Electronic Colloquium on Computational Complexity, Report No. 97 (2004)

MAL'TSEV CONSTRAINTS MADE SIMPLE

VÍCTOR DALMAU

Departament de Tecnologia, Universitat Pompeu Fabra
Estació de França, Passeig de la circumval.lació, Barcelona 08003, Spain
victor.dalmau@upf.edu

Abstract. We give in this paper a different and simpler proof of the tractability of Mal'tsev contraints.

1. Introduction

Constraint satisfaction problems arise in a wide variety of domains, such as combinatorics, logic, algebra, and artificial intelligence. An instance of the constraint satisfaction problem (CSP) consists of a set of variables, a set of values (which can be taken by the variables), called domain, and a set of constraints, where a constraint is a pair given by a list of variables, called scope, and a relation indicating the valid combinations of values for the variables in the scope; the goal is to decide whether or not there is an assignment of variables to the variables satisfying all of the constraints. It is sometimes customary to cast the CSP as a relational homomorphism problem [6], namely, the problem of deciding, given a pair $(\mathbf{A}, \mathbf{B})$ of relational structures, whether or not there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. In this formalization, each relation of $\mathbf{A}$ contains tuples of variables that are constrained together, and the corresponding relation of $\mathbf{B}$ contains the allowable tuples of values that the variable tuples may take.

The CSP is NP-complete in general, motivating the search for polynomial-time tractable cases of the CSP. A particularly useful way to restrict the CSP in order to obtain tractable cases is to restrict the types of constraints that may be expressed, by requiring the relations appearing in a constraint to belong to a given fixed set Γ; denote this restriction by CSP(Γ). This form of restriction can capture and place into a unified framework many particular cases of the CSP that have been independently investigated – for instance, the Horn Satisfiability, 2-Satisfiability, and Graph H-Colorability problems. Schaefer was the first to consider the class of problems CSP(Γ); he proved a now famous dichotomy theorem, showing that for every set Γ of relations over a two-element domain, CSP(Γ) is either tractable in polynomial time, or is NP-complete [9]. In recent years, much effort has been directed towards the program of isolating all sets Γ of relations over a finite domain, that give rise to a class of instances of CSP, CSP(Γ), solvable in polynomial time. Impressive progress has been made along these lines leading to the identification of several broad conditions on Γ that guarantee tractability. One of the most prominent achivements in this direction is a recent result due to Bulatov [2] stating that every set Γ of relations on a finite set invariant with respect to a Mal'tsev operation, that is, a ternary operation φ satisfying $\varphi(x, y, y) = \varphi(y, y, x) = x$ for all x, y, gives rise to a tractable problem class. This result encompasses and generalizes several

ISSN 1433-8092

previously known tractable cases of the CSP, such as affine problems [7, 9], constraint satisfaction problems on finite groups with near subgroups and its cosets [6], and paraprimal CSP [5]. Also, several recent advancements in the field, such as the complete classification of CSP problems over a 3-element domain [1] and the conservative CSP [3] make use of this result.

It is fair to say that the original proof of the tractability of Mal'tsev constraints due to Bulatov [2] is very complicated. Furthermore, it makes intensive use of the deep tame congruence theory. In this paper we give a different proof of the tractability of Mal'tsev contraints. The proof presented in this paper is notably simpler than the original proof due to Bulatov and does not require the use of any previous algebraic result; indeed the proof is completely self-contained.

2. Preliminaires

Let A be a finite set and n be a positive integer. A n-ary relation on A is any subset of A^n. In what follows, for every positive integer n, $[n]$ will denote the set $\{1, \ldots, n\}$.

A constraint satisfaction problem is a natural way to express simultaneous requirements for values of variables. More precisely,

Definition 1. An instance of a *constraint satisfaction problem* consists of:

- a finite set of variables, $V = \{v_1, \ldots, v_n\}$;
- a finite domain of values, A;
- a finite set of constraints $\{C_1, \ldots, C_m\}$; each constraint C_l, $l \in [m]$ is a pair $((v_{i_1}, \ldots, v_{i_{k_l}}), S_l)$ where:
  - $(v_{i_1}, \ldots, v_{i_{k_l}})$ is a tuple of variables of length k_l, called the *constraint scope* and
  - S_l is an k_l-ary relation on A, called the *contraint relation*.

A *solution* to a constraint satisfaction problem instance is a mapping $s : V \to A$ such that for each constraint C_l, $l \in [m]$, we have that $(s(v_{i_1}), \ldots, s(v_{k_l})) \in S_l$. Deciding whether or not a given problem instance has a solution is NP-complete in general, even when the constraints are restricted to binary constraints [8] or the domain of the problem has size 2 [4]. However by imposing restrictions on the constraint relations it is possible to obtain restricted versions of the problem that are tractable.

Definition 2. For any set of relations Γ, CSP(Γ) is defined to be the class of decision problems with:

- Instance: A constraint satisfaction problem instance $\mathcal{P}$, in which all constraint relations are elements of Γ.
- Question: Does $\mathcal{P}$ have a solution?

In order to introduce the family of Mal'tsev constraints we need to introduce a some algebraic concepts:

Definition 3. Let $\varphi : A^m \to A$ be an m-ary operation on A and let R be a n-ary relation over A. We say that R is invariant under φ if for all (not necessarily different) tuples $\mathbf{t_1} = (t_1^1, \ldots, t_n^1), \ldots, \mathbf{t_m} = (t_1^m, \ldots, t_n^m)$ in R, the tuple $\varphi(\mathbf{t_1}, \ldots, \mathbf{t_m})$ defined as

$$(\varphi(t_1^1, \ldots, t_1^m), \ldots, \varphi(t_n^1, \ldots, t_n^m))$$

belongs to R.

Given a relation R and an operation φ, we denote by $\langle R\rangle_\varphi$ the smallest relation R' that contains R and that it is invariant under φ. Very often, the operation φ will be clear from the context and we will drop it writting $\langle R\rangle$ instead of $\langle R\rangle_\varphi$.

Let $\varphi : A^m \to A$ be any operation on A. We denote by $\text{Inv}(\varphi)$ the set containg all relations on A invariant under φ.

Definition 4. An ternary operation $\varphi : A^3 \to A$ on a finite set A is called Mal'tsev if it satisfies the following identities

$$\varphi(x, y, y) = \varphi(y, y, x) = x, \quad \forall x, y \in A$$

In this paper we proof the following result:

Theorem 1. *Let φ be a Mal'tsev operation. Then $\text{CSP}(\text{Inv}(\varphi))$ is solvable in polynomial time.*

This result was first proved in [2]. Our proof is given in Section 4.

3. Signatures and Representations

Let A be a finite set, let n be a positive integer, let $\mathbf{t} = (t_1, \ldots, t_n)$ be a n-ary tuple, and let $i_1, \ldots, i_j$ elements in $[n]$. By $\text{pr}_{i_1,\ldots,i_j} \mathbf{t}$ we denote the tuple $(t_{i_1}, \ldots, t_{i_j})$. Similarly, for every n-ary relation R on A and for every $i_1, \ldots, i_j \in [n]$ we denote by $\text{pr}_{i_1,\ldots,i_j} R$ the j-ary relation given by $\{\text{pr}_{i_1,\ldots,i_j} \mathbf{t} : \mathbf{t} \in R\}$.

Let n be a positive integer, let A be a finite set, let $\mathbf{t}$, $\mathbf{t}'$ be n-ary tuples and let (i, a, b) be any element in $[n] \times A^2$. We say that $(\mathbf{t}, \mathbf{t}')$ witnesses (i, a, b) if $\text{pr}_{1,\ldots,i-1} \mathbf{t} = \text{pr}_{1,\ldots,i-1} \mathbf{t}'$, $\text{pr}_i \mathbf{t} = a$, and $\text{pr}_i \mathbf{t}' = b$. We also say that $\mathbf{t}$ and $\mathbf{t}'$ witness (i, a, b) meaning that $(\mathbf{t}, \mathbf{t}')$ witnesses (i, a, b).

Let R be any n-ary relation on A. We define the *signature* of R, $\text{Sig}_R \subseteq [n] \times A^2$, as the set containing all those $(i, a, b) \in [n] \times A^2$ witnessed by tuples in R, that is

$$\text{Sig}_R = \{(i, a, b) \in [n] \times A^2 : \exists \mathbf{t}, \mathbf{t}' \in R \text{ such that } (\mathbf{t}, \mathbf{t}') \text{ witnesses } (i, a, b)\}$$

A subset R' of R is called a *representation* of R if $\text{Sig}_R = \text{Sig}'_R$. If furthermore, $|R'| \leq 2|\text{Sig}_R|$ then R is called a *compact* representation of R. Observe that every relation R has compact representations.

Example 1. Fix a set A, an element $d \in A$, and an integer n. For every $(i, a) \in [n] \times A$ we define the tuple $\mathbf{e}^d_{i,a}$ as the only tuple satisfying

$$\text{pr}_j \mathbf{e}^d_{i,a} = \begin{cases} a & \text{if } i = j \\ d & \text{otherwise} \end{cases}$$

It is easy to observe that for every $(i, a, b) \in [n] \times A^2$, $(\mathbf{e}^d_{i,a}, \mathbf{e}^d_{i,b})$ witnesses (i, a, b). Consequently, the set of tuples $\{\mathbf{e}^d_{i,a} : i \in [n], a \in A\}$ is a representation of the relation A^n. Notice also that it is indeed a compact representation.

The algorithm we propose relies on the following lemma.

Lemma 1. *Let A be a finite set, let $\varphi : A^3 \to A$ be a Mal'tsev operation, let R be a relation on A invariant under φ and let R' be a representation of R. Then $\langle R'\rangle = R$*

Proof. Let n be the arity of R. We shall show that for every $i \in \{1, \ldots, n\}$, $\text{pr}_{1,\ldots,i}\langle R'\rangle = \text{pr}_{1,\ldots,i} R$ by induction on i. The case $i = 1$ follows easily from the fact that for each $\mathbf{t} \in R$, $(1, \text{pr}_1 \mathbf{t}, \text{pr}_1 \mathbf{t})$ is in Sig_R and hence in $\text{Sig}_{R'}$.

So, let us assume that the claim holds for i and let $\mathbf{t}$ be any tuple in R. We will show that $\mathrm{pr}_{1,\dots,i+1}\,\mathbf{t} \in \mathrm{pr}_{1,\dots,i+1}\langle R'\rangle$. By induction hypothesis there exists a tuple $\mathbf{t_1}$ in $\langle R'\rangle$ such that $\mathrm{pr}_{1,\dots,i}\,\mathbf{t_1} = \mathrm{pr}_{1,\dots,i}\,\mathbf{t}$. We have that $(i+1, \mathrm{pr}_{i+1}\,\mathbf{t_1}, \mathrm{pr}_{i+1}\,\mathbf{t})$ belongs to Sig_R, and therefore, there exists some tuples $\mathbf{t_2}$ and $\mathbf{t_3}$ in R' witnessing it. Consequently the tuple $\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})$ belongs to $\langle R'\rangle$. Let us see that $\mathrm{pr}_{1,\dots,i+1}\,\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_{1,\dots,i+1}\,\mathbf{t}$. First, observe that since $\mathbf{t_2}$ and $\mathbf{t_3}$ witness $(i+1, \mathrm{pr}_{i+1}\,\mathbf{t}, \mathrm{pr}_{i+1}\,\mathbf{t_1})$ we have that $\mathrm{pr}_{1,\dots,i}\,\mathbf{t_2} = \mathrm{pr}_{1,\dots,i}\,\mathbf{t_3}$. Because φ is Mal'tsev we can infer that $\mathrm{pr}_{1,\dots,i}\,\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_{1,\dots,i}\,\mathbf{t_1} = \mathrm{pr}_{1,\dots,i}\,\mathbf{t}$. Also, we have that $\mathrm{pr}_{i+1}\,\mathbf{t_1} = \mathrm{pr}_{i+1}\,\mathbf{t_2}$ and $\mathrm{pr}_{i+1}\,\mathbf{t} = \mathrm{pr}_{i+1}\,\mathbf{t_3}$ and consequently $\mathrm{pr}_{i+1}\,\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_{i+1}\,\mathbf{t}$. ■

4. Proof of Theorem 1

We prove Theorem 1 by giving a polynomial-time algorithm that decides correctly whether a CSP(Inv(φ)) instance has a solution.

Let $\mathcal{P} = (\{v_1,\dots,v_n\}, A, \{C_1,\dots,C_m\})$ be a CSP(Inv(φ)) instance which will be the input of the algorithm.

For each $l \in \{0,\dots,m\}$ we define $\mathcal{P}_l$ as the CSP instance that contains the first l constraints of $\mathcal{P}$, that is $\mathcal{P}_l = (\{v_1,\dots,v_n\}, A, \{C_1,\dots,C_l\})$. Furthermore, we shall denote by R_l the n-ary relation on A defined as

$$R_l = \{(s(v_1),\dots,s(v_n)) : s \text{ is a solution of } \mathcal{P}_l)$$

In a nutshell, the algorithm introduced in this section computes for each $l \in \{0,\dots,m\}$ a compact representation R'_l of R_l. In the initial case ($l=0$), $\mathcal{P}_0$ does not have any constraint at all, and consequently, $R_0 = A^n$. Hence, a compact representation of R_0 can be easily obtained as in Example 1. Once a compact representation R'_0 of R_0 has been obtained then the algorithm starts an iterative process in which a compact representation R'_{l+1} of R_{l+1} is obtained from R'_l and the constraint C_{l+1}. This is achieved by means of a call to procedure `Next`, which constitutes the core of the algorithm. The algorithm then, goes as follows:

Algorithm `Solve`$((\{v_1,\dots,v_n), A, \{C_1,\dots,C_m\}))$
Step 1 **select** an arbitrary element d in A
Step 2 **set** $R'_0 := \{\mathbf{e}^d_{i,a} : (i,a) \in [n] \times A\}$
Step 3 **for each** $l \in \{0,\dots,m-1\}$ **do**
(let C_{l+1} be $((v_{i_1},\dots,v_{i_{l+1}}), S_{l+1})$)
Step 3.1 **set** $R'_{l+1} :=$ `Next`$(R'_l, i_1,\dots,i_{l+1}, S_{l+1})$
end for each
Step 4 **if** $R'_m \neq \emptyset$ **return yes**
Step 5 **otherwise return no**

Observe that if we modify step 4 so that the algorithm returns an arbitrary tuple in R'_m instead of "yes" then we have an algorithm that does not merely solve the decision question but actually provides a solution.

Correctness and polynomial time complexity of the algorithm is a direct consequence of the correctness and the running time of the procedure `Next`: As it is shown in Section 4.3 (Lemma 2) at each iteration of Step 3.1, the output of the call `Next`$(R'_l, i_1,\dots,i_{l+1}, S_{l+1})$ is a compact representation of the relation $\{\mathbf{t} \in R_l : \mathrm{pr}_{i_1,\dots,i_{l+1}}\,\mathbf{t} \in S_{l+1}\}$ which is indeed R_{l+1}. Furthermore the cost of the call is $O(n^9+(n+|S_{l+1}|)^4|S_{l+1}|n^3)$ which gives as a total running time for the algorithm polynomial on the size of the input. This finishes the proof of the correctness and time complexity of the algorithm, and hence, of Theorem 1. ∎

The remaining of the paper is devoted to defining and analyzing procedure `Next`. In order to define procedure `Next` it is convenient to introduce previously a pair of simple procedures, namely `Nonempty` and `Fix-values`, which will be intensively used by our procedure `Next`.

4.1. **Procedure `Nonempty`.** This procedure receives as input a compact representation R' of a relation R invariant under φ, a sequence $i_1,\dots,i_j$ of elements in $[n]$ where n is the arity of R, and a j-ary relation S also invariant under φ. The output of the procedure is either an n-ary tuple $\mathbf{t}\in R$ such that $\mathrm{pr}_{i_1,\dots,i_j}\mathbf{t}\in S$ or "no" meaning that such a tuple does not exist.

Procedure `Nonempty`$(R', i_1,\dots,i_j, S)$
Step 1 **set** $U := R'$
Step 2 **while** $\exists \mathbf{t_1},\mathbf{t_2},\mathbf{t_3}\in U$ such that $\mathrm{pr}_{i_1,\dots,i_j}\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})\notin \mathrm{pr}_{i_1,\dots,i_j} U$ **do**
Step 2.1 **set** $U := U\cup\{\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})\}$
endwhile
Step 3 **if** $\exists\mathbf{t}$ in U such that $\mathrm{pr}_{i_1,\dots,i_j}\mathbf{t}\in S$ **then return t**
Step 4 **else return** "no"

We shall start by studying its correctness. First observe that every tuple in U belongs initially to R' (and hence to R), or it has been obtained by applying φ to some tuples $\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}$ that previously belong to U. Therefore, since R is invariant under φ, we can conlude that $U\subseteq R$. Consequently, if a tuple $\mathbf{t}$ is returned in step 3, then it belongs to R and also satisfies that $\mathrm{pr}_{i_1,\dots,i_j}\mathbf{t}\in S$, as desired. It only remains to show that if a "no" is returned in step 4 then there not exists any tuple $\mathbf{t}$ in R such that $\mathrm{pr}_{i_1,\dots,i_j}\mathbf{t}\in S$. In order to do this we need to show some simple facts about U. Notice that at any point of the execution of the procedure $R'\subseteq U$. Then U is also a representation of R and hence $\langle U\rangle = R$. Therefore we have that

$$\langle \mathrm{pr}_{i_1,\dots,i_j} U\rangle = \mathrm{pr}_{i_1,\dots,i_j}\langle U\rangle = \mathrm{pr}_{i_1,\dots,i_j} R$$

By the condition on the "while" of step 2 we have that when the procedure leaves the execution of step 2 it must be case that for all $\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}\in U$, $\mathrm{pr}_{i_1,\dots,i_j}\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})\in \mathrm{pr}_{i_1,\dots,i_j} U$ and consequently $\mathrm{pr}_{i_1,\dots,i_j} U = \langle\mathrm{pr}_{i_1,\dots,i_j} U\rangle = \mathrm{pr}_{i_1,\dots,i_j} R$. Hence, if there exists some $\mathbf{t}$ in R such that $\mathrm{pr}_{i_1,\dots,i_j}\in S$ then it must exists some $\mathbf{t'}$ in U such that $\mathrm{pr}_{i_1,\dots,i_j}\in S$ and we are done.

Let us study now the running time of the procedure. It is only necessary to focus on steps 2 and 3. At each iteration of the loop in step 2, cardinality of U increases by one. So we can bound the number of iterations by the size $|U|$ of U at the end of the execution of the procedure.

The cost of each of the iteration is basically dominated by the cost of checking whether there exists some tuples $\exists\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}\in U$ such that $\mathrm{pr}_{i_1,\dots,i_j}\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})\notin \mathrm{pr}_{i_1,\dots,i_j} U$ done in step 2. In order to try all possible combinations for $\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}$ in U, $|U|^3$ steps suffice. Each one of these steps requires time $O(|U|n)$, as tuples have arity n and checking whether $\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})$ belongs to U can be do naively by a sequential search in U. Thus, the total running time of step 2 is $O(|U|^4n)$.

The cost of step 3 is the cost of finding a tuple $\mathbf{t}$ in U satisfying $\mathrm{pr}_{i_1,\dots,i_j}\mathbf{t} \in S$ which is $O(|U||S|n)$. Putting all together we obtain that the complete running time of the procedure is $O(|U|^5 n + |U||S|n)$ which we can bound by $O(|U|^5|S|n)$. Let us bound the size of U (at the end of the execution of the procedure): At each iteration of the loop in step 2, the size of $\mathrm{pr}_{i_1,\dots,i_j} U$ increases. Hence, the number of such iterations is bounded by $|\mathrm{pr}_{i_1,\dots,i_j} R|$. Since R' is compact its cardinality is bounded by $2n|A|^2$ wich is $O(n)$ as $|A|$ is fixed. Consequently the total running time of the procedure can be bounded by $O((n + |\mathrm{pr}_{i_1,\dots,i_j} R|)^5|S|n)$.

4.2. **Procedure `Fix-values`.** This procedure receives as input a canonical representation R' of a relation R invariant under φ and a sequence $a_1,\dots,a_m$, $m \le n$ of elements of A (n is the arity of R). The output is a compact representation of the relation given by

$$\{\mathbf{t} \in R : \mathrm{pr}_1\mathbf{t} = a_1, \dots, \mathrm{pr}_m\mathbf{t} = a_m\}$$

Procedure `Fix-values`$(R', a_1, \dots, a_m)$

Step 1 **set** $j := 0$; $U_j := R'$
Step 2 **while** $j < m$ **do**
Step 2.1 **set** $U_{j+1} := \emptyset$
Step 2.2 **for each** $(i,a,b) \in [n] \times A^2$ **do**
Step 2.2.1 **if** $\exists \mathbf{t_2}, \mathbf{t_3} \in U_j$ witnessing (i,a,b) and
(we assume that if $a = b$ then $\mathbf{t_2} = \mathbf{t_3}$) and
`Nonempty`$(U_j, j+1, i, \{(a_{j+1}, a)\}) \neq$"no" and
$i > j+1$ or $a = b = a_i$ **then**
(let $\mathbf{t_1}$ be the tuple returned by `Nonempty`$(U_j, j+1, i, \{a_{j+1}, a\})$)
set $U_{j+1} := U_{j+1} \cup \{\mathbf{t_1}, \varphi(\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3})\}$
end for each
Step 2.4 **set** j:=j+1
end while
Step 3 **return** U_m

Let us study the correctness of the procedure. We shall show by induction on $j \in \{0,\dots,m\}$ that U_j is a compact representation of $R_j = \{\mathbf{t} \in R : \mathrm{pr}_1\mathbf{t} = a_1, \dots, \mathrm{pr}_j\mathbf{t} = a_j\}$. The case $j = 0$ is correctly settled in step 1. Hence it is only necessary to show that at every iteration of the while loop in step 2, if U_j is a compact representation of R_j then U_{j+1} is a compact representation of R_{j+1}. It is easy to see that if any of the conditions of the "if" in step 2.2.1 is falsified then (i,a,b) is not in Sig_R. So it only remains to see that when the "if" in step 2.2.1 is satisfied, we have that (a) $\mathbf{t_1}$ and $\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})$ are tuples in R_{j+1}, and (b) $(\mathbf{t_1}, \varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}))$ witnesses (i,a,b).

Proof of (a): As $\mathbf{t_1} =$ `Nonempty`$(U_j, j+1, i, \{(a_{j+1}, a)\})$, we can conclude that $\mathbf{t_1}$ belongs to R_j, $\mathrm{pr}_{j+1}\mathbf{t_1} = a_{j+1}$, and $\mathrm{pr}_i\mathbf{t_1} = a$. Consequently $\mathbf{t_1}$ belongs to R_{j+1}. Furthermore, as $\mathbf{t_1}$, $\mathbf{t_2}$, and $\mathbf{t_3}$ are in R_j and R_j is invariant under φ, $\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})$ belongs to R_j. Let us see now that $\mathrm{pr}_{j+1}\mathbf{t_2} = \mathrm{pr}_{j+1}\mathbf{t_3}$ by means of a case analisis. If $i > j+1$ then we have that $\mathrm{pr}_{j+1}\mathbf{t_2} = \mathrm{pr}_{j+1}\mathbf{t_3}$ as $(\mathbf{t_2},\mathbf{t_3})$ witnesses (i,a,b). If $i \le j+1$ then $a = b = a_i$ and hence $\mathbf{t_2}$ and $\mathbf{t_3}$ are identical.

Finally, since φ is Mal'tsev, $\mathrm{pr}_{j+1}\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_{j+1}\mathbf{t_1} = a_{j+1}$ and hence $\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3})$ belongs to R_{j+1}.

Proof of (b): Since $(\mathbf{t_2}, \mathbf{t_3})$ witnesses (i,a,b) we have that $\mathrm{pr}_{1,\dots,i-1}\,\mathbf{t_2} = \mathrm{pr}_{1,\dots,i-1}\,\mathbf{t_3}$ Consequently, $\mathrm{pr}_{1,\dots,i-1}\,\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_{1,\dots,i-1}\,\mathbf{t_1}$. Furthermore we also have that $\mathrm{pr}_i\,\varphi(\mathbf{t_1},\mathbf{t_2},\mathbf{t_3}) = \mathrm{pr}_i(a,a,b) = b$.

Notice that at each iteration at most 2 tuples are added for each (i,a,b) in $\mathrm{Sig}_{R_{j+1}}$. Consequently, U_{j+1} is compact. This completes the proof of its correctness.

Let us study now its time complexity. The "while" loop at step 2 is performed $m \le n$ times. At each iteration the procedure executes another loop (step 2.2). The "for each" loop at step 2.2 is executed for each (i,a,b) in $[n]\times A^2$, that is, a total number of $n|A|^2$ times. The cost of each iteration of the loop is basically dominated by the cost of the call to procedure `Nonempty` which costs $O((n+|A|^2)^4 n) = O(n^5)$. Thus the total cost of the procedure is then $O(n^7)$.

4.3. **Procedure `Next`.** We are now almost in a position to introduce procedure `Next`. Procedure `Next` receives as input a canonical representation R' of a relation R invariant under φ, a sequence $i_1,\dots,i_j$ of elements in $[n]$ where n is the arity of R, and a j-ary relation S invariant under φ. The output of `Next` is a compact representation of the relation $R^* = \{\mathbf{t}\in R : \mathrm{pr}_{i_1,\dots,i_j}\,\mathbf{t}\in S\}$. It is an easy exercise to verify that R^* must also be invariant under φ.

We shall start by defining a procedure, called `Next-beta` that although equivalent to `Next` has a worse running time. In particular, the running time of `Next-beta` might be exponential with respect to the size of its input.

Procedure `Next-beta`$(R', i_1,\dots,i_j,S)$
Step 1 **set** $U := \emptyset$
Step 2 **for each** $(i,a,b)\in[n]\times A^2$ **do**
Step 2.1 **if** `Nonempty`$(R', i_1,\dots,i_j, i, S\times\{a\})\} \neq$"no" **then**
(let $\mathbf{t}$ be `Nonempty`$(R', i_1,\dots,i_j, i, S\times\{a\})\})$
Step 2.2 **if** `Nonempty`(`Fix-values`$(R', \mathrm{pr}_1\,\mathbf{t},\dots,\mathrm{pr}_{i-1}\,\mathbf{t}), i_1,\dots,i_j,i,S\times\{b\}) \neq$"no"
(let $\mathbf{t}'$ be `Nonempty`(`Fix-values`$(R', \mathrm{pr}_1\,\mathbf{t},\dots,\mathrm{pr}_{i-1}\,\mathbf{t}), i_1,\dots,i_j,i,S\times\{b\}))$
set $U := U\cup\{\mathbf{t},\mathbf{t}'\}$
end for each
Step 3 **return** U

The overall structure of procedure `Next-beta` is similar to that procedure `Fix-values`. Observe that the condition of the "if" statement

$$\texttt{Nonempty}(R', i_1,\dots,i_j, i, S\times\{a\})\} \neq \text{"no"}$$

of step 2.1 is satisfied if and only if there exists a tuple $\mathbf{t}\in R$ such that $\mathrm{pr}_{i_1,\dots i_j}\,\mathbf{t}\in S$ and $\mathrm{pr}_i\,\mathbf{t} = a$. Hence if such a tuple does not exist then (i,a,b) is not in Sig_{R^*} and nothing needs to be done for (i,a,b). Now consider the condition of the "if" statement in step 2.2 which is given by

$$\texttt{Nonempty}(\texttt{Fix-values}(R', \mathrm{pr}_1\,\mathbf{t},\dots,\mathrm{pr}_{i-1}\,\mathbf{t}), i_1,\dots,i_j,i,S\times\{b\}) \neq \text{"no"}$$

This condition is satisfied if and only if there exists some $\mathbf{t}'$ in R such that $\mathrm{pr}_{i_1,\dots,i_j}\,\mathbf{t}'\in S$ such that $\mathrm{pr}_{1,\dots,i-1}\,\mathbf{t}' = \mathrm{pr}_{1,\dots,i-1}\,\mathbf{t}$ and $\mathrm{pr}_i\,\mathbf{t}' = b$. It is immediate to see that if the condition holds then $(\mathbf{t},\mathbf{t}')$ witnesses (i,a,b). It only remains to show that if $(i,a,b)\in\mathrm{Sig}_{R^*}$ then such a $\mathbf{t}'$ must exist. In order to do it, it is only necessary to verify that if $\mathbf{t_a},\mathbf{t_b}$ are tuples in R^* witnessing (i,a,b) then, since φ is Mal'tsev, the tuple $\varphi(\mathbf{t},\mathbf{t_a},\mathbf{t_b})$ satisfies the desired properties (here $\mathbf{t}$ is the tuple returned by the call to procedure `Nonempty` in step 2.1).

Again, the cardinality of U is bounded by $2|\operatorname{Sig}_{R^*}|$ and, hence, U is a compact representation.

Let us study the running time of procedure `Next-beta`. The loop of step 2 is performed $n|A|$ times and the cost of each iteration is basically the cost of steps 2.1 and 2.2 in which other procedures are called. The cost of calling $\texttt{Nonempty}(R', i_1, \ldots, i_j, i, S \times \{a\})\}$ in step 2.1 is $O((n+r)^4|S|n)$ where r is $|\operatorname{pr}_{i_1,\ldots,i_j} R|$. The cost of calling

$$\texttt{Nonempty}(\texttt{Fix-values}(R', \operatorname{pr}_1 \mathbf{t}, \ldots, \operatorname{pr}_{i-1} \mathbf{t}), i_1, \ldots, i_j, i, S \times \{b\})$$

in step 2.2 is the sum of the call to `Fix-values` which is $O(n^7)$ and the call to `Nonempty` which is $O((n+r)^4|S|n)$. Therefore, the total cost of an iteration of the loop of step 2 is $O((n+r)^4|S|n + n^7)$ and hence, the total running time for the procedure is $O((n+r)^4|S|n^2 + n^8)$

Let us take a closer look at the value of $r = |\operatorname{pr}_{i_1,\ldots,i_j} R$. It is important to notice here that the set of possible constraints S that can appear in an instance is *infinite* and henceforth it is not possible to bound the value of j. Consequently, the value of r might be exponential in the worst case. However, it would be possible to bound the value of j and get a polynomial bound for r if a *finite* subset Γ of $\operatorname{Inv}(\varphi)$ is fixed beforehand and we assume that all constraint instances use only constraint relations from Γ. That is, by using the procedure `Next-beta` if could be possible to define a polynomial-time algorithm that solves $\mathrm{CSP}(\Gamma)$ for every finite subset Γ of $\operatorname{Inv}(\varphi)$. However we are aiming here for a more general result. To this end, we define a new procedure `Next` which makes a sequence of calls to `Next-beta`.

Procedure $\texttt{Next}(R', i_1, \ldots, i_j, S)$

Step 1 **set** $l := 0$, $U_l := R'$

Step 2 **while** $l < j$ **do**

Step 2.1 **set** $U_{l+1} := \texttt{Next-beta}(U_l, i_1, \ldots, i_{l+1}, \operatorname{pr}_{i_1,\ldots,i_{l+1}} S)$

end while

Step 3 **return** U_j

Observe that at each call of the procedure `Next` in step 2.1, the value of r can be bounded by $|\operatorname{pr}_{i_1,\ldots,i_l} S||A|$, and hence the running time of each call is $O(n^8 + (n+|S|)^4|S|n^2)$. Thus, we have just proved

Lemma 2. *For every $n \geq 1$, every n-ary relation R invariant under φ, every compact representation R' of R, every $i_1, \ldots, i_j \in [n]$, and every j-ary relation S invariant under φ,* $\texttt{Next}(R', i_1, \ldots, i_j, S)$ *computes a compact representation of $R^* = \{\mathbf{t} \in R : \operatorname{pr}_{i_1,\ldots,i_j} \in S\}$ in time $O(n^9 + (n+|S|)^4|S|n^3)$. Furthermore R^* is invariant under φ.*

Corollary 1. *Algorithm* `Solve` *decides correctly if an instance $\mathcal{P}$ of* $\mathrm{CSP}(\inf(\varphi))$ *is satisfiable in time $O(mn^8 + m(n+|S^*|)^4|S^*|n^2)$ where n is the number of variables of $\mathcal{P}$, m is its number of contraints and S^* is the largest constraint relation occurring in $\mathcal{P}$.*

References

[1] A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *In Proceedings of 43rd IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 649–658, 2002.

[2] A. Bulatov. Mal'tsev constraints are tractable. Technical Report PRG-02-05, Computing Laboratory, Oxford University, 2002.

[3] A.A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th Annual IEEE Simposium on Logic in Computer Science*, pages 321–330, Ottawa, Canada, June 2003. IEEE Computer Society.

[4] S.A. Cook. The Complexity of Theorem-Proving Procedures. In *3rd Annual ACM Symposium on Theory of Computing STOC'71*, pages 151–158, 1971.

[5] V. Dalmau. A New Tractable Class of Constraint Satisfaction Problems. In *6th International Symposium on Artificial Intelligence and Mathematics*, 2000.

[6] T. Feder and M.Y. Vardi. The Computational Structure of Monotone Monadic SNP and Contraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Computing*, 28(1):57–104, 1998.

[7] P. Jeavons, D. Cohen, and M. Gyssens. Closure Properties of Constraints. *Journal of the ACM*, 44(4):527–548, July 1997.

[8] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[9] T.J. Schaefer. The Complexity of Satisfiability Problems. In *10th Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'