ELSEVIER

Information Processing Letters 74 (2000) 27–33

Information Processing Letters

www.elsevier.com/locate/ipl

The $\exists\forall^2$ fragment of the first-order theory of atomic set constraints is Π_1^0-hard

Jean-Marc Talbot [1]

Max-Planck Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany

Received 13 July 1999; received in revised form 17 January 2000
Communicated by H. Ganzinger

Abstract

Set constraints are logical formulas involving inclusions between expressions denoting sets of trees. For atomic set constraints, set expressions do not contain any set operators. In this article we show that valid formulas of the $\exists\forall^2$ fragment of the first-order theory of atomic set constraints is Π_1^0-hard, i.e., co-recursively enumerable hard. This is proved by the encoding of a 2-register machine with a recursively enumerable-complete domain into the dual $\forall\exists^2$ fragment. This improves a result from Charatonik (1998). © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Set constraints; Theory of computation

1. Introduction

Set constraints are conjunctions of atoms and negated atoms defined as inclusions between expressions. These expressions are built over variables, a signature Σ and various set operators such as Boolean and projection operators (see [8,12] for surveys). The first-order theory of set constraints (FO_{SC}) consists of the set of formulas built on set constraints, usual logical connectives ($\vee, \wedge, \neg, \Rightarrow$) and quantifiers. Set constraints and their first-order theory are interpreted over the structure of sets of ground terms for which inclusion is interpreted as the subset relation over sets.

There exists a straightforward reduction from the monadic second-order theory of trees into FO_{SC} (even when no set operator is considered) [15]. Hence, FO_{SC} is undecidable. However, by reduction to the satisfiability problem of positive and negative set constraints, the $\exists^*$ (and by duality, the $\forall^*$) fragment of FO_{SC} is decidable when Boolean operators are considered [1,4,7] and also with both Boolean and projection operators [5].

Seynhaeve, Tommasi and Treinen have given a proof in [13] for the undecidability of the $\exists^*\forall^*$ fragment of FO_{SC} where the unique (but necessary) considered set operator is the union operator. Their proof is based on the encoding of Turing machines into that fragment by means of grid structures. Their result has been recently improved by Charatonik [3] who showed a direct encoding of the undecidable Hilbert's tenth problem into the $\exists^*\forall^*$ fragment of the first-order theory of atomic set constraints (FO_{ASC}), that is, based on set constraints without set operators.

In this article we show that one existential and two universal quantifiers are enough, i.e., that the $\exists\forall^2$ fragment of FO_{ASC} is already undecidable. The construction presented here relies on the idea used in [15] to prove the undecidability of the $\exists\forall^*$ fragment

[1] E-mail: talbot@mpi-sb.mpg.de.

PII: S0020-0190(00)00030-2

of FO_{ASC} by an encoding of the Post correspondence problem. Our method can be divided into two steps: first we consider unary-predicate logic programs and the so-called set-valued abstraction [6] consisting in:

(1) replacing memberships with inclusions,
(2) interpreting the so-obtained formulas over sets of trees.

We prove that for unary-predicate logic programs defined by Horn clauses with a linear atom in their conclusion part and all variables from the premise also occurring in the conclusion, this abstraction is exact: the Herbrand models of the program and the models of the set valued-abstraction (over the structure of sets of ground terms) coincide. At a second step, we use these particular logic programs to encode the executions of a fixed 2-register machine whose domain is recursively enumerable complete. According to the previous step, this allows us to conclude that the $\forall\exists^2$ fragment of FO_{ASC} is Σ_1^0-hard (recursively enumerable hard), and thus, by duality that the $\exists\forall^2$ fragment is Π_1^0-hard.

We argue that the idea presented here is as simple as the encoding proposed by Charatonik in [3] and improves this latter on several aspects: first, the $\exists\forall^2$ fragment of the theory considered here is smaller than the one ($\exists^*\forall^*$) addressed in [3]. This proves that adding only one outermost universal quantifier to the decidable $\exists^*$ fragment leads to undecidability. Furthermore, we consider here only *limited* atomic set constraints of the form $t \subseteq X$ whereas in [3], constraints of the form $t \subseteq t'$ (or equivalently, both $t \subseteq X$ and $X \subseteq t$) are needed. For instance, unlike atomic set constraints, conjunctions of (positive) limited atomic set constraints are always satisfiable. Moreover, when infinite signatures are considered, the related entailment problem is PSPACE-complete for atomic set constraints, but in PTIME for limited atomic set constraints (both are DEXPTIME-hard for finite signatures) [11]. Finally, we give a more precise undecidability result since we prove that the set of valid sentences of the form $\exists\forall^2\Phi$, where Φ is a quantifier-free formula of the first-order theory of atomic set constraints, is Π_1^0-hard, that is co-recursively enumerable hard.

2. Preliminaries

2.1. Generalities

We consider a ranked set of function symbols (or signature) Σ. Let V be a countable set of variables. Let $T_{\Sigma,V}$ be the set of terms built over Σ and V. For a term t in $T_{\Sigma,V}$, $V(t)$ denotes the variables from V occurring in t. A term without variable is said to be ground. T_Σ denotes the set of all ground terms. Ambiguously, T_Σ also denotes the free algebra of ground terms whose domain is the set of all ground terms and in which function symbols are interpreted as tree constructors.

Let $\mathcal{P}$ be a set of predicate symbols and A be an algebra. A relational interpretation $\mathcal{I}$ for $\mathcal{P}$ over A associates with each n-ary symbol p from $\mathcal{P}$ a set of tuples $(a_1, \ldots, a_n)$, where each a_i is an element of the carrier of A. Note that if p is unary, then $\mathcal{I}(p)$ is simply a subset of the carrier of the algebra A.

For an algebra A, a set of predicate symbols $\mathcal{P}$ and some relational interpretation $\mathcal{I}$, $\langle \mathsf{A}, \mathcal{I}\rangle$ will denote the corresponding structure. For some structure R and some valuation ν from the free variables of a formula φ into the semantics domain R, $\nu \models \varphi$ denotes that φ is true in R for the valuation ν.

Finally, we will denote $\forall\varphi$ the universal closure of the formula φ and for a set *Var* of first-order variables, $\forall_{-Var}\varphi$ will stand for $\forall x_1 \ldots \forall x_n \varphi$ if $\{x_1, \ldots, x_n\}$ are precisely the first-order variables of φ that do not occur in *Var*.

2.2. Set constraints

An atomic set constraint is an atom $t \subseteq t'$ where t, t' are terms from $T_{\Sigma,V}$. These constraints are interpreted over the structure SC whose domain is the set of all sets of finite trees built over Σ, i.e., $\wp(T_\Sigma)$. In the structure SC the inclusion symbol is interpreted as the subset relation and function symbols f from Σ are interpreted as "set of trees" constructors f^{SC} defined by

$$f^{\mathsf{SC}}(S_1, \ldots, S_m) = \left\{ f(\tau_1, \ldots, \tau_m) \mid \text{for } 1 \leqslant i \leqslant m,\ \tau_i \in S_i \right\}$$

for $S_1, \ldots, S_m \subseteq T_\Sigma$. Atoms in formulas of the first-order theory of atomic set constraints FO_{ASC} are atomic set constraints; more general formulas are built

from these atomic formulas, usual logical connectives ($\wedge, \vee, \neg$) and quantifiers. Formulas from FO_{ASC} are interpreted over SC.

2.3. Unary-predicate logic programs

A unary-predicate definite Horn clause is either an atom $t \in X$ or a quantifier-free implication $t_1 \in X_1 \wedge \cdots \wedge t_m \in X_m \Rightarrow t \in X$. The variables $X, X_1, \ldots, X_m$ are called predicate variables.[2] A logic program is a finite set of clauses. For a clause c, $V_{\mathcal{P}}(c)$ denotes the set of predicates variables occurring in c and for a logic program P,

$$V_{\mathcal{P}}(P) = \bigcup_{c \in P} V_{\mathcal{P}}(c).$$

Let I_{Σ}^{P} be the set of valuations from $V_{\mathcal{P}}(P)$ into $\wp(T_{\Sigma})$, that is the set of all relational interpretations for the predicates from $V_{\mathcal{P}}(P)$ over T_{Σ}. A Herbrand model $\mathcal{I}$ for a logic program P is an element of I_{Σ}^{P} which satisfies

$$\langle T_{\Sigma}, \mathcal{I} \rangle \models \forall \left(\bigwedge_{c \in P} c \right).$$

We define an ordering $\sqsubseteq$ over I_{Σ}^{P} as for $\mathcal{I}, \mathcal{I}'$ in I_{Σ}^{P}, $\mathcal{I} \sqsubseteq \mathcal{I}'$ iff for all X in $V_{\mathcal{P}}(P)$, $\mathcal{I}(X)$ is a subset of $\mathcal{I}'(X)$. A logic program P defines an operator $T_P : I_{\Sigma}^{P} \mapsto I_{\Sigma}^{P}$, monotonic for $\sqsubseteq$, given by: $T_P(\mathcal{I}) = \mathcal{I}'$ if for all X in $V_{\mathcal{P}}(P)$, $\tau \in \mathcal{I}'(X)$ iff there exists a ground instance $\theta(t_1) \in X_1 \wedge \cdots \wedge \theta(t_m) \in X_m \Rightarrow \theta(t) \in H$ of a clause in P such that $\tau = \theta(t)$ and for all $1 \leqslant i \leqslant m$, $\theta(t_i) \in \mathcal{I}(X_i)$.

It is well known that [9]:

- Herbrand models $\mathcal{I}$ for a logic program P are exactly pre-fix-points of the operator T_P, i.e., the valuations satisfying $T_P(\mathcal{I}) \sqsubseteq \mathcal{I}$,
- the least Herbrand model (in the sense of $\sqsubseteq$) for a logic program P is equal to the least fix-point of T_P, that is $\bigsqcup_{n \in \mathbb{N}} T_P^n(\mathcal{I}_{\emptyset})$, where
  - for all X in $V_{\mathcal{P}}(P)$, $\mathcal{I}_{\emptyset}(X) = \emptyset$,
  - for $\mathcal{I}$ in I_{Σ}^{P}, $T_P^0(\mathcal{I}) = \mathcal{I}$ and $T_P^{n+1}(\mathcal{I}) = T_P(T_P^n(\mathcal{I}))$,
  - for $\mathcal{I}, \mathcal{I}', \mathcal{I}''$ in I_{Σ}^{P}, $\mathcal{I} \sqcup \mathcal{I}' = \mathcal{I}''$ iff for all X in $V_{\mathcal{P}}(P)$, $\mathcal{I}(X) \cup \mathcal{I}'(X) = \mathcal{I}''(X)$.

[2] These predicate variables correspond to unary-predicate symbols.

3. Logic programs and set-valued abstraction

As in [6], we define the set-valued abstraction of an atom as the syntactic replacement of membership symbols by inclusion symbols: the set-valued abstraction of the atom $t \in X$ is the atomic formula defined over atomic set constraints (with terms from $T_{\Sigma, V \cup V_{\mathcal{P}}(P)}$) $t \subseteq X$ and denoted $\mathsf{Abs}(t \in X)$. We extend this function Abs to clauses as

$$\mathsf{Abs}\left(\bigwedge_i t_i \in X_i \Rightarrow t \in X \right) = \bigwedge_i \mathsf{Abs}(t_i \in X_i) \Rightarrow \mathsf{Abs}(t \in X)$$

and to logic programs as

$$\mathsf{Abs}(P) = \bigcup_{c \in P} \mathsf{Abs}(c).$$

The formulas in $\mathsf{Abs}(P)$ are interpreted over SC.

Note that for formulas in $\mathsf{Abs}(P)$, both variables from V and from $V_{\mathcal{P}}(P)$ are considered as first-order variables.

As an example, let us consider the logic program P and its set-valued abstraction $\mathsf{Abs}(P)$ given in Fig. 1. The least Herbrand model $\mathcal{I}$ for P associates with the predicate variable X the set $\{g(a,b), g(b,a)\} \cup \{g(f^n(a), f^n(a)) \mid n \in \mathbb{N}\} \cup \{g(f^n(b), f^n(b)) \mid n \in \mathbb{N}\}$. The formula $\mathsf{Abs}(P)$ abstracts P in the following sense: the least valuation Θ over SC such that

$$\mathsf{SC}, \Theta \models \forall y \bigwedge_{c \in \mathsf{Abs}(P)} c$$

associates with the variable X the set

$$\begin{aligned} &\{g(f^n(a), f^n(a)) \mid n \in \mathbb{N}\} \cup \\ &\{g(f^n(a), f^n(b)) \mid n \in \mathbb{N}\} \cup \\ &\{g(f^n(b), f^n(a)) \mid n \in \mathbb{N}\} \cup \\ &\{g(f^n(b), f^n(b)) \mid n \in \mathbb{N}\}. \end{aligned}$$

Hence, $\mathcal{I}(X) \subseteq \Theta(X)$.[3]

Definition 1. A clause $\bigwedge_i t_i \in X_i \Rightarrow t \in X$ is said to be *with linear head* if the term t is linear (i.e., each of its variables occurs at most once in t) and to be *without*

[3] We use here the fact that $\mathcal{I}$ and Θ have the same nature: both Herbrand models and valuations over SC associates with variables in $V_{\mathcal{P}}(P)$ a set of ground terms.

$g(a,a) \in X$	$g(a,a) \subseteq X$
$g(a,b) \in X$	$g(a,b) \subseteq X$
$g(b,a) \in X$	$g(b,a) \subseteq X$
$g(b,b) \in X$	$g(b,b) \subseteq X$
$g(y,y) \in X \Rightarrow g(f(y), f(y)) \in X$	$g(y,y) \subseteq X \Rightarrow g(f(y), f(y)) \subseteq X$

Fig. 1. The program P (on the left) and its set-valued abstraction $\mathsf{Abs}(P)$ (on the right).

body-only variable iff all variables occurring in the t_i's occur also in t.

A logic program is said to be with linear head (respectively without body-only variable) iff each of its clauses is with linear head (respectively without body-only variable).

Remark 1. For any linear term t and any valuation $\Theta : V(t) \mapsto \wp(T_\Sigma)$ over SC,

$$\forall \tau \in \Theta(t), \exists \sigma : V(t) \mapsto T_\Sigma,\ \sigma(t) = \tau.$$

Note that this statement does not hold for non-linear terms. As an example for $t = f(x,x)$ and $\Theta(x) = \{a,b\}$, there is no substitution σ such that the ground term $\tau = f(a,b)$ in $\Theta(f(x,x))$ is equal to $\sigma(f(x,x))$.

Theorem 1. *For any unary-predicate logic program P with linear head and no body-only variable, for all $\mathcal{I}$ in I_Σ^P,*

$$\langle T_\Sigma, \mathcal{I}\rangle \models \forall\Big(\bigwedge_{c\in P} c\Big) \quad \textit{iff}$$

$$\mathsf{SC}, \mathcal{I} \models \forall_{-V_\mathcal{P}(P)}\Big(\bigwedge_{c\in \mathsf{Abs}(P)} c\Big).$$

Proof. ($\Leftarrow$) This implication is shown in [6] for arbitrary logic programs.

($\Rightarrow$) Let us consider a clause c in P of the form $\bigwedge_i t_i \in X_i \Rightarrow t \in X$. We have to prove that for all valuations Θ such that $\Theta(Y) = \mathcal{I}(Y)$ for $Y \in V_\mathcal{P}(c)$, if for all i $\Theta(t_i) \subseteq \mathcal{I}(X_i)$ then $\Theta(t) \subseteq \mathcal{I}(X)$. If t is ground then, since c has no body-only variable, the terms t_i are ground as well. Since all terms are ground, the statement to prove is equivalent to proving that if for all $i, t_i \in \mathcal{I}(X_i)$ then $t \in \mathcal{I}(X)$. This holds by hypothesis. Let us inspect the case where t is not ground according to the valuation Θ:

(1) if there exists a variable x in t such that $\Theta(x) = \emptyset$ then by definition $\Theta(t) = \emptyset$ and thus, $\Theta(t) \subseteq \mathcal{I}(X)$;
(2) if there exists no variable x in t such that $\Theta(x) = \emptyset$ then, since the clause has no body-only variable, for all the terms t_i's, $\Theta(t_i) \neq \emptyset$.

Let us assume that there exists a valuation Θ such that for all i, $\Theta(t_i) \subseteq \mathcal{I}(X_i)$ and $\Theta(t) \not\subseteq \mathcal{I}(X)$. By definition there would exist a ground term τ such that $\tau \in \Theta(t)$ and $\tau \notin \mathcal{I}(X)$. Since t is linear, according to Remark 1, there exists a substitution σ such that $\tau = \sigma(t)$ and (since c has no body-only variable) for all $i, \sigma(t_i) \in \Theta(t_i) \subseteq \mathcal{I}(X_i)$. Thus, since $\langle T_\Sigma, \mathcal{I}\rangle \models \forall c$ holds by hypothesis, one can deduce that $\sigma(t) = \tau \in \mathcal{I}(X)$, leading to a contradiction. □

4. Encoding

In this section, we give an encoding of 2-register machines into unary-predicate logic programs with linear head and no body-only variable. [4]

Definition 2. A 2-*register machine* $\mathcal{M}$ is given by a 4-tuple (Q, q_i, q_f, Δ), where Q is a finite set of states, $q_i \in Q$ is an initial state, $q_f \in Q$ is a final state and Δ is a subset of $(Q \times \{0,1\} \times Q \times \{+,-,=\})$.

A configuration of 2-register machine $\mathcal{M}$ is a triple (q, n_0, n_1) element of $(Q \times \mathbb{N} \times \mathbb{N})$, where $\mathbb{N}$ is the set of natural numbers.

[4] A similar encoding can be found in [2].

Definition 3. A 2-register machine $\mathcal{M}$ defines a relation $\vdash_{\mathcal{M}}$ over $(Q \times \mathbb{N} \times \mathbb{N}) \times (Q \times \mathbb{N} \times \mathbb{N})$ as $(q, n_0, n_1) \vdash_{\mathcal{M}} (q', n'_0, n'_1)$ iff one of the following statements holds:

- $n'_i = n_i + 1$, $n'_j = n_j$ and $(q, i, q', +) \in \Delta$,
- $n_i > 0$, $n'_i = n_i - 1$, $n'_j = n_j$ and $(q, i, q', -) \in \Delta$,
- $n'_i = n_i = 0$, $n'_j = n_j$ and $(q, i, q', =) \in \Delta$,

where $i, j \in \{0, 1\}$ and $i \neq j$.

Let us denote by $\vdash^*_{\mathcal{M}}$ the transitive closure of $\vdash_{\mathcal{M}}$. We say that a natural number n is accepted by the 2-register machine $\mathcal{M}$ if and only if there exists two natural numbers n_1, n_2 such that $(q_i, n, 0) \vdash^*_{\mathcal{M}} (q_f, n_1, n_2)$.

A Σ^0_1-set is a recursively enumerable set. A Σ^0_1-set A is said to be *complete* if for every Σ^0_1-set B, there exists a total recursive function f such that $x \in B$ iff $f(x) \in A$. It is well known that there exists a 2-register machine $\mathcal{M}$ whose domain, i.e., the set of natural numbers n accepted by $\mathcal{M}$, is a complete Σ^0_1-set [10,14].

From now on, we assume that $\mathcal{M} = (Q, q_i, q_f, \Delta)$ is a fixed 2-register machine whose domain is a complete Σ^0_1-set and without loss of generality that $\mathcal{M}$ satisfies the two following conditions:

- the machine stops after reaching a final configuration, i.e., there is no transition in Δ of the form $(q_f, i, q', \circ)$ for $\circ$ in $\{+, -, =\}$,
- there exists a unique final configuration $(q_f, 0, 0)$.

For a set of configurations C, let us define $S_{\mathcal{M}}(C)$ the set of configurations reachable in one-step from C by $\vdash_{\mathcal{M}}$ as $\{c' \mid \exists c \in C\colon\ c \vdash_{\mathcal{M}} c'\}$. We define also $S^0_{\mathcal{M}}(C) = C$; for a strictly positive number n,

$$S^n_{\mathcal{M}}(C) = S_{\mathcal{M}}\big(S^{n-1}_{\mathcal{M}}(C)\big)$$

and

$$S^*_{\mathcal{M}}(C) = \bigcup_{n \in \mathbb{N}} S^n_{\mathcal{M}}(C).$$

Remark 2. For the 2-register machine $\mathcal{M}$, the natural number p is accepted by $\mathcal{M}$ if and only if $(q_f, 0, 0) \in S^*_{\mathcal{M}}(\{(q_i, p, 0)\})$.

Let us consider a signature Σ consisting of the states from Q and 0 as constants, a unary function symbol s and a ternary one f. Note that the signature Σ depends on the machine to be encoded (since Σ contains the set of states Q). However, this choice is only for the sake of readability. It would be possible to choose a simpler signature with only one constant and one binary function symbol. Hence, the signature used for the encoding would be totally independent from the 2-register machine.

The symbols s and 0 are used as natural number constructors. For some natural number n, we define recursively $s^n(0)$ as the term 0 if $n = 0$ and $s(s^{n-1}(0))$ otherwise. Let us define the logic program $P_{\mathcal{M}}$ as the one satisfying the six following statements:

- $f(q, x_0, x_1) \in X \Rightarrow f(q', s(x_0), x_1) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 0, q', +) \in \Delta$,
- $f(q, x_0, x_1) \in X \Rightarrow f(q', x_0, s(x_1)) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 1, q', +) \in \Delta$,
- $f(q, s(x_0), x_1) \in X \Rightarrow f(q', x_0, x_1) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 0, q', -) \in \Delta$,
- $f(q, x_0, s(x_1)) \in X \Rightarrow f(q', x_0, x_1) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 1, q', -) \in \Delta$,
- $f(q, 0, x_1) \in X \Rightarrow f(q', 0, x_1) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 0, q', =) \in \Delta$,
- $f(q, x_0, 0) \in X \Rightarrow f(q', x_0, 0) \in X$ belongs to $P_{\mathcal{M}}$ iff $(q, 1, q', =) \in \Delta$.

For a set of configurations C and a set of terms T, we write $C \leftrightharpoons T$ if the two following statements are equivalent:

(i) $(q, n_0, n_1) \in C$,
(ii) $f(q, s^{n_0}(0), s^{n_1}(0)) \in T$.

Lemma 1. *For any set of configurations C and any $\mathcal{I}$ in $I_\Sigma^{P_{\mathcal{M}}}$,*

if $C \leftrightharpoons \mathcal{I}(X)$ then $S_{\mathcal{M}}(C) \leftrightharpoons \big(T_{P_{\mathcal{M}}}(\mathcal{I})\big)(X)$.

Proof. Straightforward from the definition for the program $P_{\mathcal{M}}$ and Definition 3. □

Theorem 2. *For the 2-register machine $\mathcal{M}$ and any natural number p, the two following statements are equivalent*:

(i) *p is accepted by $\mathcal{M}$.*
(ii) *$f(q_f, 0, 0)$ belongs to the interpretation of X for every Herbrand model of $Q^p_{\mathcal{M}} = P_{\mathcal{M}} \cup \{f(q_i, s^p(0), 0) \in X\}$.*

Proof. By Remark 2, statement (i) is equivalent to

$$(q_f, 0, 0) \in S^*_{\mathcal{M}}\big(\{(q_i, p, 0)\}\big) = \bigcup_{n \in \mathbb{N}} S^n_{\mathcal{M}}\big(\{(q_i, p, 0)\}\big).$$

Moreover, from Lemma 1, one has

$$\bigcup_{n\in\mathbb{N}} S^n_{\mathcal{M}}\big(\{(q_i, p, 0)\}\big) \leftrightharpoons \bigcup_{n\in\mathbb{N}} \big(T^n_{P_{\mathcal{M}}}(\mathcal{I}_F)(X)\big),$$

where $\mathcal{I}_F$ is the element of $I_\Sigma^{P_{\mathcal{M}}}$ such that $\mathcal{I}_F(X) = \{f(q_i, s^p(0), 0)\}$. Hence, statement (i) is equivalent to $f(q_f, 0, 0) \in \bigcup_{n\in\mathbb{N}}(T^n_{P_{\mathcal{M}}}(\mathcal{I}_F)(X))$.

Statement (ii) is equivalent to the fact that $f(q_f, 0, 0)$ belongs to the interpretation of X in the least Herbrand model of $Q^p_{\mathcal{M}}$, that is

$$\left(\bigsqcup_{n\in\mathbb{N}} T^n_{Q^p_{\mathcal{M}}}(\mathcal{I}_\emptyset)\right)(X) = \bigcup_{n\in\mathbb{N}} \big(T^n_{Q^p_{\mathcal{M}}}(\mathcal{I}_\emptyset)(X)\big).$$

The proof is simply completed by stating that

$$\bigcup_{n\in\mathbb{N}} \big(T^n_{P_{\mathcal{M}}}(\mathcal{I}_F)(X)\big) = \bigcup_{n\in\mathbb{N}} \big(T^n_{Q^p_{\mathcal{M}}}(\mathcal{I}_\emptyset)(X)\big).$$

The proof for this latter comes from the fact that $(T^0_{Q^p_{\mathcal{M}}}(\mathcal{I}_\emptyset))(X) = \emptyset$ and that it is obvious to show by induction over $n > 0$ that

$$\big(T^n_{Q^p_{\mathcal{M}}}(\mathcal{I}_\emptyset)\big)(X) = \bigcup_{k<n} \big(T^k_{P_{\mathcal{M}}}(\mathcal{I}_F)\big)(X). \qquad \square$$

Let us consider now $\phi^p_{\mathcal{M}}$ and $\psi^p_{\mathcal{M}}$, sentences parameterized by the natural number p defined as:

$$\phi^p_{\mathcal{M}} \equiv \forall X\left(\left(\forall x_0 \forall x_1 \bigwedge_{c\in \mathsf{Abs}(Q^p_{\mathcal{M}})} c\right) \Rightarrow f(q_f, 0, 0) \subseteq X\right),$$

$$\psi^p_{\mathcal{M}} \equiv \exists X\left(\left(\forall x_0 \forall x_1 \bigwedge_{c\in \mathsf{Abs}(Q^p_{\mathcal{M}})} c\right) \wedge f(q_f, 0, 0) \not\subseteq X\right).$$

Corollary 1. *For the* 2*-register machine* $\mathcal{M}$ *and any natural number* p*,*

- p *is accepted by* $\mathcal{M}$ *iff* $\mathsf{SC} \models \phi^p_{\mathcal{M}}$,
- p *is not accepted by* $\mathcal{M}$ *iff* $\mathsf{SC} \models \psi^p_{\mathcal{M}}$.

Proof. Straightforward from Theorems 1 and 2. $\square$

Hence, we can deduce from Corollary 1 that the set $\Phi_{\mathcal{M}} = \{\phi^p_{\mathcal{M}} \mid p \in \mathbb{N} \wedge \mathsf{SC} \models \phi^p_{\mathcal{M}}\}$ is a complete Σ^0_1-set and that the set $\Psi_{\mathcal{M}} = \{\psi^p_{\mathcal{M}} \mid p \in \mathbb{N} \wedge \mathsf{SC} \models \psi^p_{\mathcal{M}}\}$ is a complete Π^0_1-set. Therefore, from the prenex form of the formulas $\psi^p_{\mathcal{M}}$, we have the following:

Corollary 2. *The* $\exists\forall^2$ *fragment of the first-order theory of atomic set constraints is* Π^0_1*-hard.*

5. Conclusion

In this article we have established that the $\exists\forall^2$ fragment of the first-order theory of atomic set constraints is undecidable (actually, co-recursively enumerable hard) by reduction of a 2-register machine with a complete recursively enumerable domain.

As mentioned in [3], since inclusions $t \subseteq t'$ can be defined as $t \cup t' = t$ (respectively as $t \cap t' = t$), the result proved here implies that the $\exists\forall^2$ fragment of the first-order theory of equational set constraints with union (respectively with intersection) is not decidable.

It is noticed in [6] that the least solution of the set-valued abstraction of a logic program may not be a regular set of finite trees. As an aside, we prove here that the membership problem to this set is undecidable.

Our result settles the last fragment for which decidability is still open: this is the *two-variable* fragment of the first-order theory of set constraints. However, as the $\exists^*$ and the $\forall^*$ fragment are decidable, only the fragment $\exists\forall$ is worth to be investigated. Further researches may also consider restricted-syntax formulas, such as the positive fragment of FO_{ASC}.

References

[1] A. Aiken, D. Kozen, E. Wimmers, Decidability of systems of set constraints with negative constraints, Inform. and Comput. 122 (1) (1995) 30–44.
[2] E. Boerger, E. Graedel, Y. Gurevich, The Classical Decision Problem, Springer, Berlin, 1997.
[3] W. Charatonik, An undecidable fragment of the theory of set constraints, Inform. Process. Lett. 68 (3) (1998) 147–151.
[4] W. Charatonik, L. Pacholski, Negative set constraints with equality, in: Proc. 9th IEEE Symposium on Logic in Computer Science, 1994, pp. 128–136.
[5] W. Charatonik, L. Pacholski, Set constraints with projections are in NEXPTIME, in: Proc. 35th Symposium on Foundations of Computer Science, 1994, pp. 642–653.
[6] W. Charatonik, A. Podelski, Directional type inference for logic programs, in: Proc. 5th International Static Analysis Symposium, Lecture Notes in Comput. Sci., Vol. 1503, Springer, Berlin, 1998, pp. 278–294.

[7] R. Gilleron, S. Tison, M. Tommasi, Solving systems of set constraints with negated subset relationships, in: Proc. 34th Symposium on Foundations of Computer Science, 1993, pp. 372–380.
[8] N. Heintze, J. Jaffar, Set constraints and set-based analysis, in: Proc. 2nd Workshop on Principles and Practice of Constraint Programming, Lecture Notes in Comput. Sci., Vol. 874, Springer, Berlin, 1994.
[9] J. Lloyd, Foundations of Logic Programming, Springer, Berlin, 1987.
[10] M. Minsky, Recursive unsolvability of Post's problem of "tag" and others topics in the theory of Turing machines, Ann. of Math. 74 (1961) 437–455.
[11] J. Niehren, M. Müller, J.-M. Talbot, Entailment of atomic set constraints is PSPACE-complete, in: Proc. 14th IEEE Symposium on Logic in Computer Science, 1999, pp. 285–294.
[12] L. Pacholski, A. Podelski, Set constraints: A pearl in research on constraints, in: G. Smolka (Ed.), Proc. 3rd Internat. Conference on Principles and Practice of Constraint Programming, Lecture Notes in Comput. Sci., Vol. 1330, Springer, Berlin, 1997, pp. 549–561. Tutorial.
[13] F. Seynhaeve, M. Tommasi, R. Treinen, Grid structures and undecidable constraint theories, in: Proc. 7th International Joint Conference CAAP/FASE—(TAPSOFT'97), Lecture Notes in Comput. Sci., Vol. 1214, Springer, Berlin, 1997, pp. 357–368.
[14] J. Shepherdson, H. Sturgis, Computability of recursive functions, J. ACM 10 (1963) 217–255.
[15] J.-M. Talbot, Contraintes ensemblistes définies et co-définies: Extensions et applications, Ph.D. Thesis, Université des Sciences et Technologies de Lille, 1998.