The Meta-Problem for Conservative Mal'tsev Constraints

Clément Carbonnel

The Meta-Problem for Conservative Mal'tsev Constraints

Clément Carbonnel*
LAAS-CNRS
University of Toulouse, INP Toulouse, France
carbonnel@laas.fr

Abstract

In the algebraic approach to CSP (Constraint Satisfaction Problem), the complexity of constraint languages is studied using closure operations called polymorphisms. Many of these operations are known to induce tractability of any language they preserve. We focus on the meta-problem: given a language Γ, decide if Γ has a polymorphism with nice properties. We design an algorithm that decides in polynomial-time if a constraint language has a conservative Mal'tsev polymorphism, and outputs one if one exists. As a corollary we obtain that the class of conservative Mal'tsev constraints is uniformly tractable, and we conjecture that this result remains true in the non-conservative case.

1 Introduction

The complexity of constraint satisfaction problems is a very active and fruitful research area. In particular, the study of CSP over fixed constraint languages has attracted considerable interest since it was conjectured that for every finite constraint language Γ, CSP(Γ) is either in P or NP-hard (the Feder-Vardi Dichotomy Conjecture) (Feder and Vardi 1998). The most remarkable achievements to date include a characterization of languages that can be solved by local consistency methods (Barto and Kozik 2014) or Gaussian-like algorithms (Idziak et al. 2007), and a proof of the Dichotomy Conjecture for conservative languages (languages with all possible unary relations over the domain) (Bulatov 2003). These results use the algebraic approach to CSP: every language Γ can be associated with a set of closure operations, called polymorphisms, which have been shown to entirely determine the complexity of CSP(Γ) (Jeavons, Cohen, and Gyssens 1997).

Given an operation $f : \mathcal{D}^k \to D$, a language Γ over the domain $\mathcal{D}$ admits f as a polymorphism if every constraint relation $R \in \Gamma$ is closed under componentwise application of f. For example, the affine relation $x + y + z = c$ is closed under the polymorphism $f(x_1, x_2, x_3) = x_1 - x_2 + x_3$, since $x_1 + y_1 + z_1 = c$, $x_2 + y_2 + z_2 = c$, $x_3 + y_3 + z_3 = c$ imply that $f(x_1, x_2, x_3) + f(y_1, y_2, y_3) + f(x_3, y_3, z_3) = (x_1 - x_2 + x_3) + (y_1 - y_2 + y_3) + (z_1 - z_2 + z_3) = c$. A number of sufficient conditions for tractability have been identified this way; for instance, CSP(Γ) is solved by enforcing generalized arc-consistency (GAC) if Γ has a semilattice polymorphism (Jeavons, Cohen, and Gyssens 1997). Each sufficient condition defines a *tractable class*, that is, a set T of languages such that $\forall \Gamma \in \mathrm{T}$, CSP($\Gamma$) is in P.

There are some desirable properties that good tractable classes can be expected to have. First, we know that there exists a polynomial-time algorithm for each *fixed* $\Gamma \in \mathrm{T}$, but there is no guarantee that there exists one polynomial-time algorithm that solves *every* CSP(Γ), $\Gamma \in \mathrm{T}$. This can be formalized as a promise problem: if CSP(T) is CSP together with the promise that the instance is over a language in T, is it true that CSP(T) $\in$ P? If the answer is yes, we say that T is *uniformly tractable* (or equivalently that T *uniformizes* (Kolaitis and Vardi 2000)).

We shall illustrate this notion with an example. Consider the tractable class T_c of all languages Γ such that CSP(Γ) can be solved by enforcing strong k-consistency, where k only depends on Γ. Since there is no bound on k in the definition of T_c, it is not clear that T_c is uniformly tractable. However, a powerful result by Bulatov implies that enforcing a form of consistency called $(2, 3)$-minimality suffices to solve CSP(Γ) for each $\Gamma \in \mathrm{T}_c$ (Bulatov 2010). Enforcing $(2, 3)$-minimality is polynomial-time, so T_c is uniformly tractable.

Even if the class is uniformly tractable, one problem remains: how hard is it to decide if a given language Γ is in T? This is the *meta-problem* for T. In its full generality, the meta-problem has no restriction on the input language. In particular, the domain size is not assumed to be bounded. In the worst case the meta-problem is not necessarily decidable, but in practice it is often in NP. If the class is defined by the existence of polymorphisms satisfying a certain set of identities (which is usually the case), the meta-problem is a *polymorphism detection problem*. For instance, the class of languages that admit a semilattice polymorphism is uniformly tractable since it is solved by GAC, but the meta-problem is NP-complete (Green and Cohen 2008).

Beyond pure academic interest, the main reason for investigating the complexity of meta-problems concerns

*supported by ANR Project ANR-10-BLAN-0210.

general-purpose solvers. It is great to know that languages with a nice polymorphism can be solved efficiently, but this information is virtually useless for practical constraint solvers if they cannot decide quickly if the language of the instance they are trying to solve has the desired polymorphism. Furthermore, it was observed that constraint solvers may perform poorly even on instances that are theoretically very easy (Petke and Jeavons 2009), which suggests that spending some time analyzing the instance before starting search could be beneficial. Beyond preprocessing uses, a very efficient detection algorithm could be exploited in the framework of backdoors, which aims to provide performance improvements even if only a fraction of the constraints have a nice polymorphism (Williams, Gomes, and Selman 2003). In this setting, conservative polymorphisms are of special interest (Bessiere et al. 2013).

Sometimes, the complexity of the meta-problem is strongly related to the uniform tractability question. This is true for the tractable class T_{Mal} of all languages that admit a Mal'tsev polymorphism, which include as particular cases the languages whose relations are linear equations over a field (Bulatov 2002). The solution algorithm resembles Gaussian elimination, in that it starts from an instance without any constraint and then adds the constraints one by one while maintaining at all times a polynomial-sized representation of the solution set (Bulatov and Dalmau 2006)(Dyer and Richerby 2013). This algorithm remains polynomial time even if the domain size or the number of tuples are not fixed, but it does not entail uniform tractability because *it assumes that the Mal'tsev polymorphism is known*. Since there are roughly d^{d^3} possible Mal'tsev operations over a domain of size d and it is possible that only one of them is a polymorphism of the language, an exhaustive approach is not satisfying. However, should a polynomial-time algorithm that outputs a Mal'tsev polymorphism if one exists be engineered, we could interface it with the state-of-the-art solution algorithm and prove uniform tractability of T_{Mal}. But then, this polymorphism detection algorithm would also prove that the meta-problem is in P.

This paper builds around the observation that T_{Mal} is likely to have an easy meta-problem and be uniformly tractable. Although we cannot prove this claim in its full generality, we present a proof for the restricted case of *conservative* Mal'tsev polymorphisms. This extends previous results showing that conservative Mal'tsev polymorphisms can be detected in polynomial time in digraphs (Carvalho et al. 2011) and binary relational structures (Bessiere et al. 2013). As a byproduct, we obtain a greatly improved algorithm for detecting conservative *majority* polymorphisms, which generalise 2SAT and connected row-convex constraints.

Besides being a first step towards proving the uniform tractability of Mal'tsev constraints, our result for the conservative case is interesting in its own right. The tractable class of languages having a conservative Mal'tsev polymorphism has seen little pratical use, but is of great theoretical importance. For instance, conservative Mal'tsev polymorphisms are one of the main ingredients in Libor Barto's proof of the conservative Dichotomy Conjecture (Barto 2011). Moreover, the existence of a conservative Mal'tsev polymorphism is a necessary condition for the tractability of CCSP(Γ), a variant of CSP(Γ) in which global cardinality constraints are allowed in addition to the relations of Γ (Bulatov and Marx 2010). Examples of conservative Mal'tsev operations include *extreme value functions*, which map any triplet $\{x, y, z\}$ of natural numbers to $\alpha \in \{x, y, z\}$ such that $|\alpha - \mathrm{median}(x, y, z)|$ is maximum.

2 Preliminaries

CSP. A Constraint Satisfaction Problem (CSP) is a triple $(\mathcal{X},\mathcal{D},\mathcal{C})$ where $\mathcal{X}$ is a set of variables, $\mathcal{D}$ is a finite set of values, and $\mathcal{C}$ is a set of constraints. A constraint C of arity r is a pair $(\mathcal{S}(C), \mathcal{R}(C))$ where $\mathcal{S}(C) \in \mathcal{X}^r$ is the *scope* of C and $\mathcal{R}(C) \subseteq D^r$ is the *relation* of C. Note that $\mathcal{R}(.)$ and $\mathcal{S}(.)$ can be seen as operators that return the relation and scope of a constraint. A solution of I is an assignment $\phi : \mathcal{X} \to \mathcal{D}$ such that $\forall C \in \mathcal{C}$, $\phi(\mathcal{S}(C)) \in \mathcal{R}(C)$, and the goal is to decide if I has a solution. A *constraint language* is a set of relations, and the *language* of a CSP instance I is the set $\Gamma_I = \{\mathcal{R}(C) \mid C \in \mathcal{C}\}$. Given a fixed constraint language Γ, CSP(Γ) is the set of all instances I such that $\Gamma_I \subseteq \Gamma$. We assume that all relations are given in extension (i.e. as lists of tuples).

Polymorphisms. An operation $f : \mathcal{D} \to \mathcal{D}^k$ is a *polymorphism* of a language Γ over $\mathcal{D}$ if for all $R \in \Gamma$ of arity r and $\mathbf{t_1}, \ldots, \mathbf{t_k} \in R$, $< f(\mathbf{t_1}[1], \ldots, \mathbf{t_k}[1]), \ldots, f(\mathbf{t_1}[r], \ldots, \mathbf{t_k}[r]) > \in R$. The set of all polymorphisms of Γ is denoted by Pol(Γ) and constitutes an *operational clone*, that is, a set of operations closed under composition that contains all projections (Jeavons, Cohen, and Gyssens 1997). It has been shown that the complexity of CSP(Γ) is entirely determined by Pol(Γ) (Jeavons, Cohen, and Gyssens 1997). An operation $f : \mathcal{D} \to \mathcal{D}^k$ is *conservative* if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for all $x_1, \ldots, x_k \in \mathcal{D}$, *Mal'tsev* if it is ternary and $\forall x, y \in \mathcal{D}$, $f(x, x, y) = f(y, x, x) = y$, and *majority* if it is ternary and $\forall x, y \in \mathcal{D}$, $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$.

Tools. The most useful tool used to design polymorphism detection algorithms is the *indicator problem*. Formally, given an integer k and a finite constraint language Γ, the indicator problem of order k of Γ is a CSP instance $IP^k(\Gamma)$ with one variable $x_{v_1,\ldots,v_k}$ for every k-tuple $(v_1, \ldots, v_k)$ of elements from $\mathcal{D}$. Then, for each $R \in \Gamma$ of arity r and $t_1, \ldots, t_k \in R$, $IP^k(\Gamma)$ contains a constraint $C^R_{t_1,\ldots,t_k}$ with scope $(x_{t_1[1],\ldots t_k[1]}, \ldots, x_{t_1[r],\ldots,t_k[r]})$ and relation R. Going back to the definition of a polymorphism, it follows that an operation f of arity k is a polymorphism of Γ if and only if $x_{v_1,\ldots,v_k} \leftarrow f(v_1, \ldots, v_k)$ is a solution of $IP^k(\Gamma)$.

If we are only looking for polymorphisms with special properties, sometimes the solution set of $IP^k(\Gamma)$ can be restricted to exactly those polymorphisms. For in-

stance, if $k = 3$ adding the unary constraints $x_{v_1,v_1,v_2} \in \{v_1\}$, $x_{v_1,v_2,v_1} \in \{v_1\}$, $x_{v_2,v_1,v_1} \in \{v_1\}$ for every $v_1, v_2 \in \mathcal{D}$ will ensure that every solution of this modified indicator problem $IP^{maj}(\Gamma)$ is a majority polymorphism. This type of restriction is sometimes not possible without increasing exponentially the size of the indicator problem (e.g. semilattices). It was first observed in (Feder and Vardi 1998) that the language of $IP^{maj}(\Gamma)$ is Γ plus unary relations with a single tuple, and thus has a majority polymorphism if and only if $IP^{maj}(\Gamma)$ has a solution. Furthermore, by the properties of majority polymorphisms it follows that if $IP^{maj}(\Gamma)$ has a solution, then it can be solved backtrack-free by applying singleton arc-consistency at each node of the search tree (Chen, Dalmau, and Grußien 2013). The standard algorithm for detecting majority polymorphisms starts by building $IP^{maj}(\Gamma)$ and then solves it by a standard search, maintaining SAC at each node. If the algorithm backtracks, we can conclude that Γ has no majority polymorphism, so the whole procedure is polynomial-time. This approach has been used for several other tractable classes (Barto 2015)(Feder and Vardi 1998). However, this kind of detection algorithm requires the existence of a uniform algorithm for the tractable class, which is not known for several polymorphisms including (conservative) Mal'tsev polymorphisms. In this paper we introduce a different technique, based on a detailed analysis of the structure of the indicator problem.

3 First observation

Recall that the existence of a uniform algorithm for Mal'tsev constraints is equivalent to the tractability of the problem $\mathrm{CSP}(\mathrm{T}_{Mal})$, where we only have the promise that the language of the instance has a Mal'tsev polymorphism. The complexity of this problem is open, but is it easy to see that the following is true.

Observation 1. *$CSP(T_{Mal}) \in NP \cap coNP$.*

Proof. Membership in NP follows from that of the general CSP. For membership in coNP, a Mal'tsev polymorphism f of the constraint language is a certificate: with the knowledge of f, the algorithm from (Dyer and Richerby 2013) provides a way to check in polynomial time that the instance has no solution. □

Unless NP = coNP, this observation rules out the possibility that this problem is NP-hard (Goldreich 2010). Besides, examples of NP ∩ coNP problems that are not believed to be in P are quite rare, so we regard this observation as evidence that Mal'tsev constraints may have a uniform algorithm. Using the same kind of reasoning as for majority polymorphisms, it would follow that Mal'tsev polymorphisms can be detected in polynomial time. The next section will provide additional evidence by proving that *conservative* Mal'tsev constraints are uniformly tractable.

We note that this observation also applies to the larger tractable class of languages having a k-edge polymorphism (for a fixed k) by using the algorithm of (Idziak et al. 2007) for membership in coNP. However, for the sake of simplicity we shall focus on the case $k = 2$, which corresponds to Mal'tsev polymorphisms.

4 Conservative Mal'tsev constraints

In this section, we show that the existence of a conservative Mal'tsev polymorphism can be decided in polynomial time. The outline of the proof is as follows. We first reduce the problem to that of finding a conservative *minority* polymorphism (i.e. a ternary polymorphism m such that $\forall x, y,\ m(x, x, y) = m(x, y, x) = m(y, x, x) = y$). Then, we show that enforcing arc-consistency on the indicator problem associated with conservative minority polymorphisms leaves an extremely well-structured instance, and a simple reduction rule allows us to eliminate every variable whose domain contains more than two values. The residual instance is then shown to be equivalent to a system of linear equations over $GF(2)$, and can be solved by Gaussian elimination.

Lemma 1. *Let $\mathcal{F}$ be an operational clone. $\mathcal{F}$ contains a conservative Mal'tsev operation if and only if it contains a conservative minority operation.*

Proof. Every minority operation is a Mal'tsev operation, hence one implication is trivial. Suppose that $\mathcal{F}$ contains a conservative Mal'tsev operation m, and let

$$f(x, y, z) = m(z, m(y, m(x, z, y), x), m(x, z, y))$$

This operation belongs to $\mathcal{F}$ because $\mathcal{F}$ is a clone, and is conservative since m is conservative. Furthermore, for every a, b we have

$$f(a, b, a) = m(a, m(b, m(a, a, b), a), m(a, a, b)) = b$$
$$f(b, a, a) = m(a, m(a, m(b, a, a), b), m(b, a, a)) = b$$

and it is fairly straightforward to see that $f(a, a, b) = m(b, m(a, m(a, b, a), a), m(a, b, a))$ is always equal to b, whether $m(a, b, a) = b$ or $m(a, b, a) = a$. Hence, f is a minority operation of $\mathcal{F}$. □

Although this lemma may be known to some, it appears to have never been pointed out in the literature. The closest results we could find were that digraphs with a conservative Mal'tsev polymorphisms also have a conservative minority polymorphism (Carvalho et al. 2011) and constraint languages with both a conservative majority and a conservative Mal'tsev polymorphism also have a conservative minority polymorphism (Bulatov and Marx 2010). In our case, this lemma is crucial, since the indicator problem corresponding to conservative minority polymorphisms has interesting (i.e., algorithmically exploitable) properties that its counterpart for Mal'tsev polymorphisms does not have.

Given a language Γ, we denote by $IP^{cmin}(\Gamma)$ the indicator problem of order 3 of Γ with the additional constraints $x_{v_1,v_1,v_2} \in \{v_2\}$, $x_{v_1,v_2,v_1} \in \{v_2\}$, $x_{v_2,v_1,v_1} \in$

$\{v_2\}$ for every $v_1, v_2 \in \mathcal{D}$ and $x_{v_1,v_2,v_3} \in \{v_1, v_2, v_3\}$ for every $v_1, v_2, v_3 \in \mathcal{D}$. By construction, the solutions of $IP^{cmin}(\Gamma)$ are exactly the conservative minority polymorphisms of Γ. Given a constraint $C = (S, R)$ and $S' \subseteq S$, we denote by $C[S']$ the projection of C onto S'. For our structural analysis we will assume that for every $R^* \in \Gamma$, $IP^{cmin}(\Gamma)$ also contains a constraint $C^{R'}_{\mathbf{t'_1},\mathbf{t'_2},\mathbf{t'_3}}$ for every projection R' of R^* and $\mathbf{t'_1}, \mathbf{t'_2}, \mathbf{t'_3} \in R'$. These additional constraints are only needed to facilitate our analysis and will not be required by the algorithm.

In a generalized arc-consistent instance, the domain $D(x)$ of a variable x is the set of values for x that have supports in every constraint whose scope contains x. For the remainder of the paper, we will assume that GAC has been enforced on $IP^{cmin}(\Gamma)$. The following observation describes an immediate but very important property that will be used repeatedly in our proofs.

Observation 2. *If $C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}} = (R, S)$ is a constraint in $IP^{cmin}(\Gamma)$ and $\mathbf{t}, \mathbf{t}', \mathbf{t}'' \in R$, then $\mathcal{R}(C^{R^*}_{\mathbf{t},\mathbf{t}',\mathbf{t}''}) \subseteq R$.*

Proof. Let $C^{R^*}_{\mathbf{t},\mathbf{t}',\mathbf{t}''} = (R', S')$, and $|S| = |S'| = r$. Before GAC was enforced, both $C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}}$ and $C^{R^*}_{\mathbf{t},\mathbf{t}',\mathbf{t}''}$ had R^* as relation. Thus, by definition of generalized arc-consistency, we have $R = R^* \cap (\pi_{x \in S} D(x))$ and $R' = R^* \cap (\pi_{x \in S'} D(x))$. However, since $\mathbf{t}, \mathbf{t}', \mathbf{t}'' \in R$, the conservativity constraints ensure that for each $i = 1..r$, $D(S'[i]) \subseteq D(S[i])$. Therefore, $R' \subseteq R$. □

Throughout the paper we will treat elements of a scope S as *occurences* of variables, and not simply variables. For example, given $x \in S$, the restricted scope $S \backslash x$ removes the occurence x from S, but not *every* occurence of the variable represented by x. A constraint $C = (S, R)$ is *functional* in $x \in S$ if for every valid assignment t of $S \backslash x$ there is at most one value $d \in \mathcal{D}$ such that $(S \backslash x \leftarrow t, x \leftarrow d)$ is an assignment to S that satisfies C. Finally, if two relations R and R' differ only by a permutation of their columns, we write $R \approx R'$. The proof of the next lemma gives a simple example of the use we will make of Observation 2.

We remind the reader that if $C = C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}}$ is a constraint of $IP^{cmin}(\Gamma)$, the kth variable in its scope is $x_{\mathbf{t_1}[k],\mathbf{t_2}[k],\mathbf{t_3}[k]}$. Therefore, if $\mathbf{t_1}[k] = \mathbf{t_2}[k]$, the unary constraints will ensure that $x_{\mathbf{t_1}[k],\mathbf{t_2}[k],\mathbf{t_3}[k]}$ is ground (i.e. has a singleton domain) with value $\mathbf{t_3}[k]$.

Lemma 2. *Let $C = (R, S)$ be a constraint in $IP^{cmin}(\Gamma)$, and let $x \in S$. Either C is functional in x, or $R \approx \mathcal{R}(C[S \backslash x]) \times D(x)$.*

Proof. Let $C = C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}}$ and $x = x_{v_1,v_2,v_3}$. Without loss of generality, we assume that x occurs last in S. First, suppose that there exists $\mathbf{t} \in \mathcal{R}(C[S \backslash x])$ such that $(\mathbf{t}, v_k) \in \mathcal{R}(C)$ for every $v_k \in D(x)$. We will show that every tuple must have the same property as t. Let $\mathbf{t}' \in \mathcal{R}(C[S \backslash x])$ be such that $(\mathbf{t}', v_\alpha) \in \mathcal{R}(C)$ but $(\mathbf{t}', v_\beta) \notin \mathcal{R}(C)$ for some $\{v_\alpha, v_\beta\} \subseteq D(x)$. Then, because of the unary constraints, the constraint $C^{R^*}_{(\mathbf{t},v_\alpha),(\mathbf{t},v_\beta),(\mathbf{t}',v_\alpha)}$ has only ground variables in its scope, and its only possible support is $(\mathbf{t}', v_\beta)$. By Observation 2, $\mathcal{R}(C^{R^*}_{(\mathbf{t},v_\alpha),(\mathbf{t},v_\beta),(\mathbf{t}',v_\alpha)}) \subseteq \mathcal{R}(C)$ and hence $(\mathbf{t}', v_\beta) \in \mathcal{R}(C)$, a contradiction. Therefore, such a partial tuple $\mathbf{t}'$ cannot exist and $R \approx \mathcal{R}(C[S \backslash x]) \times D(x)$.

Now, suppose that $D(x) = \{v_1, v_2, v_3\}$ and there exists $\mathbf{t} \in \mathcal{R}(C[S \backslash x])$ such that $(\mathbf{t}, v_k) \in \mathcal{R}(C)$ for exactly two indices k, say 1 and 2. Since C is arc-consistent, there exists $\mathbf{t}'$ such that $(\mathbf{t}', v_3) \in \mathcal{R}(C)$. However, the scope of constraint $C^{R^*}_{(\mathbf{t},v_1),(\mathbf{t},v_2),(\mathbf{t}',v_3)}$ contains only ground variables and x; therefore $\mathcal{R}(C^{R^*}_{(\mathbf{t},v_1),(\mathbf{t},v_2),(\mathbf{t}',v_3)})$ contains the tuple $(\mathbf{t}', v_k)$ for all $k \in \{1, 2, 3\}$. By Observation 2 we have $\mathcal{R}(C^{R^*}_{(\mathbf{t},v_1),(\mathbf{t},v_2),(\mathbf{t}',v_3)}) \subseteq \mathcal{R}(C)$, and the partial tuple $\mathbf{t}'$ brings us back to the first case.

If no tuples satisfy either of the above two conditions, C is functional in x. □

The key observation in our proof will be that variables with domain size 1 or 2 have very limited interactions with variables with domain size 3 once arc-consistency has been enforced. Given a constraint C in $IP^{cmin}(\Gamma)$, we denote by $\mathcal{S}_{|1,2}(C)$ the restriction of S to variables with domain size 1 or 2, and by $\mathcal{S}_{|3}(C)$ the restriction of S to variables with domain size 3.

Lemma 3. *Let C be a constraint in $IP^{cmin}(\Gamma)$ and $x \in \mathcal{S}_{|3}(C)$. $\mathcal{R}(C[\mathcal{S}_{|1,2}(C) \cup x]) \approx \mathcal{R}(C[\mathcal{S}_{|1,2}(C)]) \times D(x)$.*

Proof. Let $C_1 = C[\mathcal{S}_{|1,2}(C)] = (R_1, S_1)$, $C_2 = C[\mathcal{S}_{|1,2}(C) \cup x] = (R_2, S_2)$ and assume that $x = x_{v_1,v_2,v_3}$ occurs last in the scope of C_2. By Lemma 2, either $R_2 = R_1 \times D(x)$ or C_2 is functional in x. If it is functional, then by GAC there exist $\mathbf{t}, \mathbf{t}', \mathbf{t}'' \in R_1$ such that R_2 contains $(\mathbf{t}, v_1)$, $(\mathbf{t}', v_2)$ and $(\mathbf{t}'', v_3)$. Then, the scope of $C' = C^{R^*}_{(\mathbf{t},v_1),(\mathbf{t}',v_2),(\mathbf{t}'',v_3)}$ has only ground variables (those corresponding to $\mathcal{S}_{|1,2}(C)$) plus x_{v_1,v_2,v_3}. Therefore, there exists $\mathbf{t}^*$ such that $\mathcal{R}(C')$ contains $(\mathbf{t}^*, v_1)$, $(\mathbf{t}^*, v_2)$ and $(\mathbf{t}^*, v_3)$. By Observation 2, $\mathcal{R}(C') \subseteq R_2$ and C_2 is not functional in x, a contradiction. □

Lemma 3 only deals with constraints whose scope contains exactly *one* variable with domain size 3. Unfortunately, for k variables it is not completely true that $\mathcal{R}(C[\mathcal{S}_{|1,2}(C) \cup \{x_1, \ldots, x_k\}]) \approx \mathcal{R}(C[\mathcal{S}_{|1,2}(C)]) \times D(x_1) \times \ldots \times D(x_k)$. Let $x_{v_1^1,v_2^1,v_3^1}, \ldots, x_{v_1^k,v_2^k,v_3^k}$ be k variables of the indicator problem. The *index-equality* constraint between these variables has three satisfying assignments: $(v_1^1, \ldots, v_1^k), (v_2^1, \ldots, v_2^k)$ and $(v_3^1, \ldots, v_3^k)$. The next Proposition is the keystone of our proof, and gives the correct generalization of Lemma 3 to an arbitrary number of variables with domain size 3.

Proposition 1. *Let C be a constraint in $IP^{cmin}(\Gamma)$. There exists $n \geq 0$ and a set of constraints $C^*, C_1, \ldots, C_n$ such that*

$$C = C^* \wedge \left(\bigwedge_{i=1..n} C_i \right)$$

where the scope of C^* is $\mathcal{S}(C)$, the constraints C_i are (possibly unary) index-equalities whose scope are disjoint and cover $\mathcal{S}_{|3}(C)$, and

$$\mathcal{R}(C^*) \approx \mathcal{R}(C[\mathcal{S}_{|1,2}(C)]) \times \Pi_{x \in \mathcal{S}_{|3}(C)} D(x)$$

Proof. We proceed by induction on the size of $\mathcal{S}_{|3}(C)$. Let $k > 0$ and suppose that Proposition 1 is true for all constraints C' such that $|\mathcal{S}_{|3}(C')| \leq k$. Let $C = C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}} = (S, R)$ be a constraint with $|\mathcal{S}_{|3}(C)| = k+1$, and $x \in \mathcal{S}_{|3}(C)$. By Lemma 2, either C is functional in x or $\mathcal{R}(C) = \mathcal{R}(C[S\backslash x]) \times D(x)$. In the latter case, C satisfies Proposition 1 by induction. Therefore, we shall assume that C is functional in x.

By induction, we know that $C[S\backslash x] = C^* \bigwedge_{i=1..n} C_i$. Let $y \in \{1..n\}$ and $Y = \mathcal{S}(C_y)$. Let $\mathbf{v_i}$, $i = 1, 2, 3$ be the three possible assignments to Y. We assume without loss of generality that $x = x_{u_1,u_2,u_3}$ (hence, $D(x) = \{u_1, u_2, u_3\}$) and (Y, x) are the last variables in S. Let $\mathbf{t} \in \mathcal{R}(C[S\backslash\{Y, x\}])$, and define $\phi_\mathbf{t} : D(Y) \to D(x)$ such that $\phi_\mathbf{t}(\mathbf{v}) = \{u \in D(x) \,|\, (\mathbf{t}, \mathbf{v}, u) \in \mathcal{R}(C)\}$. We distinguish three cases.

1. $\phi_\mathbf{t}$ has range $\{u_i, u_j\}$ for some $i \neq j$. One of these two values, say u_i, has a preimage of size 2. Let $\{\mathbf{v_p}, \mathbf{v_s}\} = \phi_\mathbf{t}^{-1}(\{u_i\})$, $\mathbf{v_l} \notin \{\mathbf{v_p}, \mathbf{v_s}\}$, and $\mathbf{t^y_1}, \mathbf{t^y_2}, \mathbf{t^y_3}$ be the permutation of $(\mathbf{t}, \mathbf{v_p}, u_i), (\mathbf{t}, \mathbf{v_s}, u_i), (\mathbf{t}, \mathbf{v_l}, u_j)$ such that $\mathbf{t^y_h}[Y] = \mathbf{v_h}$. The constraint $C^{R^*}_{\mathbf{t^y_1},\mathbf{t^y_2},\mathbf{t^y_3}}$ has only the variables in Y as active variables, and by arc consistency its relation must contain $(\mathbf{t}, \mathbf{v_p}, u_j), (\mathbf{t}, \mathbf{v_s}, u_j), (\mathbf{t}, \mathbf{v_l}, u_j)$. By Observation 2, R must contain these tuples, a contradiction.
2. $\phi_\mathbf{t}$ is bijective. Suppose that there exist i, j such that $i \neq j$ and $\phi_\mathbf{t}(\mathbf{v_i}) = u_j$. Let $u_s \notin \{u_j, u_i\}$, $\mathbf{t'} = (\mathbf{t}, \mathbf{v_i}, u_j)$ and $\mathbf{t''} = (\mathbf{t}, \phi_\mathbf{t}^{-1}(u_s), u_s)$. Let $\mathbf{t^x_1}, \mathbf{t^x_2}, \mathbf{t^x_3}$ be the permutation of the tuples $\mathbf{t_i}, \mathbf{t'}, \mathbf{t''}$ such that $\mathbf{t^x_h}[x] = u_h$. Recall that $\mathbf{t_i}$ is one of the three tuples associated with the constraint $C = C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}}$, and hence $\mathbf{t_i} \in R^*$, $\mathbf{t_i}[Y] = \mathbf{v_i}$ and $\mathbf{t_i}[x] = u_i$. Then, the constraint $C^{R^*}_{\mathbf{t^x_1},\mathbf{t^x_2},\mathbf{t^x_3}}$ has x as the only active variable in its scope, and for every $u \in D(x)$ its relation must contain the tuple $\mathbf{t^u}$ such that $\mathbf{t^u}[l] = \mathbf{t_i}[l]$ if $l \notin Y \cup \{x\}$, $\mathbf{t^u}[Y] = \phi^{-1}(u_s)$, and $\mathbf{t^u}[x] = u$. Note that at this point, Observation 2 cannot be applied because $\mathbf{t_i}$ may not belong to $\mathcal{R}(C)$. Let $\mathbf{t^a_1}, \mathbf{t^a_2}, \mathbf{t^a_3}$ be the permutation of $\mathbf{t_i}, \mathbf{t^s}, \mathbf{t'}$ such that $\mathbf{t^a_h}[x] = u_h$. The constraint $C^{R^*}_{\mathbf{t^a_1},\mathbf{t^a_2},\mathbf{t^a_3}}$ has only ground variables in its scope except x, and its relation R' must contain the tuple $\mathbf{t^f}$ such that $\mathbf{t^f}[x] = u_j$ and $\mathbf{t^f}[l] = \mathbf{t''}[l]$ otherwise. However, since $R' \subseteq R$ we have $t_f \in R$, a contradiction. Therefore, if $\phi_\mathbf{t}$ is bijective then it must map every $\mathbf{v_i}$ to u_i.
   Now, suppose that there exists a partial tuple $\mathbf{t'}$ such that $\phi_{\mathbf{t'}}$ is not equal to $\phi_\mathbf{t}$. By Case 1 and the reasoning above, $\phi_{\mathbf{t'}}$ must map every $\mathbf{v_i}$ to the same value u_p. Let $\{\mathbf{v_i}, \mathbf{v_j}\} = D(Y)\backslash\mathbf{v_p}$. If we denote by $\mathbf{t^b_1}, \mathbf{t^b_2}, \mathbf{t^b_3}$ the permutation of $(\mathbf{t'}, \mathbf{v_j}, u_p)$, $(\mathbf{t'}, \mathbf{v_p}, u_p)$ and $(\mathbf{t}, \mathbf{v_i}, u_i)$ such that $\mathbf{t^b_h}[Y] = \mathbf{v_h}$, the constraint $C^{R^*}_{\mathbf{t^b_1},\mathbf{t^b_2},\mathbf{t^b_3}}$ has only the variables in Y as active variables in its scope, and by arc consistency its relation must contain the tuple $(\mathbf{t}, \mathbf{v_p}, u_i)$. By Observation 2, this tuple must belong to R, a contradiction.
   Finally, in this case every tuple must induce an index-equality between Y and x. Therefore, we can add x to the scope of C_y and continue the induction.
3. $\phi_\mathbf{t}$ has range $\{u\}$. By Cases 1 and 2, we know that the only situation where the induction may not hold is when $\phi_{\mathbf{t'}}$ is in this case for every partial tuple $\mathbf{t'}$ and every choice of Y. For each $\mathbf{t'} \in \mathcal{R}(C[S\backslash x])$ and index-equality constrained set of variables Y', we define $\mathcal{J}_{Y'}(\mathbf{t'})$ to be $\mathbf{t'}$ plus the set of all tuples that differ from $\mathbf{t'}$ only on the assignment to Y'. By functionality, for each $\mathbf{t'} \in \mathcal{R}(C[S\backslash x])$ we can define $\psi(\mathbf{t'})$ to be the sole value $u \in D(x)$ such that $(\mathbf{t'}, u) \in R$. It is immediate that $\psi(\mathbf{t}^\alpha) = \psi(\mathbf{t}^\beta)$ for each $\mathbf{t}^\alpha, \mathbf{t}^\beta \in \mathcal{J}_{Y'}(\mathbf{t'})$, for any fixed $Y', \mathbf{t'}$. Furthermore, for any two tuples $\mathbf{t}^\alpha, \mathbf{t}^\beta \in \mathcal{R}(C[S\backslash x])$ such that $\mathbf{t}^\alpha[\mathcal{S}_{|1,2}(C)] = \mathbf{t}^\beta[\mathcal{S}_{|1,2}(C)]$, there exists $\mathbf{t_{Y_1}}, \ldots, \mathbf{t_{Y_n}}$ such that $\mathbf{t_{Y_1}} \in \mathcal{J}_{Y_1}(\mathbf{t}^\alpha)$, $\mathbf{t}^\beta \in \mathcal{J}_{Y_h}(\mathbf{t_{Y_n}})$ and for each i, $\mathbf{t_{Y_{i+1}}} \in \mathcal{J}_{Y_i}(\mathbf{t_{Y_i}})$. Unformally, starting from $\mathbf{t}^\alpha$ one can obtain $\mathbf{t}^\beta$ by changing the assignments to each Y_i one by one. By transitivity of the equality, this means that $\psi(\mathbf{t}^\alpha) = \psi(\mathbf{t}^\beta)$. Since this is true for any pair $\mathbf{t}^\alpha, \mathbf{t}^\beta$ that share the same values for $\mathcal{S}_{|1,2}(C)$, it follows that $C[\mathcal{S}_{|1,2}(C) \cup x]$ is functional in x, a contradiction with Lemma 3.

□

Theorem 1. *There exists an algorithm that decides in polynomial time if a constraint language Γ admits a conservative Mal'tsev polymorphism, and outputs one if one exists.*

Proof. By Lemma 1, we can look for a conservative minority polymorphism instead. The algorithm builds $IP^{cmin}(\Gamma)$ in time $O(rlt^3)$, where l is the number of relations and t, r are respectively the maximum number of tuples and the maximum arity of a relation. $IP^{cmin}(\Gamma)$ has $O(lt^3 + d^3)$ constraints and $O(d^3)$ variables. Then, we enforce GAC in time $O(rlt^4)$. By Proposition 1, assigning every variable x_{v_1,v_2,v_3} with domain size 3 to v_1 does not violate any constraint (since it respects index-equalities) and is consistent with every satisfying assignment to the remaining variables. Therefore, we can eliminate every variable with domain size 3.

We are left with an instance whose active variables have domain size 2, and if it has a solution its language must have a conservative minority polymorphism (conservative polymorphisms are preserved by GAC). Note that all minority operations coincide on 2-elements domains; therefore, we can rename each domain by $\{0, 1\}$ (arbitrarily) and obtain a CSP instance whose language has the unique Boolean minority polymorphism

$m(x, y, z) = x - y + z \bmod 2$. This instance is equivalent to a system of linear equations over GF(2), and any such instance with n variables and m constraints can be solved in time $O(n^2 m)$ by Gaussian elimination. In our case, the running time is $O(rlt^3 d^6)$, and hence the complexity of the whole algorithm is $O(rlt^3 d^6 + rlt^4)$. □

If we interface our detection algorithm with the algorithm of (Dyer and Richerby 2013), we obtain the following corollary.

Corollary. *The class of constraint languages with a conservative Mal'tsev polymorphism is uniformly tractable.*

5 Conservative majority constraints

Unlike conservative Mal'tsev polymorphisms, it is already known that conservative majority polymorphisms can be detected in polynomial time (Feder and Vardi 1998). The state-of-the-art algorithm, described in Section 2, has $O(rd^6lt^4)$ time complexity (Bessiere et al. 2013). In the section, we will show that this algorithm can be greatly improved using the approach we described for conservative Mal'tsev polymorphisms.

As seen in Section 4, analyzing the structure of the indicator problem for languages of large arities can be tedious. Fortunately we need not do this twice, as languages with majority polymorphisms are *2-decomposable*: each constraint can be replaced by its binary projections without altering the solution set of the instance (Jeavons, Cohen, and Cooper 1998).

It is fairly straightforward to see that if a language Γ has a majority polymorphism, then the indicator problem of its 2-decomposition Γ_2 is equivalent to the 2-decomposition of the indicator problem of Γ. We denote by $IP^{cmaj}(\Gamma_2)$ the indicator problem of order 3 of Γ_2 with the additional constraints $x_{v_1,v_1,v_2} \in \{v_1\}$, $x_{v_1,v_2,v_1} \in \{v_1\}$, $x_{v_2,v_1,v_1} \in \{v_1\}$ for every $v_1, v_2 \in \mathcal{D}$ and $x_{v_1,v_2,v_3} \in \{v_1, v_2, v_3\}$ for every $v_1, v_2, v_3 \in \mathcal{D}$. The solutions of $IP^{cmaj}(\Gamma_2)$ are exactly the conservative majority polymorphisms of Γ_2.

Note that Observation 2 can be applied to $IP^{cmaj}(\Gamma_2)$ since its proof only uses conservativity.

Lemma 4. *If $IP^{cmaj}(\Gamma_2)$ is GAC, the assignment*

$$x_{u_1,u_2,u_3} \leftarrow (u_i \in D(x_{u_1,u_2,u_3}) \mid i \text{ is minimum})$$

is a solution.

Proof. We start by considering $IP^{cmaj}(\Gamma_2)$ *before* GAC is applied. Let $C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}} = (S, R^*)$ be a constraint of $IP^{cmaj}(\Gamma_2)$ with scope $(x_{u_1,u_2,u_3}, x_{v_1,v_2,v_3})$ such that both variables are active (i.e. $|\{u_1, u_2, u_3\}| = 3$ and $|\{v_1, v_2, v_3\}| = 3$, as otherwise the unary majority constraints would force the variable to be ground). Suppose that there exists a pair $i \neq j$ such that $\mathbf{t} = (u_i, v_j) \in R$. Let k be the index such that $k \notin \{i, j\}$ and $(\mathbf{t'_1}, \mathbf{t'_2}, \mathbf{t'_3})$ be the permutation of the tuples $\mathbf{t}, \mathbf{t_i}, \mathbf{t_k}$ such that $\mathbf{t'_1}[2] = v_1$, $\mathbf{t'_2}[2] = v_2$ and $\mathbf{t'_3}[2] = v_3$. Consider the constraint $C^{R^*}_{\mathbf{t'_1},\mathbf{t'_2},\mathbf{t'_3}} = (S', R^*)$. The second variable in S' is x_{v_1,v_2,v_3} and after arc-consistency the first variable will be fixed to the value u_i. Therefore, by Observation 2, after arc-consistency the constraint $C^{R^*}_{\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}} = (S, R)$ will contain the tuple (u_i, v) for every $v \in D(x_{v_1,v_2,v_3})$. From this we can deduce that, after arc-consistency, for every i we have either $(u_i, v_i) \in R$ or $(u_i, v) \in R$ for every v in the domain of x_{v_1,v_2,v_3}. In particular, if i and j are the minimum indices such that both u_i and v_i are in the domains, (u_i, v_j) always belongs to R. □

Theorem 2. *Conservative majority polymorphisms can be detected in time $O(rlt^4)$ in constraint languages with l distinct relations of arity at most r and containing at most t tuples.*

Proof. The algorithm starts by *assuming* that a conservative majority polymorphism exists. We build $IP^{cmaj}(\Gamma)$ and enforce GAC in time $O(rlt^4)$. Since $IP^{cmaj}(\Gamma)$ is equivalent to $IP^{cmaj}(\Gamma_2)$, we can use Lemma 4 to find a solution of the resulting instance. If this solution is a majority polymorphism of Γ (which can be verified in time $O(rlt^4)$) the algorithm returns YES; otherwise it returns NO. The complexity of the whole procedure is $O(rlt^4)$. □

This time bound improves on that of (Bessiere et al. 2013) by a factor of d^6. Besides, the time complexity of our algorithm is roughly that of *checking* if a given conservative majority operation is a polymorphism of Γ, so there is little room for improvement.

6 Conclusion

Using a detailed analysis of the indicator problem for conservative minority polymorphisms, we have designed a polynomial-time algorithm for detecting conservative Mal'tsev polymorphisms in arbitrary constraint languages, and obtained as a side result a greatly improved algorithm for detecting conservative majority polymorphisms.

As noted in the introduction, our results imply a uniform algorithm for constraint languages with a conservative Mal'tsev polymorphism. Motivated by Observation 1, we make the following conjecture.

Conjecture 1. *There exists a uniform algorithm for constraint languages with a Mal'tsev polymorphism, and the meta-problem is decidable in polynomial-time.*

The techniques we have developed in this paper make essential use of the fact that we are looking for conservative polymorphisms, and are unlikely to be sufficient to prove Conjecture 1 in its full generality. New ideas are needed, and it may be interesting to see if the algorithm from (Dyer and Richerby 2013) can be uniformized by using a different notion of compact representation of solution sets that only requires the *promise* that a Mal'tsev polymorphism exists.

References

Barto, L., and Kozik, M. 2014. Constraint satisfaction problems solvable by local consistency methods. *J. ACM* 61(1):3.

Barto, L. 2011. The dichotomy for conservative constraint satisfaction problems revisited. In *LICS*, 301–310. IEEE Computer Society.

Barto, L. 2015. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*.

Bessiere, C.; Carbonnel, C.; Hebrard, E.; Katsirelos, G.; and Walsh, T. 2013. Detecting and exploiting subproblem tractability. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 468–474. AAAI Press.

Bulatov, A. A., and Dalmau, V. 2006. A simple algorithm for Mal'tsev constraints. *SIAM J. Comput.* 36(1):16–27.

Bulatov, A. A., and Marx, D. 2010. The complexity of global cardinality constraints. *Logical Methods in Computer Science* 6:1–27.

Bulatov, A. 2002. Mal'tsev constraints are tractable. Technical report, Computing Laboratory, University of Oxford, Oxford, UK.

Bulatov, A. 2003. Tractable conservative constraint satisfaction problems. In *Proceedings 18th IEEE Symposium on Logic in Computer Science, LICS'03*, 321–330.

Bulatov, A. A. 2010. Bounded relational width. Technical report, School of Computer Science, Simon Fraser University.

Carvalho, C.; Egri, L.; Jackson, M.; and Niven, T. 2011. On Maltsev digraphs. In *Computer Science–Theory and Applications*. Springer. 181–194.

Chen, H.; Dalmau, V.; and Grußien, B. 2013. Arc consistency and friends. *J. Log. Comput.* 23(1):87–108.

Dyer, M., and Richerby, D. 2013. An effective dichotomy for the counting constraint satisfaction problem. *SIAM Journal on Computing* 42(3):1245–1274.

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

Goldreich, O. 2010. *P, NP, and NP-Completeness: The basics of computational complexity*. Cambridge University Press.

Green, M. J., and Cohen, D. A. 2008. Domain permutation reduction for constraint satisfaction problems. *Artif. Intell.* 172(8-9):1094–1118.

Idziak, P. M.; Markovic, P.; McKenzie, R.; Valeriote, M.; and Willard, R. 2007. Tractability and learnability arising from algebras with few subpowers. In *LICS*, 213–224. IEEE Computer Society.

Jeavons, P. G.; Cohen, D. A.; and Cooper, M. C. 1998. Constraints, consistency and closure. *Artif. Intell.* 101(1–2):251–265.

Jeavons, P.; Cohen, D. A.; and Gyssens, M. 1997. Closure properties of constraints. *J. ACM* 44(4):527–548.

Kolaitis, P., and Vardi, M. 2000. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences* 61:302–332.

Petke, J., and Jeavons, P. 2009. Tractable benchmarks for constraint programming. Technical report, Technical Report RR-09-07, Computing Laboratory, University of Oxford.

Williams, R.; Gomes, C. P.; and Selman, B. 2003. Backdoors to typical case complexity. In Gottlob, G., and Walsh, T., eds., *IJCAI*, 1173–1178. Morgan Kaufmann.