Negative set constraints with equality*

Witold Charatonik†
Institute of Computer Science
University of Wrocław,
wch@ii.uni.wroc.pl

Leszek Pacholski‡
Institute of Mathematics
Polish Academy of Sciences
pacholsk@ii.uni.wroc.pl

Abstract

Systems of set constraints describe relations between sets of ground terms. They have been successfully used in program analysis and type inference. So far two proofs of decidability of mixed set constraints have been given: by R. Gilleron, S. Tison and M. Tommasi [12] and A. Aiken, D. Kozen, and E.L. Wimmers [3]. However, both these proofs are long, involved and do not seem to extend to more general set constraints.

Our approach is based on a reduction of set constraints to the monadic class given in a recent paper by L. Bachmair, H. Ganzinger, and U. Waldmann [7]. We first give a new proof of decidability of systems of mixed positive and negative set constraints. We explicitely describe a very simple algorithm working in NEXPTIME and we give in all detail a relatively easy proof of its correctness. Then, we sketch how our technique can be applied to get various extensions of this result. In particular we prove that the problem of consistency of mixed set constraints with restricted projections and unrestricted diagonalization is NEXPTIME.

1 Introduction

Set constraints give a natural formalism for many problems in program analysis and type inference. They have been studied and applied in many papers including [5], [4], [14], [15], [17], [18], [20].

The first general results concerning the decidability of set constraints were obtained by Heintze and Jaffar [13], who studied the so-called definite set constraints. The decidability of systems of positive set constraints was established by A. Aiken and E.L. Wimmers [6]. Later other proofs have been obtained. R. Gilleron, S. Tison, and M. Tommasi [11] gave a proof based on automata theoretic techniques and L. Bachmair, H. Ganzinger, and U. Waldmann [7] gave a simple and very elegant proof using the decision procedure for the first order theory of monadic predicates, providing also NEXPTIME-completeness of the problem of solvability of positive set constraints. In a paper by A. Aiken, D. Kozen, M. Vardi, and E.L. Wimmers [2], yet another algorithm has been presented and a detailed analysis of the complexity of positive set constrains has been given.

The problem of the decidability of sets constraints with negated inclusion was more difficult. Two solutions were obtained independently by Aiken, Kozen, and Wimmers [3], and Gilleron, Tison and Tommasi [12]. Both solutions are quite non-trivial.

The solution by Aiken, Kozen and Wimmers uses ideas of [2] and goes through normal forms, a reduction to another problem concerning hypergraphs, a reduction to the Diophantine (nonlinear) reachability problem, and solving the last one in some sense again using graph-theoretic tools. Later an improvement of this result was obtained by K. Stefansson [19], who simplified the last part of the proof, obtained NP-completeness of the Diophantine reachability problem, and thus established NEXPTIME-completeness of the original problem.

The solution by Gilleron, Tison and Tommasi extends ideas of [11] and is based on the notion of a tree set automaton. The paper presents an elegant theory of tree set automata, describes how set constraints can be expressed using this notion and proof the decidability of the non-emptiness problems for tree set automata thus establishng the decidability of sets constraints. The first two parts of the paper are quite elegant. However, the proof of decidability of the emptiness problem is long and contains several difficult technical lemmas whose proof are rather difficult to follow.

Our proof is based on the idea of L. Bachmair, H. Ganzinger, and U. Waldmann [7] to reduce the decidability problem for positive set constraints to the problem of consistency of first order theories of unary predicates. It has been known for some time (see [1], [16]), that it is decidable if a finite set of first order sentences in a language $\mathcal{L}$ having only unary predicates has a model. Moreover, for sentences without equality the size of such a model can be bounded by 2^N, where N is the number of predicate symbols in $\mathcal{L}$. In [7] a simple method was given for translating the problem of consistency of positive set constraitns to the problem of consistency of finite sets of sentences in a first order unary language, and a method of obtaining solutions of systems of positive set constraints from the corresponding finite models of monadic theories.

*This resarch has been supported by KBN grants 2 1197 91 01 and 2 1368 91 01.

†Partially supported by TEMPUS JEP 1941.

‡Partially supported by CNRS through INRIA-CRIN and by Université Paris–Sud

The translation of set constraints to monadic theories works also for negative set constraints. Our extension solves the problem of how to use a finite model of a monadic theory to get a solution of the corresponding system of set constraints. The main problem is to make sure that some sets of terms are non empty. The first observation is that each solution of a system of set constraints gives a finite approximation, called a *history* which is sufficient to reconstruct a (perhaps different) regular solution. Then, the proof proceeds by distinguishing a small subset, *the skeleton* of a history, which is small, is not always a history, but can be collapsed to obtain a small history. Now, what is left, is to make an exhaustive search through all *small histories*. Such a search is described below in the algorithm **EASY**.

In the last section we consider set constraints with projections and diagonalizations. First, just by changing some parameters in **EASY**, we prove that we can extend to the case of mixed constraints some results of [7] on positive set constraints with restricted projection and diagonalization. Then we sketch a proof of a theorem saying that in fact the restrictions on diagonalizations can be lifted. The diagonalization is a way of introducing a weak form of equality which allows to compare brothers in a term. It has been proved by B. Bogaert and S. Tison [8] that if the comparison of cousins is allowed, then the problem (or even its special case) becomes undecidable.

2 Basic definitions

Let $\Sigma = \{a_1^0, \ldots, a_{i_0}^0, f_1^1, \ldots, f_{i_1}^1, \ldots, f_1^m, \ldots, f_{i_m}^m\}$, where $a_1^0, \ldots, a_{i_0}^0$ are constant symbols, $f_1^1, \ldots, f_{i_m}^m$ are function symbols and f_j^i stands for j-th symbol of arity i in Σ, and let $\mathcal{V}$ be a set of second order variables. Set expressions are defined by the grammar

$$E ::= 0 \mid 1 \mid \alpha \mid E \cup E \mid E \cap E \mid \bar{E} \mid f_i^k(E_1, ..., E_k)$$

where α is a set variable in $\mathcal{V}$ and $f_i^k \in \Sigma$. A *positive* set constraint is a relation of the form $E \subseteq E'$, and a *negative* set constraint has a form $E \not\subseteq E'$.

Consider a system (SC) of set constraints

$$E_1 \not\subseteq E_1' \wedge \ldots \wedge E_l \not\subseteq E_l' \wedge E_{l+1} \subseteq E_{l+1}' \wedge \ldots \wedge E_k \subseteq E_k'.$$

Let $E(SC)$ denote set of all subexpressions of $E_1, E_1', \ldots, E_k, E_k'$. Following [7] for each expression $E \in E(SC)$ we introduce a predicate symbol P_E. Let φ' be the universal closure of the conjunction of the formulas defining predicates P_E for all $E \in E(SC)$ (i.e. formulas of the form
$P_{E_1 \cup E_2}(y_1) \leftrightarrow P_{E_1}(y_1) \vee P_{E_2}(y_1), \ldots,$
$P_{f_j^i(E_1,\ldots,E_i)}(f_j^i(y_1, \ldots, y_i)) \leftrightarrow P_{E_1}(y_1) \wedge \ldots \wedge P_{E_i}(y_i),$
$P_{f_j^i(E_1,\ldots,E_i)}(g(y_1, \ldots, y_{i'})) \leftrightarrow \mathit{false}$ for $g \in \Sigma \setminus \{f_j^i\}$).

For the system (SC) the formula

$$\begin{aligned}&\exists x_1 \ldots \exists x_l \forall y_1\\ &\quad (P_{E_1}(x_1) \wedge \neg P_{E_1'}(x_1)) \wedge \ldots \wedge (P_{E_l}(x_l) \wedge \neg P_{E_l'}(x_l))\\ &\quad \wedge (P_{E_{l+1}}(y_1) \rightarrow P_{E_{l+1}'}(y_1)) \wedge \ldots \wedge\\ &\quad \wedge (P_{E_k}(y_1) \rightarrow P_{E_k'}(y_1)) \wedge \varphi'\end{aligned}$$

expresses the satisfiability of (SC). Moving the quantifiers of φ' outward, we obtain a formula of the form

$$\begin{aligned}&\exists x_1 \ldots \exists x_l \forall y_1 \ldots \forall y_m\\ &\quad \varphi(x_1, \ldots, x_l, y_1, \ldots, y_m, a_1^0, \ldots, a_{i_0}^0, f_1^1(y_1), \ldots\\ &\qquad \ldots, f_{i_m}^m(y_1, \ldots, y_m)),\end{aligned}$$

with φ built using only monadic predicate symbols (and the symbols listed above). It can easily be seen that this formula is a (partial) skolemization of the formula ψ below.

$$\begin{aligned}&\exists x_1 \ldots \exists x_l \exists x_1^0 \ldots \exists x_{i_0}^0 \forall y_1 \exists x_1^1 \ldots \exists x_{i_1}^1\\ &\forall y_2 \exists x_1^2 \ldots \exists x_{i_2}^2 \ldots \forall y_m \exists x_1^m \ldots \exists x_{i_m}^m\\ &\quad \varphi(x_1, \ldots, x_l, y_1, \ldots, y_m, x_1^0, \ldots, x_{i_0}^0, x_1^1, \ldots, x_{i_m}^m)\end{aligned}$$

Let N be the number of predicate symbols in $\mathcal{P} = \{P_E \mid E \in E(SC)\}$. If I is a model of ψ then we consider the equivalence relation $\equiv$ in I such that $x \equiv y$ if and only if for all $P \in \mathcal{P}$ we have $P^I(x) \leftrightarrow P^I(y)$, where P^I is the interpretation of P in I. Let M be the number of different nonempty equivalence classes of the relation $\equiv$. We will often identify the interpretation of a unary predicate symbol P^I with the subset $\{x \mid P(x)\}$ of the domain of I.

Lemma 2.1 (see [1], p.34) *A monadic formula (without equality) is satisfiable if and only if it has a finite model (of cardinality $M \leq 2^N$) such that in each equivalence class of the relation $\equiv$ there is at most one element.*

3 The algorithm

In this section we give an algorithm solving set constraints with negated inclusion. The algorithm is nondeterministic with exponential time complexity. This together with the results from [7] gives NEXPTIME-completeness of the problem. Note that except the first step, when we choose a model of a monadic formula (which requires nondeterministic exponential time, see [16] for details), the algorithm is nondeterministic polynomial.

Steps 2 and 3 implement a standard method of building Herbrand models. Step 4 gives a definition of a solution, if one exists. Note that in this step we do not compute the whole solution, we just give a description of how to compute it. Since this description is finite, the solution is regular (see [12] for a definition of regularity).

algorithm EASY

input: formula ψ representing set constraint (SC)
output: solution of (SC) if such a solution exists, "NO SOLUTION" if it does not exist

1. Nondeterministically choose a model of ψ consisting of a finite domain D (of cardinality $M \leq 2^N$) and a subset P_E^D of D, for each $E \in E(SC)$, such that in each equivalence class there is at most one element. If such a model does not exist then **return(**"NO SOLUTION"**)**.

2. Nondeterministically choose $d_1^0, \dots d_{i_0}^0 \in D$ such that

$$\exists x_1 \dots \exists x_l \forall y_1 \exists x_1^1 \dots \exists x_{i_1}^1 \forall y_2 \dots \exists x_{i_m}^m \\ \varphi(x_1, \dots, x_l, y_1, \dots, y_m, d_1^0, \dots, d_{i_0}^0, x_1^1, \dots, x_{i_m}^m)$$

holds in D. Such $d_1^0, \dots d_{i_0}^0$ exist because ψ is satisfied in our interpretation. Put $\Phi(a_j^0) = d_j^0$ and $H = \{a_1^0, \dots a_{i_0}^0\}$.

3. Repeat this step until $|H| \geq 2M^3$ or $\Phi(H) = D$:

Choose a function symbol $f_j^k \in \Sigma$ and a sequence $t_1, \dots t_k$ of terms from H such that $\Phi(f_j^k(t_1, \dots t_k))$ has not yet been defined. Choose $d_j^k \in D$ such that

$$\exists x_1 \dots \exists x_l \exists x_1^1 \dots \exists x_{j-1}^k \exists x_{j+1}^k \dots \exists x_{j_k}^k \forall y_{k+1} \dots \exists x_{i_m}^m \\ \varphi(x_1, \dots, x_l, \Phi(t_1), \dots, \Phi(t_k), y_{k+1}, \dots, y_m, \\ d_1^0, \dots, d_{i_0}^0, x_1^1, \dots, d_j^k, \dots, x_{i_m}^m)$$

holds in D. Add $f_j^k(t_1, \dots, t_k)$ to H and define $\Phi(f_j^k(t_1, \dots, t_k)) = d_j^k$.

4. If $\Phi(H) = D$ then fix terms $t^1, \dots, t^M \in H$ such that $\Phi(\{t^1, \dots, t^M\}) = D$. If $\Phi(f_j^i(t^{j_1}, \dots, t^{j_i}))$ is not defined for some $f_j^i \in \Sigma$ and some $t^{j_1}, \dots, t^{j_i}$ among $t^1, \dots, t^M$, then define it in the same way as in the step 3. Extend Φ to $T(\Sigma)$ by defining $\Phi(f_j^i(t_1, \dots, t_i)) = \Phi(f_j^i(t^{j_1}, \dots, t^{j_i}))$ where $\Phi(t^{j_k}) = \Phi(t_k)$.

Now for each set variable $X \in E(SC)$ **return(**"SOLUTION:" $X = \Phi^{-1}(P_X^D)$**)**.

If for all nondeterministic choices in steps 1–3 $\Phi(T) \subsetneq D$ then **return(**"NO SOLUTION"**)**.

4 Correctness of the algorithm

4.1 Soundness

Theorem 4.1 *The algorithm gives only correct solutions.*

Proof. Consider an interpretation with the domain $T(\Sigma)$ and predicates $P_E^{T(\Sigma)}$ for $E \in E(SC)$ defined as follows: $P_E^{T(\Sigma)}(t) \leftrightarrow P_E^D(\Phi(t))$. It suffices to show that under this interpretation the formula

$$\exists x_1 \dots \exists x_l \forall y_1 \dots \forall y_m \\ \varphi^{T(\Sigma)}(x_1, \dots, x_l, y_1, \dots, y_m, a_1^0, \dots, a_{i_0}^0, \\ f_1^1(y_1), \dots, f_{i_m}^m(y_1, \dots, y_m))$$

is satisfied. Take $d_1, \dots d_l \in D$ such that the formula

$$\exists x_1^0 \dots \exists x_{i_0}^0 \forall y_1 \exists x_1^1 \dots \exists x_{i_m}^m \\ \varphi(d_1, \dots, d_l, y_1, \dots, y_m, x_1^0, \dots, x_{i_0}^0, x_1^1, \dots, x_{i_m}^m)$$

is satisfied, and $x_i \in \Phi^{-1}(d_i)$. Then for any $t_1, \dots, t_m \in T(\Sigma)$ we get

$$\begin{aligned} & \varphi^{T(\Sigma)}(x_1, \dots, x_l, t_1, \dots, t_m, a_1^0, \dots, a_{i_0}^0, \\ & \quad f_1^1(t_1), \dots, f_{i_m}^m(t_1, \dots, t_m)) \leftrightarrow \\ \leftrightarrow\ & \varphi^D(\Phi(x_1), \dots, \Phi(x_l), \Phi(t_1), \dots, \Phi(t_m), \Phi(a_1^0), \dots \\ & \dots, \Phi(a_{i_0}^0), \Phi(f_1^1(t_1)), \dots, \Phi(f_{i_m}^m(t_1, \dots, t_m))) \leftrightarrow \\ \leftrightarrow\ & \varphi^D(d_1, \dots, d_l, \Phi(t_1), \dots, \Phi(t_m), \\ & \quad d_1^0, \dots, d_{i_0}^0, d_1^1, \dots, d_{i_m}^m) \leftrightarrow \\ \leftrightarrow\ & true \end{aligned}$$

□

4.2 Completeness

A solution of (SC) is a function $\mathcal{S} : \mathcal{X} \to \mathcal{P}(T(\Sigma))$, where $\mathcal{X}$ is the set of variables occurring in (SC). This function can be extended in a unique way to a function from $E(SC)$ to $\mathcal{P}(T(\Sigma))$. If $E(SC) = \{E_1, \dots, E_N\}$, then let $\mathcal{D} = \{\mathcal{S}(\varepsilon(E_1)) \cap \dots \cap \mathcal{S}(\varepsilon(E_N)) \mid \varepsilon(E_i) \in \{E_i, \overline{E_i}\}\}$. We can identify $\mathcal{D}$ with the set $T(\Sigma)/\equiv$ of equivalence classes of the relation $\equiv$, so $\mathcal{D}$ contains at most 2^N elements which are disjoint subsets of $T(\Sigma)$. Let Φ be the quotient mapping from $T(\Sigma)$ into $\mathcal{D}$, i.e. $\Phi(t) = d \in \mathcal{D}$ if $t \in d$, so the sentence $\Phi(t_1) = \Phi(t_2)$ is equivalent to the sentence $t_1 \equiv t_2$. In this section while talking about a solution of (SC) we will think of a function Φ rather then of $\mathcal{S}$. Of course if we have the function Φ we can reconstruct the function $\mathcal{S}$ putting for all $X \in \mathcal{X}$ $\mathcal{S}(X) = \Phi^{-1}(\bigcup\{\varepsilon(E_1) \cap \dots \cap \varepsilon(E_N) \in \mathcal{D} \mid \varepsilon(X) = X\})$.

We fix a linear ordering $\prec$ on terms such that if t is a term of depth lower then the depth of s, then $t \prec s$. The maximal subterm of a term $f(s_1, \dots, s_k)$ is a term s_j among $s_1, \dots, s_k$ such that for all $i \leq k$, $s_i \prec s_j$ (if there are two maximal subterms $s_j = s_l$ with $j < l$ then the first occurrence, i.e. s_j is the maximal subterm).

Definition 4.2 *A history of a solution Φ of (SC) is a finite set of terms (labeled by their image under Φ) containing the minimal terms in each equivalence class and closed under taking subterms.*

Lemma 4.3 *A history of a solution gives all necessary information to construct a (perhaps different, regular) solution.*

Proof. With a history of solution on input, **EASY** does not need to proceed with steps 1–3; it can at once go to step 4 and extend this information to a solution. □

In this subsection we will show that if this set is too big then we are able to construct a new solution with a smaller one. The key observation here is that if $f(s_1, \ldots, t, \ldots, s_n)$ belongs to the history H and there is no other term in H containing t as a subterm, then if we can replace $f(s_1, \ldots, t, \ldots, s_n)$ by $f(s'_1, \ldots, t, \ldots, s'_n)$ where $s_i \equiv s'_i$ and s'_i are minimal in their equivalence classes, we obtain a smaller history.

In the text below $c[i]$ denotes the i-th term (according to the ordering $\prec$) in the set $\Phi^{-1}(c)$, and the set of *milestones* is a subset of the history (defined in the background of the definitions below). We call a term *composed* if it is not a constant symbol.

Definition 4.4 *The semi-skeleton of a history of a solution Φ of (SC) is a labeled graph $\mathcal{M}$ constructed in the following way:*

- *initialize the set of* milestones *as the set containing all the minimal terms in each equivalence class,*
- *initialize the set of nodes of $\mathcal{M}$ as the minimal set containing the set of milestones and closed under the operation of taking the maximal subterm,*
- *label each node t in $\mathcal{M}$ with $\Phi(t)$,*
- *connect each composed term in $\mathcal{M}$ with its maximal subterm by an edge directed from a term to its superterm,*
- *label each edge $t \to f(s_1, \ldots, t, \ldots, s_n)$ with the rule $f(\Phi(s_1)[1], \ldots, *, \ldots, \Phi(s_n)[1])$.*

Before we say how to extend the semi-skeleton to a skeleton, we want to give some intuition of how to interpret this graph. A label of an edge can be interpreted as information for the algorithm from section 3 what to choose in its nondeterministic choice to get a solution. The rule $f(c_1[1], \ldots, *, \ldots, c_n[1])$ says: take the predecessor t of the edge and first ([1] indicates first term) terms $t_1, \ldots, t_n$ from the sets $\Phi^{-1}(c_1), \ldots, \Phi^{-1}(c_n)$ and define $\Phi(f(t_1, \ldots, t, \ldots, t_n))$ as the label of the successor of the edge. Of course we have to decide what to do if conflict arises, i.e. if we are to define Φ on one term in several different ways.

Definition 4.5 *A fork of degree $k \geq 2$ is a term t such that in the semi-skeleton there are k edges connecting t with k its superterms, labeled with the same rule.*

Definition 4.6 *A skeleton of a history is a graph with the following properties:*

- *it is a forest with directed edges leading from roots to leaves*
- *it contains the semi-skeleton as a subgraph*
- *labels of edges outgoing from a single node are different*
- *labels of edges are compatible with the sentence ψ defined in the section 2, i.e. if $f^k_j(c_1[j_1], \ldots, *, \ldots, c_k[j_k])$ is a label of an edge $t \to s$, then the formula ψ with $c_1, \ldots, \Phi(t), \ldots, c_k$ substituted for $y_1, \ldots, y_k$ and $\Phi(s)$ substituted for x^k_j is satisfied*
- *if $c[j]$ is used in a label of an edge $t \to s$ then there are at least j milestones labeled by c, and each of these nodes (according to the linear ordering $\prec$) is a term lower then t*
- *all leaves are milestones*

Lemma 4.7 *If Φ is a solution of (SC), then there exists a skeleton of the history of Φ.*

Proof. If t is a fork of degree k, then we have in original history k different terms $f(s^1_1, \ldots, t, \ldots, s^1_n), \ldots, f(s^k_1, \ldots, t, \ldots, s^k_n)$, with $s^i_j \prec t$. Without loss of generality we can assume that the terms s^1_1 and s^2_1 are different. Now if the terms $s^1_1, \ldots, s^k_1$ are different, then we add $k-1$ minimal terms from $\Phi^{-1}(\Phi(s^1_1))$ to the set of milestones (remember that $\Phi(s^i_1) = \Phi(s^j_1)$, so we have at least k terms in this set) and replace the labels of edges with k different rules $f(\Phi(s^1_1)[1], \ldots, *, \ldots, \Phi(s^1_n)[1]), \ldots, f(\Phi(s^1_1)[k], \ldots, *, \ldots, \Phi(s^1_n)[1])$. If there are less then k terms among $s^1_1, \ldots, s^k_1$, then it is even better, because we need less new milestones. Consider the following examples

Example 1 t is a fork of degree 4 and we have the following four terms as nodes in the skeleton of the history of a solution Φ: $f(t, x_1, y), f(t, x_2, y), f(t, x_3, y), f(t, x_4, y)$ with $\Phi(x_1) = \Phi(x_2) = \Phi(x_3) = \Phi(x_4) = x$. Then we label edges $t \to f(t, x_j, y)$ with the rules $f(*, x[j], \Phi(y)[1])$ which (while executing step 3 of **EASY** and building a new solution Φ') we interpret as follows: take the term t (predecessor of these edges), the four minimal terms $t_1, \ldots, t_4$ from the set $\Phi^{-1}(x)$ (these are also four minimal terms from $\Phi'^{-1}(x)$), the minimal term s in $\Phi^{-1}(\Phi(y))$ and define $\Phi'(f(t, t_j, s)) = \Phi(f(t, x_j, y))$

Example 2 t is a fork of degree 4 and we have the following four terms as nodes in the skeleton of the history of solution Φ: $f(t,x_1,y_1), f(t,x_2,y_1)$, $f(t,x_1,y_2)$,$f(t,x_2,y_2)$ with $\Phi(x_1) = \Phi(x_2) = x$ and $\Phi(y_1) = \Phi(y_2) = y$. Then we have to create only two new milestones - the second minimal terms from the sets $\Phi^{-1}(x)$ and $\Phi^{-1}(y)$ and label the four edges with the rules $f(*, x[1], y[1])$, $f(*, x[2], y[1])$, $f(*, x[1], y[2])$, $f(*, x[2], y[2])$

Note that if we have a fork of degree k then we introduce at most $k-1$ new milestones, which are among the first $k-1$ terms in some equivalence classes. Moreover, these milestones are of depth less then or equal to the depth of the fork. □

Below we shall prove that a solution of a system of set constraints can be reconstructed from a skeleton of a history. In fact a solution can be reconstructed from any labeled graph satisfying all the conditions described in Definition 4.6 except containing the semi-skeleton, slightly extended to allow dealing with constants in the definition below.

Definition 4.8 *Given a skeleton $\mathcal{S}$ we say that a term t' is a collapse of a node $t \in \mathcal{S}$ if either t is a constant symbol and $t' = t$ or $t' = f(t_1, \ldots, s' \ldots, t_k)$ and there is an edge $s \to t$ labeled with $f(c_1[j_1], \ldots, *, \ldots, c_k[j_k])$, the term s' is the collapse of the node s and $t_1, \ldots, t_k$ are collapses of respective milestones. The collapsing function is a function mapping each node of the skeleton of the history to its collapse.*

Lemma 4.9 *The collapsing function is 1 to 1.*

Proof. (induction on depth of terms) It is obvious that a constant symbol cannot be a collapse of two different nodes. Let us suppose that all collapsed terms of depth less than or equal to n are collapses of at most one node and there is a term of depth $n+1$ which is a collapse of two different nodes. Without loss of generality we can assume that this is a term of the form $f(t', s')$ where t' and s' are collapses of nodes t and s respectively, with $s \prec t$. $f(t', s')$ must be produced with the two rules of the form $f(c_1[k], *)$ and $f(*, c_2[l])$, so t' must be the collapse of the k-th milestone from $\Phi^{-1}(c_1)$ which is not t (because $s \prec t$ and this k-th milestone is a term lower then s). Hence t' is a collapse of two different nodes which is impossible. □

Lemma 4.10 *If Φ is a solution of (SC) then the collapse of the skeleton of a history of Φ is a history of a solution Φ'.*

Proof. Consider the following path of choices of **EASY**:

1. choose the model with domain D consisting of the set of equivalence classes of the equivalence relation defined by Φ.

2. choose $d_1^0, \ldots, d_{i_0}^0$ as $\Phi(a_0^0), \ldots, \Phi(a_{i_0}^0)$ (note that now Φ' is defined on all the roots of the skeleton of the history).

3. if Φ' is defined on a collapse t' of a node t in the skeleton of the history and there is an edge $t \to s$ labeled with a rule $f_j^k(c_1[j_1], \ldots, *, \ldots, c_k[j_k])$ then choose the function symbol f_j^k, terms t_i as the j_i-th minimal terms from the set $\Phi'^{-1}(c_i)$ (and the term t') and d_j^k as $\Phi(s)$. If we define Φ' on s_1 before defining it on s_2 whenever $s_1 \prec s_2$ then we are sure that Φ' is already defined on the terms $t_1, \ldots t_k$ when we want to define it on $f_j^k(t_1, \ldots, t', \ldots, t_k)$. Lemma 4.9 ensures us that when we want to define Φ' on any term, it was not defined earlier.

4. since the skeleton of a history contains nodes labeled with all elements of D, then **EASY**, with the bound $2M^3$ omitted, ends with success.

From the theorem 4.1 it follows that Φ' is a solution of (SC). □

Lemma 4.11 *There are at most M^2 milestones.*

Proof. We start building the skeleton of a history (from a semi-skeleton) with M milestones. This gives for each number d at most M different terms of depth d in the semi-skeleton. Then we add new milestones only if a fork is encountered. Let t be a fork of degree k and let d denotes its depth. We have in the skeleton at least k superterms of t of depth $d+1$ and t is the only one their subterm of depth d. Then we add at most $k-1$ new terms (milestones) of depth less than or equal to d. Hence still we have at most M terms of depth d in the skeleton, which means that degrees of forks are bounded by M and the set of milestones has at most M minimal terms in each equivalence class. So the cardinality of this set is bounded by M^2. □

Lemma 4.12 *There are at most M^2 nodes of outdegree greater than one.*

Proof. In a forest the number of nodes of outdegree greater than one is less then the number of leaves. Each leaf of the skeleton of the history is a milestone. □

Lemma 4.13 *If a system of set constraints (SC) has a solution, then it has one with the history consisting of at most $2M^3$ terms.*

Proof. Assume that Φ is a solution of (SC). By lemma 4.10 we can assume that all terms in the history of Φ belong to the skeleton of this history. Suppose that there are more than $2M^3$ nodes. By lemmas 4.11 and 4.12 the number of milestones and the nodes of out-degree greater then one is bounded by $2M^2$, so there exists a path $u_1 \to \ldots \to u_m$ with $m \geq M$ such that $u_1, \ldots, u_m$ are terms of outdegree one and none of them is a milestone. So, there are two terms u_i and u_j with $1 \leq i < j \leq m$ such that $\Phi(u_i) = \Phi(u_j)$. Now, we can cut off the path $u_i \to \ldots \to u_{j-1}$ thus obtaining a smaller skeleton of a history. In fact, to build a new solution Φ' of (SC) with a lower number of terms in its history we proceed as in the proof of lemma 4.10. We have only to show that in step 3 we can assume that Φ' is not yet defined on $f_j^k(t_1, \ldots, t_k)$, and that Φ' is defined on $t_1, \ldots, t_k$. The former condition is fulfilled since different nodes collapse to different terms - if a term s' is a collapse two nodes s_1, s_2 then $s'[u_i/u_j]$ is a collapse of two nodes $s_1[u_i/u_j]$ and $s_2[u_i/u_j]$ where $s[u_i/u_j]$ denotes a term with all occurrences of u_i replaced by u_j, which contradicts Lemma 4.9. The letter condition is satisfied since Φ is defined on $s[u_i/u_j]$.

Repeating the procedure described above we can find a solution with at most $2M^3$ different terms in its history. □

As corollaries to lemma 4.13 we get

Theorem 4.14 *If a system of set constraints (SC) has a solution then* **EASY** *provides one.*

Theorem 4.15 *The problem if a system of set constraints has a solution is NEXPTIME-complete.*

5 Extensions

In [7] Bachmair, Ganzinger and Waldmann proved that for positive set constraints some extensions of the problem are also decidable. These were the satisfiability problems for set constraints without positive occurrences of projections for non-monadic function symbols and for set constraints without positive occurrences of diagonalizations. These extensions are decidable also in the case of mixed (positive and negative) set constraints.

Theorem 5.1 *The problem of satisfiability of mixed set constraints with projections where projections of non-monadic function symbols occur only negatively in positive inclusions is NEXPTIME-complete.*

Proof. The transformation described in the section 2 gives for such a system a monadic formula (note that projections for monadic function symbols are not restricted here). We can just proceed with **EASY**, the arguments given in section 4 work here without any change. □

The case with diagonalization requires more arguments. If diagonalization operators do not occur positively (this time they may occur in negative inclusions too) we get a monadic formula with equality, without positive occurrences of the equality predicate. We need an extension of Lemma 2.1 to deal with the equality.

Definition 5.2 *Let ψ be a monadic formula of quantifier depth at most q, with equality. Two interpretations I, J of ψ are similar if for each pair c_I, c_J of corresponding equivalence classes of the relations $\equiv_I$ and $\equiv_J$ either $|c_I| = |c_J|$ or both c_I and c_J have at least q elements.*

As an immediate application of Ehrenfeucht games [10] one can prove (see also [9]) that if I is a model of ψ and I is similar to J, then J is also a model of ψ. If ψ has no positive occurrences of the equality predicate, the similarity condition can be weakened: for J to be a model of ψ it suffices that for each pair of corresponding equivalence classes we have $|c_I| \leq |c_J|$.

Now, to check that a finite model I of ψ (of cardinality less then or equal to Mq) corresponds to a Herbrand model, we have to define Φ in such a way, that for each c in the domain of I we have $|\Phi^{-1}(c)| \geq |c|$. Replacing in **EASY** the condition $\Phi(\mathcal{D}) = \mathcal{D}$ (which is equivalent to $|\Phi^{-1}(c)| \geq 1$ for each $c \in \mathcal{D}$) by $|\Phi^{-1}(c)| \geq |c|$ for each $c \in \mathcal{D}$, and $2M^3$ by $2M^3q^2$, we get the algorithm solving extended systems of set constraints. The proof of correctness of this algorithm is just the same as in Section 4. The only difference is that we start building the skeleton of the history with (at most) Mq milestones, getting at most $2(Mq)^2M$ terms in the collapse of the skeleton of the history. As a corollary we get

Theorem 5.3 *The problem of satisfiability of mixed set constraints with negative diagonalizations is NEXPTIME-complete.*

Using the method presented here we can also solve the satisfiability problem for systems of set constraints with unrestricted diagonalization. The main new problem here is to make sure that in the process of construction, when $\Phi^{-1}(c)$ has reached the desired cardinality, then we can extend Φ to a solution without adding any new terms to $\Phi^{-1}(c)$.

We have two solutions. The simpler is as follows. Given a set $\mathcal{T}$ of terms, by $\Sigma(\mathcal{T})$ we denote the set of terms of the form $f(t_1, ..., t_k)$, where $f \in \Sigma$ and $t_i \in \mathcal{T}$. Using the techniques of Section 4 we can easily prove that if a set $\mathcal{T}$ and Φ have been constructed so that all finite (and bounded) classes have been saturated (i.e. have the desired number of elements), then it suffices to make sure that Φ can be extended to all terms of $\Sigma^{M+1}(\mathcal{T})$ without adding new elements to the saturated classes. Therefore the problem of consistency reduces to the problem of finding a set $\mathcal{T}$ and a function Φ on $\Sigma^{M+1}(\mathcal{T})$ as described above, which can be obtained by an extension of **EASY**. This unfortunately leads to a nondeterministic double exponential algorithm.

There is, however, a more subtle solution which gives a NEXPTIME decision procedure. The details will be presented in the full version of the paper.

Theorem 5.4 *The problem of satisfiability of mixed set constraints with projections for non-monadic function symbols occurring only negatively in positive inclusions and with unrestricted diagonalizations is decidable.*

Acknowledgements

We are indebted to Harald Ganzinger for several interesting conversations on the subject of set constraints.

The first author would like to thank Krzysiek Loryś for many hours he spend listening to consecutive version of the proof and first of all for his encouragement.

This research was partially supported by KBN grants 2 1197 91 01 and 2 1368 91 01.

This research was conducted while Witold Charatonik visited Max-Planck-Institut für Informatik (Saarbrücken). His stay there was supported by TEMPUS JEP 1941.

During the final stage of preparation of this paper Leszek Pacholski visited CRIN and INRIA-Lorraine (Nancy) supported by CRNS through a research project EURECA. He would like to thank Pierre Lescanne for his hospitality.

References

[1] W. Ackermann. *Solvable Cases of the Decision Problem.* North-Holland, Amsterdam, 1954.

[2] A. Aiken, D. Kozen, M. Vardi, and E. L. Wimmers. The complexity of set constraints. Technical Report 93-1352, Computer Science Department, Cornell University, May 1993.

[3] A. Aiken, D. Kozen, and E. L. Wimmers. Decidability of systems of set constraints with negative constraints. Technical Report 93-1362, Computer Science Department, Cornell University, June 1993.

[4] A. Aiken and B. Murphy. Implementing regular tree expressions. In *ACM Conference on Functional Programming Languages and Computer Architecture*, pages 427–447, August 1991.

[5] A. Aiken and B. Murphy. Static type inference in a dynamically typed language. In *Eighteenth Annual ACM Symposium on Principles of Programming Languages*, pages 279–290, January 1991.

[6] A. Aiken and E. L. Wimmers. Solving systems of set constraints (extended abstract). In *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 329–340, 1992.

[7] L. Bachmair, H. Ganzinger, and U. Waldmann. Set constraints are the monadic class. In *Eight Annual IEEE Symposium on Logic in Computer Science*, pages 75–83, 1993.

[8] B. Bogaert and S. Tison. Automata with equality tests. Technical Report IT 207, Laboratoire d'Informatique Fondamentale de Lille, 1991.

[9] B. Dreben and W. D. Goldfarb. *The Decision Problem. Solvable Classes of Quantificational Formulas.* Addison-Wesley Publishing Company, Inc., 1979.

[10] A. Ehrenfeucht. An application of games to the completeness problems for fromalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[11] R. Gilleron, S. Tison, and M. Tommasi. Solving systems of set constraints using tree automata. In *10th Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 665, pages 505–514. Springer-Verlag, 1993.

[12] R. Gilleron, S. Tison, and M. Tommasi. Solving systems of set constraints with negated subset relationships. In *Proceedings of the* 34^{th} *Symp. on Foundations of Computer Science*, pages 372–380, 1993. A full version *Technical report IT 247, Laboratoire d'Informatique Fondamentale de Lille.*

[13] N. Heintze and J. Jaffar. A decision procedure for a class of set constraints (extended abstract). In *Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 42–51, 1990.

[14] N. Heintze and J. Jaffar. A finite presentation theorem for approximating logic programs. In *Seventeenth Annual ACM Symposium on Principles of Programming Languages*, pages 197–209, January 1990.

[15] N. D. Jones and S. S. Muchnick. Flow analysis and optimization of lisp-like structures. In *Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 244–256, January 1979.

[16] H. R. Lewis. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences*, 21:317–353, 1980.

[17] P. Mishra and U. Reddy. Declaration-free type checking. In *Twelfth Annual ACM Symposium on the Principles of Programming Languages*, pages 7–21, 1985.

[18] J. C. Reynolds. Automatic computation of data set definitions. *Information Processing*, 68:456–461, 1969.

[19] K. Stefansson. Systems of set constraints with negative constraints are *NEXPTIME*-complete. Technical Report 93-1380, Computer Science Department, Cornell University, August 1993.

[20] J. Young and P. O'Keefe. Experience with a type evaluator. In D. Bjørner, A. P. Ershov, and N. D. Jones, editors, *Partial Evaluation and Mixed Computation*, pages 573–581. North-Holland, 1988.