$ be a unary operation with minimal range such that $f \circ f = f$. Then Γ is tractable (resp. NP-complete) if and only if Γ_f^{id} is tractable (resp. NP-complete).*

The complexity of a core $f(\Gamma)$ does not depend on the choice of f, moreover for two cores $f_1(\Gamma)$ and $f_2(\Gamma)$ it holds that $f_1(f_2(\Gamma)) = f_1(\Gamma)$. Therefore we denote Γ_f^{id} by Γ^{id}. Notice that the term operations of $\mathbb{A}_{\Gamma+}$ are exactly those from $\mathbb{A}_\Gamma$ which are idempotent. If every term operation of an algebra is idempotent we call the algebra *idempotent*. Thus, $\mathbb{A}_{\Gamma_f^{\mathrm{id}}}$ is idempotent. The next proposition will be needed later.

Proposition 10 *Let Γ be a constraint language on a finite domain such that Γ is its own core. Then all unary polymorphisms of Γ are permutations.*

Proof Let Γ be a constraint language on a finite domain A that is its own core and assume there is an unary polymorphism of Γ that is no permutation. Let f be an operation with minimal range l under those unary polymorphisms and let $k = |A|$. Then $l < k$ and since the range of f is minimal, it holds that $f \circ f$ has range l. That means $f|_{f(A)}$ is a permutation on $f(A)$. Therefore $f^{|f(A)|!}|_{f(A)}$ is the identity on $f(A)$. The range of $f^{|f(A)|!}$ is l and it holds that $f^{|f(A)|!} \circ f^{|f(A)|!} = f^{|f(A)|!}$. Thus $f^{|f(A)|!}(\Gamma)$ is a core of Γ. But $f^{|f(A)|!}(\Gamma)$ is different from Γ, therefore Γ cannot be its own core. This is a contradiction. □

In [Bul02c] Bulatov states the conjecture that every algebra Γ such that Γ^{id} satisfies (No-G-Set) is tractable and all others are NP-complete. In the next section we will see, that this conjecture holds for 3-element algebras.

3 The 3-Element Case

3.1 Properties Providing Tractability

In [Bul02c] Bulatov defined seven properties for 3-element algebras that will guarantee tractability. If we want to know if an algebra $\mathbb{A}$ is tractable or NP-complete it is sufficient to determine the complexity of $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ because of Proposition 9. Therefore we only need to consider idempotent algebras. For an n-ary relation R, $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ such that $i_1 \leq \dots \leq i_k$, and $\overrightarrow{a} \in R$ we denote the tuple $(\overrightarrow{a}[i_1], \dots, \overrightarrow{a}[i_k])$ by $\overrightarrow{a}_I$ and the relation $\{\overrightarrow{a}_I \mid \overrightarrow{a} \in R\}$ by R_I. R is said to be *irreducible* if for any $i, j \leq n$ with $i \neq j$ holds that $R_{\{i,j\}}$ is not of the form $\{(a, f(a)) \mid a \in A\}$ for a bijective mapping $f : A \to B$.

Definition *Let $\mathbb{A} = (A, F)$ be an idempotent algebra with a 3-element universe such that $\mathbb{A}$ satisfies* (No-G-Set) *and let $U(R) = \{i \leq n \mid R_{\{i\}} = A\}$ for every n-ary relation R.*

1. *$\mathbb{A}$ satisfies the* partial zero property *if there exists a set of subuniverses Z such that $\forall B \in Z \; \exists z_B \in B$ such that (a) $A \in Z$ and (b) $\forall R \in \mathrm{Inv}\ F \; \forall \overrightarrow{a} \in R \; \exists \overrightarrow{b} \in R$ with*

$$\overrightarrow{b}[i] = \begin{cases} z_B, & \text{if } R_{\{i\}} = B \in Z \\ \overrightarrow{a}[i], & \text{otherwise} \end{cases}$$

2. *$\mathbb{A}$ satisfies the* splitting property *if for any n-ary relation $R \in \mathrm{Inv}\ F$ and for $N = \{1, \dots, n\} \setminus U(R)$ it holds that*

$$R = R_{U(R)} \times R_N \;\; \textit{and} \;\; R_{U(R)} = A^{|U(R)|}$$

Let $B \subseteq A$ be a 2-element subuniverse of $\mathbb{A}$ and $W(R) = \{i \leq n \mid B \subseteq R_{\{i\}}\}$ for every n-ary relation R. Let $\theta_B(R)$ be the equivalence relation on $W(R)$ generated by the set $\{(i, j) \mid \forall \overrightarrow{a} \in R : \overrightarrow{a}[i], \overrightarrow{a}[j] \in B \text{ or } \overrightarrow{a}[i], \overrightarrow{a}[j] \notin B\}$ and $W_1(R), \dots, W_k(R)$ the corresponding equivalence classes.

3. *Let $a \in A - B$, and $b \in B$. Then $\mathbb{A}$ satisfies the* $(a - b)$-replacement property *if $\forall R \in \mathrm{Inv}\ F \; \forall \overrightarrow{a} \in R \; \exists \overrightarrow{b} \in R$ with*

$$\overrightarrow{b}[i] = \begin{cases} b, & \text{if } \overrightarrow{a}[i] = a \text{ and } a, b \in R_{\{i\}} \\ \overrightarrow{a}[i], & \text{otherwise} \end{cases}$$

4. *$\mathbb{A}$ satisfies the* B-extendibility property *if $\forall R \in \mathrm{Inv}\ F$*
   - $\forall k \in W(R) \; \forall a \in B \; \exists \overrightarrow{a} \in R \; \forall i \in W(R) : \overrightarrow{a}[i] \in B$ *and* $\overrightarrow{a}[k] = a$
   - $\forall k, l \in W(R) \; \forall (a, b) \in R_{\{k,l\}} \; \exists \overrightarrow{a} \in R \; \forall i \in W(R) : \overrightarrow{a}[i] \in B, \overrightarrow{a}[k] = a$ *and* $\overrightarrow{a}[l] = b$
   - $\forall \overrightarrow{a} \in B^{|W(R)|}$ *with* $(\overrightarrow{a}[i], \overrightarrow{a}[j]) \in R_{\{i,j\}} \; \forall i, j \in W(R) \; \exists \overrightarrow{b} \in R$ *such that*

$$\overrightarrow{b}[i] = \begin{cases} \overrightarrow{a}[i], & \text{if } i \in W(R) \\ d, & \text{if } |R_{\{i\}}| = 2 \text{ and } \{d\} = R_{\{i\}} \cap B \\ d, & \text{if } \{d\} = R_{\{i\}} \end{cases}$$

5. $\mathbb{A}$ *satisfies the* B-rectangularity property *if* $\forall\, R \in \text{Inv } F$

   - $R_{W(R)} \cap B^{|W(R)|} = (R_{W_1(R)} \cap B^{|W_1(R)|}) \times \cdots \times (R_{W_k(R)} \cap B^{|W_k(R)|})$
   - $\forall \overrightarrow{a} \in R$ *such that* $\forall\, i \in W(R)\ \overrightarrow{a}[i] \in B\ \exists\, \overrightarrow{b} \in R$ *with*

$$\overrightarrow{b}[i] = \begin{cases} \overrightarrow{a}[i], & \text{if } i \in W(R) \text{ or } |R_{\{i\}}| = 1 \\ c, & \text{otherwise, with } \{c\} = B \cap R_{\{i\}} \end{cases}$$

6. $\mathbb{A}$ *satisfies the* B-semirectangularity property *if the equivalence relation* η *with classes* B *and* $A - B = \{c\}$ *is a congruence of* $\mathbb{A}$ *and if the following holds:* $\forall\, R \in \text{Inv } F\ \forall\, \overrightarrow{a} \in R\ \forall j \in \{1, \ldots, k\}\ \forall \overrightarrow{a}_j \in R_{W_j(R)} \cap B^{|W_j(R)|}\ \exists\, \overrightarrow{b} \in R$ *with*

$$\overrightarrow{b}[i] = \begin{cases} \overrightarrow{a}[i], & \text{if } B \not\subseteq R_{\{i\}} \\ c, & \text{if } B \subseteq R_{\{i\}} \text{ and } \overrightarrow{a}[i] = c \\ \overrightarrow{a}_j[i], & \text{if } i \in W(R) \text{ and } \overrightarrow{a}[i] \in B. \end{cases}$$

7. $\mathbb{A}$ *satisfies the* B-semisplitting property *if for any* n*-ary* $R \in \text{Inv } F$ *such that* R *is irreducible and* $N = \{1, \ldots, n\} \setminus U(R)$ *the following holds:*

   - $(R_{U(R)} \cap B^{|U(R)|}) \times R_N \subseteq R$ *and*
   - $\forall i, j \in U(R)\ \forall\, (a_i, a_j) \in R_{\{i,j\}} \cap B^2$ *there is* $\overrightarrow{a} \in R_{U(R)} \cap B^{|U(R)|}$ *such that* $\overrightarrow{a}[i] = a_i$, $\overrightarrow{a}[j] = a_j$.

In [Bul02c] Bulatov proved that an algebra that satisfies one of this properties is tractable.

Theorem 11 *Let* $\mathbb{A} = (A, F)$ *be an idempotent algebra with a* 3*-element universe such that* $\mathbb{A}$ *satisfies* (No-G-Set) *and one of the seven properties of the previous definition. Then* $\mathbb{A}$ *is tractable.*

To prove this theorem, Bulatov gives a polynomial time algorithm for each property that solves the constraint satisfaction problem for problem instances with constraint relations that give raise to an algebra satisfying the property. First we present an example, then we take a closer look on two of the properties. We write an n-ary relation R as $n \times m$-matrix where m is the cardinality of R and every tuple in R is represented by a column in the matrix.

Example

$$\text{Let } R_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 \end{pmatrix}, R_2 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}, R_3 = (2).$$

Let $\Gamma = \{R_1, R_2, R_3\}$ be a constraint language. It is obvious to see that all term operations of $\mathbb{A}_\Gamma$ are idempotent.
We examine $\mathbb{A}_\Gamma$ for subalgebras: Since $\mathbb{A}_\Gamma$ is idempotent it holds that $\{0\}, \{1\}$, and $\{2\}$ are subuniverses of $\mathbb{A}_\Gamma$. Let f be an n-ary term operation of $\mathbb{A}_\Gamma$. $\{0,1\}$ is a subuniverse of $\mathbb{A}_\Gamma$ as well: Assume there is $\overrightarrow{a} = (a_1, \dots, a_n) \in \{0,1\}^n$ such that $f(\overrightarrow{a}) = 2$. Without loss of generality let $a_1 = \dots = a_k = 0$ and $a_{k+1} = \dots = a_n = 1$ for a k such that $1 \leq k \leq n$. Let g be the term operation defined by

$$g(x, y) = f(\underbrace{x, \dots, x}_{k}, \underbrace{y, \dots, y}_{n-k}).$$

Then it holds that $g(0,1) = 2$ and $g(0,0) = 0$. But that is a contradiction because $(0,0,0), (0,1,1) \in R_1$, and $(g(0,0), g(0,1), g(0,1)) = (0,2,2) \notin R_1$. Thus, $\{0,1\}$ is a subuniverse of $\mathbb{A}_\Gamma$.
Analogously $\{0,2\}$ is a subuniverse because $(0,0,1), (2,0,1) \in R_1$ but $(1,0,1) \notin R_1$.

$\{1,2\}$ is not a subuniverse of $\mathbb{A}_\Gamma$ because f_0 defined by the table

f_0	0	1	2
0	0	0	0
1	1	1	0
2	0	1	2

is a term operation of $\mathbb{A}_\Gamma$.

Now we examine $\mathbb{A}_\Gamma$ for factor algebras. We are only interested in factor algebras that have 2 elements and therefore we consider the equivalence relations on $\{0,1,2\}$ with two equivalence classes. They are of the form Let $Q_c = \begin{pmatrix} a & a & b & b & c \\ a & b & a & b & c \end{pmatrix}$ with $\{a,b,c\} = \{0,1,2\}$. If $c = 0$ it holds that $(1,1), (1,2) \in Q_0$, but $(f_0(1,1), f_0(1,2)) = (1,0) \notin Q_0$. For $c = 1$ it holds that $(1,0), (1,2) \in Q_1$, but $(f_0(1,0), f_0(1,2)) = (1,0) \notin Q_1$. It can be verified that the operation f_1 defined by

f_1	0	1	2
0	0	1	0
1	1	1	2
2	0	2	2

is a term operation of $\mathbb{A}_\Gamma$. If $c = 2$ we have that $(2,0), (2,1) \in Q_2$, but $(f_1(2,0), f_1(2,1)) = (0,2) \notin Q_2$. So we have no congruences with two equivalence classes, hence there are no 2-element factor algebras of $\mathbb{A}_\Gamma$.

To verify that $\mathbb{A}_\Gamma$ satisfies (No-G-Set), we must examine all factors with at least to elements for G-Sets. The interesting factors are $\mathbb{A}_\Gamma$ itself and the subalgebras with the universes $\{0,1\}$ and $\{0,2\}$. $\mathbb{A}_\Gamma$ is no G-Set because f_0 is not essentially unary and the two subalgebras are no G-Sets because $f_1|_{\{0,1\}}$ and $f_1|_{\{0,2\}}$ are not essentially unary.
We prove that $\mathbb{A}_\Gamma$ satisfies the partial zero property. Let $Z = \{\{0,1,2\}, \{0,2\}\}$ and $z_A = z_{\{0,2\}} = 0$. Now let R be an n-ary relation that is invariant under all term operations of $\mathbb{A}_\Gamma$ and let $\overrightarrow{a} \in R$. We define $\overrightarrow{a}_0 = \overrightarrow{a}$ and $\overrightarrow{a}_i = f_0(f_0(\overrightarrow{a}_{i-1}, \overrightarrow{b}_i), \overrightarrow{c}_i)$ for

$1 \leq i \leq n$ and $\overrightarrow{b}_i, \overrightarrow{c}_i \in R$ such that $\overrightarrow{b}_i[i] = 0$, $\overrightarrow{c}_i[i] = 2$ if $R_{\{i\}} \in Z$ and $\overrightarrow{b} = \overrightarrow{c} = \overrightarrow{a}_{i-1}$ otherwise. Since $f_0(f_0(x, 0), 2) = 0$ and $f(0, x) = 0$ for any $x \in \{0, 1, 2\}$, it holds that $\overrightarrow{a}_n[i] = 0$ for any $1 \leq i \leq n$ with $R_{\{i\}} \in Z$. It is obvious that $\overrightarrow{a}_n[i] = \overrightarrow{a}[i]$ for any $1 \leq i \leq n$ with $|R_{\{i\}}| = 1$. Because $f_0(x, y) = x$ for $x, y \in \{0, 1\}$, we have that $\overrightarrow{a}_n[i] = \overrightarrow{a}[i]$ for any $1 \leq i \leq n$ with $R_{\{i\}} = \{0, 1\}$. The last case, $R_{\{i\}} = \{1, 2\}$, can not occur since $\{1, 2\}$ is not a subuniverse of $\mathbb{A}_\Gamma$. Thus we have proved that $\mathbb{A}_\Gamma$ satisfies the partial zero property.

Most of the polynomial time algorithms for the properties modify the given problem instance in such way that it can be solved with the knowledge about Boolean constraint satisfaction problems. We need some more vocabulary:

Definition *Let $\mathcal{P} = (V, A, \mathcal{C})$ be a problem instance and $W \subseteq V$. Then $\mathcal{P}_W = (W, A, \mathcal{C}_W)$ with $\mathcal{C}_W = \{\langle s \cap W, R_{s \cap W} \rangle \mid \langle s, R \rangle \in \mathcal{C}\}$ is called a* restricted problem instance *of $\mathcal{P}$. A solution of $\mathcal{P}_W$ is said to be a* partial solution *of $\mathcal{P}$ on W and we denote the set of all partial solutions by S_W.*

Definition *We say a problem instance $\mathcal{P} = (V, A, \mathcal{C})$ is* 2-valued *if for any $v \in V$ it holds that $|S_{\{v\}}| \leq 2$.*

A 2-valued problem instance over a constraint language that gives raise to an idempotent algebra satisfying (No-G-Set) can be solved in polynomial time ([Bul02c]).

Definition *A problem instance $\mathcal{P} = (V, A, \mathcal{C})$ is said to be k-*minimal *if for any k-element subset $W \subseteq V$ the following conditions holds:*

- *there is a constraint $\langle s, R \rangle \in \mathcal{C}$ such that $W \subseteq s$;*
- *for any $\langle s, R \rangle \in \mathcal{C}$ and any $\overrightarrow{a} \in R$ there exists an partial solution $\varphi \in S_W$ with $\varphi(s[i]) = \overrightarrow{a}[i]$ for any i such that $s[i] \in W$.*

To every problem instance $\mathcal{P}$ and every $k \in \mathbb{N}$ there is a k-minimal problem instance $\mathcal{P}'$, which has the same solutions as $\mathcal{P}$ and can be obtained from $\mathcal{P}$ in polynomial time (for an algorithm see [Bul02c]).

3.1.1 The Partial Zero Property

Let $\mathbb{A} = (A, F)$ be a 3-element algebra that satisfies the partial zero property and let $\mathcal{P} = (V; A; \mathcal{C})$ be a problem instance with constraint relations from Inv F. We give Bulatov's proof that $\mathbb{A}$ is tractable by reducing $\mathcal{P}$ to a 2-valued instance:

Without loss of generality assume $\mathcal{P}$ to be 1-minimal. Because $\mathbb{A}$ satisfies the partial zero property there exists a set of subuniverses Z such that

(a) $A \in Z$ and

(b) $\forall B \in Z \ \exists z_B \in B$ such that $\forall R \in \text{Inv } F \ \forall \overrightarrow{a} \in R \ \exists \overrightarrow{b} \in R$ with

$$\overrightarrow{b}[i] = \begin{cases} z_B, & \text{if } R_{\{i\}} = B \in Z \\ \overrightarrow{a}[i], & \text{otherwise} \end{cases} .$$

Let $T_{\{v\}} = \{\varphi(v) \mid \varphi \in S_{\{v\}}\}$. We prove: If φ is a solution of $\mathcal{P}$ then ψ with

$$\psi(v) = \begin{cases} z_{T_{\{v\}}}, & \text{if } T_{\{v\}} \in Z \\ \varphi(v), & \text{otherwise} \end{cases}$$

is a solution of $\mathcal{P}$. Let φ be a solution of $\mathcal{P}$, $\langle s, R\rangle \in \mathcal{C}$ and $\overrightarrow{a} = \varphi(s)$. Then holds that $\overrightarrow{b}$ such that

$$\overrightarrow{b}[i] = \begin{cases} z_B, & \text{if } R_{\{i\}} = B \\ \overrightarrow{a}[i], & \text{otherwise} \end{cases}$$

contains to R. Let $v = s[i]$. Because $\mathcal{P}$ is 1-minimal, it holds that $\overrightarrow{c}[i] \in T_{\{v\}}$ for any $\overrightarrow{c} \in R$. That means $R_{\{i\}} \subseteq T_{\{v\}}$. Trivially it holds that $T_{\{v\}} \subseteq R_{\{i\}}$, so we have $R_{\{i\}} = S_{\{v\}}$. Hence, $\psi(s) = \overrightarrow{b} \in R$ and ψ is a solution of $\mathcal{P}$.

We reduce $\mathcal{P}$ to the problem instance $\mathcal{P}' = (V; A; \mathcal{C}')$ with $\mathcal{C}' := \mathcal{C} \cup \{\langle s, R'\rangle | \langle s, R\rangle \in \mathcal{C}\}$, where $R' := \{a \in R \mid a[i] = z_{T_{\{v\}}}$ for every v with $T_{\{v\}} \in Z$ and i such that $s[i] = v\}$: Since $\mathcal{C} \subseteq \mathcal{C}'$ it is obvious that every solution of $\mathcal{P}'$ also is a solution of $\mathcal{P}$ and we have proved that for every solution φ of $\mathcal{P}$ there exists a solution ψ of $\mathcal{P}'$.

It holds that $\mathcal{P}'$ is 2-valued: Assume there is a $v \in V$ with $T_{\{v\}} = A \in Z$. Because $\mathcal{P}$ is 1-minimal, there is a $\langle s, R\rangle \in \mathcal{C}$ with $s[i] = v$ for an i. Then there exists $\langle s, R'\rangle \in \mathcal{C}'$ with $R'_{\{i\}} = z_{T_{\{v\}}}$ Since $T_{\{v\}} \subseteq R'_{\{i\}}$ we have a contradiction.

Thus $\mathbb{A}$ is tractable.

3.1.2 The Splitting Property

Let $\mathbb{A} = (A, F)$ be a 3-element algebra that satisfies the splitting property. Let $\Gamma = \text{Inv } F$ and $\Gamma' = \{R_N \mid R \in \Gamma\}$, where $U(R) = \{i \leq n \mid R_{\{i\}} = A\}$ and $N = \{1, \ldots, n\} \setminus U(R)$ for every n-ary relation R. Any problem instance $\mathcal{P} = (V, A, \mathcal{C})$ with constraint relations from Γ has exactly the same solutions as the instance $\mathcal{P}' = (V, A, \mathcal{C}')$ over Γ' where $\mathcal{C}' = \{\langle s, R_N\rangle \mid \langle s, R\rangle \in \mathcal{C}\}$. It is easy to see, that $\mathcal{P}'$ is 2-valued and that therefore $\mathbb{A}$ is tractable.

3.2 Bulatov's Dichotomy Theorem

The following theorem is the main result from [Bul02c].

Theorem 12 *Let $\mathbb{A} = (A, F)$ be an idempotent 3-element algebra that satisfies the condition* (No-G-Set). *Then there exists a subset of basic operations $F' \subseteq F$ such that the algebra $\mathbb{A}' = (A, F')$ satisfies* (No-G-Set) *and one of the following conditions holds.*

1. *$\mathbb{A}'$ satisfies the partial zero property*
2. *$\mathbb{A}'$ satisfies the splitting property*
3. *$\mathbb{A}'$ satisfies the $(a-b)$-replacement property for $A = \{a, b, c\}$ and $\{b, c\}$ subuniverse of $\mathbb{A}'$*
4. *$\mathbb{A}'$ satisfies the B-extendibility property for a 2-element subuniverse B of $\mathbb{A}'$*
5. *$\mathbb{A}'$ satisfies the B-rectangularity property for a 2-element subuniverse B of $\mathbb{A}'$*
6. *$\mathbb{A}'$ satisfies the B-semirectangularity property for a 2-element subuniverse B of $\mathbb{A}'$*
7. *$\mathbb{A}'$ satisfies the B-semisplitting property for a 2-element subuniverse B of $\mathbb{A}'$*
8. *$\mathbb{A}'$ has a majority term operation*
9. *$\mathbb{A}'$ has a conservative commutative term operation*
10. *$\mathbb{A}'$ has a Malt'sev term operation*

Hence, the following corollary is a conclusion from Proposition 9 and Theorems 11 and 12.

Corollary 13 *A 3-element constraint language Γ is tractable if $\mathbb{A}_{\Gamma^{\text{id}}}$ satisfies* (No-G-Set) *and NP-complete otherwise.*

In case that P≠NP, Γ is tractable if and only if one of the following conditions holds:

- $\mathbb{A}_{\Gamma^{\text{id}}}$ is a 3-element algebra satisfying one of the ten conditions from Theorem 12; or
- $\mathbb{A}_{\Gamma^{\text{id}}}$ is a 2-element algebra with a semilattice, majority or affine polymorphism in its term operations; or
- $\mathbb{A}_{\Gamma^{\text{id}}}$ is a 1-element algebra.

To verify if an algebra satisfies conditions 7,8 or 9 of Theorem 12 we have to find only a term operation that has a certain property, but to verify conditions 1 - 6 is more complicated, because it is not enough to consider the term operations separated, but we need to consider all together.

The proof of Theorem 12 is very technical.

3.3 NP-complete clones

Let f be an operation on A. We denote with u_f the unary operation on A defined by $u_f(x) = f(x, \dots, x)$ for any $x \in A$. If f is essentially unary and the corresponding unary operation is a permutation we say f is an essentially unary permutation.

Lemma 14 *Let Γ be a constraint language on a finite domain A such that every unary polymorphism is a permutation. Then every polymorphism of Γ is an essentially unary permutation if every polymorphism of Γ^+ is essentially unary.*

Proof Let $f \in \mathrm{Pol}\,\Gamma$ be an n-ary operation. Then u_f is a polymorphism of Γ. Now let $d = |A|!$. Since u_f is a permutation on A, ${u_f}^d$ is the identity operation. Because ${u_f}^d(x) = {u_f}^{d-1} \circ f(x, \dots, x)$ for any $x \in A$ it holds that ${u_f}^{d-1} \circ f$ is idempotent and therefore a polymorphism of Γ^+. Thus ${u_f}^{d-1} \circ f$ is an essentially unary permutation. That means there exists a permutation $g \in \mathrm{Pol}\,\Gamma$ and $i \in \{1, \dots, n\}$ such that $g(x_i) = {u_f}^{d-1} \circ f(x_1, \dots, x_n)$ for any $x_1, \dots, x_n \in A$. Thus $u_f \circ g(x_i) = {u_f}^d \circ f(x_1, \dots, x_n) = f(x_1, \dots, x_n)$ for any $x_1, \dots, x_n \in A$ and f is an essentially unary permutation. □

Theorem 15 *Let Γ be a constraint language on the domain $\{0, 1, 2\}$. Γ is NP-complete if the polymorphisms of Γ satisfy one of the following conditions:*

1. $\mathrm{Pol}\,\Gamma \subseteq C_1 = \{f \mid f$ *is an essentially unary permutation*$\}$.
2. $\mathrm{Pol}\,\Gamma \subseteq C_2^B = \{f \mid u_f$ *is a permutation and* $f|_B$ *is essentially unary*$\}$ *for a 2-element subset* $B \subseteq \{0, 1, 2\}$.
3. $\mathrm{Pol}\,\Gamma \subseteq C_3^B = \{f \mid u_f$ *is a permutation,* $f(x_1, \dots, x_{i-1}, a, x_{i+1}, \dots, x_n) = u_f(c) \Leftrightarrow f(x_1, \dots, x_{i-1}, b, x_{i+1}, \dots, x_n) = u_f(c)$ *for any* $1 \leq i \leq n$ *and* $g \circ u_f^5 \circ f|_B$ *is an essentially unary permutation on* $B\}$ *for* $\{0, 1, 2\} = \{a, b, c\}$, $B = \{b, c\}$ *and* g *the operation defined by* $g(x) = x$ *if* $x \in B$ *and* $g(a) = b$.
4. $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ *is a 2-element algebra that is a G-Set.*

In any other case Γ is tractable.

Proof Let Γ be a constraint language on $\{0,1,2\}$. We prove: $\mathbb{A}_{\Gamma^{\text{id}}}$ does not satisfy (No-G-Set) if and only if one of the four conditions is satisfied. Then this Theorem follows due to Corollary 13. First we assume that Γ^{id} does not satisfy (No-G-Set). That means that $\mathbb{A}_{\Gamma^{\text{id}}}$ has a factor $\mathbb{B} = (B, F_B)$ with at least 2 elements that is a G-set. Because $\mathbb{B}$ is a G-Set and $\mathbb{B}$ is idempotent, all term operations of $\mathbb{B}$ are essentially unary and the corresponding unary operation is the identity. There are 4 cases:

Case 1: $\mathbb{A}_{\Gamma^{\text{id}}}$ has 3 elements and is a G-Set itself. Because of Proposition 10 every unary polymorphism is a permutation. So with Lemma 14 follows that every polymorphism of Γ is essentially unary. Hence, $\Gamma \subseteq C_1$ an thus Γ satisfies condition 1.

Case 2: $\mathbb{A}_{\Gamma^{\text{id}}}$ has 3 elements and $\mathbb{B}$ is a 2-element subalgebra of $\mathbb{A}_{\Gamma^{\text{id}}}$. Then all unary polymorphisms of Γ are permutations because of Proposition 10. Let f be an idempotent polymorphism of Γ. It holds that $f|_B$ is an operation on B. Since $\mathbb{B}$ is a G-Set, $f|_B$ is essentially unary. With the same proof as of Lemma 14 it follows that $f|_B$ is essentially unary. Thus we know Pol $\Gamma \subseteq C_2^B$ and therefore condition 2 is satisfied.

Case 3: $\mathbb{A}_{\Gamma^{\text{id}}}$ has 3 elements and $\mathbb{B}$ is a 2-element factor-algebra of $\mathbb{A}_{\Gamma^{\text{id}}}$. So all unary polymorphisms of Γ are permutations due to Proposition 10, and there exists an equivalence relation $\theta = \{(a,a),(a,b),(b,a),(b,b),(c,c)\}$ which is a congruence of $\mathbb{A}_{\Gamma^{\text{id}}}$ such that $\mathbb{B} = \mathbb{A}_{\Gamma^{\text{id}}}/_{\theta}$ and $\{a,b,c\} = \{0,1,2\}$.

Let $f \in \text{Pol } \Gamma^{\text{id}}$ be an n-ary operation. We first prove that for any $1 \le i \le n$ the following holds:

$$f(x_1,\dots,x_{i-1},a,x_{i+1},\dots,x_n) = c \Leftrightarrow f(x_1,\dots,x_{i-1},b,x_{i+1},\dots,x_n) = c$$

We assume there are $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n \in \{0,1,2\}$ such that, without loss of generality, $f(x_1,\dots,x_{i-1},a,x_{i+1},\dots,x_n) = c$ and $f(x_1,\dots,x_{i-1},b,x_{i+1},\dots,x_n) \neq c$. Since $(a,b) \in \theta$ and $(x_k,x_k) \in \theta$ for any $k \in \{1,\dots,i-1,i+1,\dots,n\}$ and θ is invariant under f, it holds that

$$(f(x_1,\dots,x_{i-1},a,x_{i+1},\dots,x_n), f(x_1,\dots,x_{i-1},b,x_{i+1},\dots,x_n)) \in \theta,$$

which contradicts the assumptions made.

Now let $f \in \text{Pol } \Gamma$ be an n-ary operation. Since u_f is a permutation on $\{0,1,2\}$, u_f^6 is the identity operation, therefore $u_f^5 \circ f$ is idempotent. We define $\overrightarrow{a} = (x_1,\dots,x_{i-1},a,x_{i+1},\dots,x_n)$ and $\overrightarrow{b} = (x_1,\dots,x_{i-1},b,x_{i+1},\dots,x_n)$ for some $1 \le i \le n$ and $x_k \in A$ for any $k \in \{1,\dots,i-1,i+1,\dots,n\}$. It follows that $u_f^5 \circ$

$f(\overrightarrow{a}[1], \ldots, \overrightarrow{a}[n]) = c \Leftrightarrow {u_f}^5 \circ f(\overrightarrow{b}[1], \ldots, \overrightarrow{b}[n]) = c$. Since ${u_f}^6$ is the identity operation we have $f(\overrightarrow{a}[1], \ldots, \overrightarrow{a}[n]) = u_f(c) \Leftrightarrow f(\overrightarrow{b}[1], \ldots, \overrightarrow{b}[n]) = u_f(c)$.

Let $f \in \text{Pol}\ \Gamma^{\text{id}}$ and g be the operation defined by $g(x) = x$ if $x \in B$ and $g(a) = b$. Without loss of generality we choose b as representative of the equivalence class $\{a, b\}$ and assume that $B = \{b, c\}$. Then f^θ corresponds to $g \circ f|_B$. Since $\mathbb{B}$ is a G-Set, it holds that $g \circ f|_B$ is an essentially unary permutation on B.

So for an arbitrary operation $f \in \text{Pol}\ \Gamma$ it holds that $g \circ u_f^5 \circ f|_B$ is an essentially unary permutation on B.

Hence $\text{Pol}\ \Gamma \subseteq C_3^B$.

Case 4: $\mathbb{A}_{\Gamma^{\text{id}}}$ has 2 elements. Then $\mathbb{B} = \mathbb{A}_{\Gamma^{\text{id}}}$ and condition 4 is satisfied.

To show the other direction, now let one of the four conditions be satisfied.

Condition 1 If $\text{Pol}\ \Gamma \subseteq C_1$ then $\mathbb{A}_\Gamma$ is a G-Set and therefore does not satisfy (No-G-Set).

Condition 2 Let $\text{Pol}\ \Gamma \subseteq C_2^B = \{f \mid u_f \text{ is an permutation and } f|_B \text{ is essentially unary}\}$ for a 2-element subset $B \subseteq \{0, 1, 2\}$

Let $f \in \text{Pol}\ \Gamma$. If f is unary it holds that $f = u_f$ is a permutation and therefore we have $\Gamma^{\text{id}} = \Gamma^+$.

Now let $f \in \text{Pol}\ \Gamma^{\text{id}} \subseteq C_2^B$ an n-ary operation. Since $f|_B$ is essentially unary and idempotent, the corresponding unary operation of $f|_B$ is the identity on B. So $\mathbb{B} = (B, \{f|_B \mid f \in \text{Pol}\ \Gamma^{\text{id}}\})$ is a subalgebra of $\mathbb{A}_{\Gamma^{\text{id}}}$ that only has term operations that are essentially unary permutations. $\mathbb{B}$ is a factor of $\mathbb{A}_{\Gamma^{\text{id}}}$ and G-Set, thus $\mathbb{A}_{\Gamma^{\text{id}}}$ does not satisfy (No-G-Set).

Condition 3 Let $\text{Pol}\ \Gamma \subseteq C_3^B = \{f \mid u_f \text{ is a permutation}, f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) = u_f(c) \Leftrightarrow (x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n) = u_f(c)$ for any $1 \leq i \leq n$ and $g \circ u_f \circ f|_B$ is an essentially unary permutation on $B\}$ for $\{0, 1, 2\} = \{a, b, c\}$, $B = \{b, c\}$ and g the operation defined by $g(x) = x$ if $x \in B$ and $g(a) = b$.

Let $f \in \text{Pol}\ \Gamma$ be a unary permutation. It holds that $f = u_f$ is a permutation and therefore $\Gamma^{\text{id}} = \Gamma^+$.

Let now $f \in \text{Pol}\ \Gamma^{\text{id}} \subseteq C_3^B$ be an n-ary operation. We prove that f is a polymorphism of the equivalence relation $\theta = \begin{pmatrix} a & a & b & b & c \\ a & b & a & b & c \end{pmatrix}$.

Let $X_0 = (\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \dots, \begin{pmatrix} x_n \\ y_n \end{pmatrix}) \in \theta^n$ and let X_i be defined by $X_i[k] = X_{i-1}[k]$ if $k \neq i$ and $X_i[i] = \begin{pmatrix} x_i \\ x_i \end{pmatrix}$ for any $i, k \in \{1, \dots, n\}$. Since $\begin{pmatrix} x_i \\ x_i \end{pmatrix} \in \theta$ for any $i \in \{1, \dots, n\}$, it follows that $X_i \in \theta^n$ f. Notice that $x_i = c \Leftrightarrow y_i = c$ for any $i \in \{1, \dots, n\}$. Let $x = f(x_1, \dots, x_n)$. Because $f \in C_3^B$ the following holds:

$$f(X_0[1], \dots, X_0[n]) = \begin{pmatrix} x \\ c \end{pmatrix} \Leftrightarrow f(X_1[1], \dots, X_1[n]) = \begin{pmatrix} x \\ c \end{pmatrix}$$
$$\Leftrightarrow \dots \Leftrightarrow f(X_n[1], \dots, X_n[n]) = \begin{pmatrix} x \\ c \end{pmatrix} \Leftrightarrow x = c$$

Thus $f(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \dots, \begin{pmatrix} x_n \\ y_n \end{pmatrix}) \in \theta$.

It follows that $\mathbb{B} = \mathbb{A}_{\Gamma^{\mathrm{id}}}/_\theta$ is a factor-algebra of $\mathbb{A}_{\Gamma^{\mathrm{id}}}$. We identify the equivalence class $\{a, b\}$ with b and $\{c\}$ with c. Then f^θ corresponds to $g \circ f|_B$ and $\mathbb{B} = (\{b, c\}, \{g \circ f|_B \mid f \in \mathrm{Pol}\ \Gamma^{\mathrm{id}}\})$. Since for any $f \in \mathrm{Pol}\ \Gamma^{\mathrm{id}}$ holds that $g \circ u_f^5 \circ f|_B = g \circ f|_B$, every term operation of $\mathbb{B}$ is an essentially unary permutation and B is a G-Set. Therefore $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ does not satisfy (No-G-Set).

Condition 4 If $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ is a G-Set then it does not satisfy (No-G-Set).

□

4 Enumeration Algorithms

Bulatov's classification of constraint languages on a 3-element set says something about the complexity of deciding if there is a solution to an instance of CSP. But it says nothing about the complexity of finding solutions. We are interested in algorithms that give us all solutions of an CSP instance. Because in general the number of solutions is exponential in the size of the input we use a special notion of efficiency.

Definition *An algorithm that generates on the input of a CSP instance all solutions of the instance without duplicates is called an* enumeration algorithm *or a* generating algorithm. *An enumeration algorithm has* polynomial delay *if the time from start to the output of the first solution, the time between the output of two solutions and between the output of the last solution and the halting is bounded by a polynomial in the input size.*

In [CH97] there is an algorithm given that enumerates all solutions of a Boolean CSP instance. This algorithm has polynomial delay for instances over a constraint language that is Schaefer. We generalize it to an enumeration algorithm for CSPs over a 3-element set. The Procedure Generate enumerates the solutions of a problem instance, by printing for every solution φ of P the vector $\varphi(v_1, \ldots, v_n)$ where $(v_1, \ldots, v_n)$ is a list of all variables that belong to P. Its parameters are the problem instance P, a list M and the number of variables p. M is empty in the beginning and in every recursion step, the assignment of one variable is put at the front. For this we use Cons(x,M) that inserts x onto the head of the list M.

```
1  Input: problem instance (V,A,C), V = (v_1,...,v_n)
2  Output: all vectors that represent a solution of P
3  Begin
4     If P is satisfiable
5     Then Generate(P,(),n)
6  End
7
8  Procedure Generate(P,M,p)
9  Begin
10    If p = 0
11    Then Output(M)
12    Else Begin
13       If (V,A,C+{<(v_p),{(0)}>}) is satisfiable
14       Then Generate((V,A,C+{<(v_p),{(0)}>}),Cons(0,M),p-1)
15       If (V,A,C+{<(v_p),{(1)}>}) is satisfiable
16       Then Generate((V,A,C+{<(v_p),{(1)}>}),Cons(1,M),p-1)
17       If (V,A,C+{<(v_p),{(2)}>}) is satisfiable
18       Then Generate((V,A,C+{<(v_p),{(2)}>}),Cons(2,M),p-1)
19    End
20 End
```

This algorithm obviously enumerates every solution of the given instance exactly one time.

Lemma 16 *If Γ is a tractable constraint language over $\{0, 1, 2\}$ and $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ is a 3-element algebra there is an enumeration algorithm with polynomial delay for each problem instance over Γ.*

Proof Let the problem instance $\mathcal{P} = (V, A, \mathcal{C})$ over Γ with $V = \{v_1, \ldots, v_n\}$ be the input. Since $\Gamma \subseteq \Gamma^+$, it holds that Γ is tractable and therefore line 4 takes only polynomial time. If $\mathcal{P}$ is not satisfiable the algorithm stops in line 6 without calling of the Procedure Generate. So let us assume $\mathcal{P}$ is satisfiable. We set $\Gamma_a = \Gamma \cup \{\{(a)\}\}$ for any $a \in \{0,1,2\}$.

Generate calls itself recursively up to a recursion depth of $n+1$. In every recursion an extended instance is given to the next. The extension is always one constraint with a unary relation. Therefore an instance that is given in a recursion of depth $n+1$ is only $n \log n$ longer than the original instance, its size is polynomial in the input size.

Since Γ^+ is tractable, it holds that Γ_a is tractable for any $a \in \{0,1,2\}$ and in every recursion lines 13, 15, and 17 take polynomial time in the size of the extended instance, that is polynomial time in the input size of the original instance.

The first output is given in the $(n+1)$th recursive call of Generate. Each of this calls tries at most three times if an extended instance is satisfiable (this takes only polynomial time) until calling the next instance of Generate. So the first solution is printed after a delay bounded by a polynomial in the input size.

Between two outputs, the algorithm goes at most n times back to the previous call of Generate and needs at most n new recursive calls. The going back to the previous call is also done in polynomial time because it is tested at most two times if an extended instance is satisfiable. Hence, the time needed is polynomial.

After the last output the algorithm only needs to finish $n+1$ recursions to stop. This is done also in polynomial time in the input size.

Thus the algorithm has polynomial delay. □

Notice that for a tractable constraint language Γ over $\{0,1,2\}$ such that $\mathbb{A}_{\Gamma^{\text{id}}}$ has less than three elements, we do not know if $\Gamma_a = \Gamma \cup \{\{(a)\}\}$ is tractable for every $a \in \{0,1,2\}$. Therefore in general the given enumeration algorithm does not have polynomial delay for such a constraint language.

Let $\mathcal{P} = (V, A, \mathcal{C})$ be a problem instance over Γ. We can enumerate all solutions that use only values from $\mathbb{A}_{\Gamma^{\text{id}}}$ with polynomial delay. In the case that $\mathbb{A}_{\Gamma^{\text{id}}}$ has only one element, this is done by giving the constant solution φ with $\varphi(v) = c$ for any $v \in V$ and c the element of $\mathbb{A}_{\Gamma^{\text{id}}}$. In the case that $\mathbb{A}_{\Gamma^{\text{id}}}$ is a 2-element algebra, Γ^{id} is a tractable constraint language and therefore Schaefer. So there is a $f \in \text{Pol}\,\Gamma$ such that $f \circ f = f$ and $\Gamma^{\text{id}} = f(\Gamma)^+$ and we can apply the original algorithm from [CH97] on the instance $\mathcal{P}' = (V, f(A), \mathcal{C}')$, where $\mathcal{C}' = \{\langle s, f(R)\rangle \mid \langle s, R\rangle \in \mathcal{C}\}$, to generate all such solutions.

However, to every solution φ of $\mathcal{P}'$ there may exist an exponential number of solutions ψ of $\mathcal{P}$ such that $f \circ \psi = \varphi$. So there is no straight way to enumerate all solutions of $\mathcal{P}$ with polynomial delay.

Cohen presents the result of Lemma 16 for the general case of constraint languages over a finite domain.

Theorem 17 ([Coh04]) *Let Γ be a tractable constraint language over a finite domain A*

with $|A| = k$*. Then it holds that there is an enumeration algorithm with polynomial delay for CSP(*Γ*) if* $\mathbb{A}_{\Gamma^{\text{id}}}$ *is a* k*-element algebra.*

References

[BCRV03] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.

[BJ00] A. Bulatov and P. Jeavons. Tractable constraints closed under a binary operation. Technical Report PRG-TR-12-00, Computing Laboratory, University of Oxford, UK, 2000.

[BJK00] A. Bulatov, P. G. Jeavons, and A. A. Krokhin. Constraint satisfaction problems and finite algebras. In *Proceedings 27th International Colloqium on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 272–282, Berlin Heidelberg, 2000. Springer Verlag.

[Bul02a] A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings 43rd Symposium on Foundations of Computer Science*, pages 649–658. IEEE Computer Society Press, 2002.

[Bul02b] A. Bulatov. Mal'tsev constraints are tractable. Technical Report 02-034, Electronic Colloqium on Computational Complexity, 2002.

[Bul02c] A. Bulatov. Tractable constraint satisfaction problems on a 3-element set. Technical Report 02-032, Electronic Colloquium on Computational Complexity, 2002.

[CH97] N. Creignou and J.-J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 31(6):499–511, 1997.

[Coh04] D. A. Cohen. Tractable decision for a constraint language implies tractable search. *Constraints*, 9(3):219–229, 2004.

[Dal00] V. Dalmau. *Computational complexity of problems over generalized formulas.* PhD thesis, Department de Llenguatges i Sistemes Informàtica, Universitat Politécnica de Catalunya, 2000.

[FV98] T. Feder and M. Y. Vardi. The computational structure of monotone monadis SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[JCC98] P. G. Jeavons, D. Cohen, and M. C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101:251–265, 1998.

[JCG97] P. G. Jeavons, D. A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[Jea98] P. G. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.