UPMAIL Technical Report No. 138
March 1997 (revised July 1997)
ISSN 1100–0686

Some Undecidable Problems Related to the Herbrand Theorem

Yuri Gurevich[*]

EECS Department
University of Michigan
Ann Arbor, MI 48109-2122
USA

gurevich@umich.edu

Margus Veanes

Computing Science Department
Uppsala University
Box 311, S-751 05 Uppsala
Sweden

margus@csd.uu.se

Abstract

We improve upon a number of recent undecidability results related to the so-called Herbrand Skeleton Problem, the Simultaneous Rigid E-Unification Problem and the prenex fragment of intuitionistic logic with equality.

[*]Partially supported by grants from NSF, ONR and the Faculty of Science and Technology of Uppsala University.

1 Introduction

We study classical first-order logic with equality but without any other relation symbols. The letters φ and ψ are reserved for quantifier-free formulas. The *signature* of a syntactic object S (a term, a set of terms, a formula, etc.) is the collection of function symbols in S augmented, in the case when S contains no constants, with a constant c. The language of S is the language of the signature of S.

Any syntactic object is *ground* if it contains no variables. A substitution is *ground* if its range is ground, and it is said to be in a given language if the terms in its range are in that language. A set of substitutions is *ground* if each member is ground. A ground substitution θ *corroborates* a formula φ (or is a *corroborator* for φ) if the formula $\varphi\theta$ is provable. Given a positive integer m, a set of m ground substitutions $\{\theta_1, \ldots, \theta_m\}$ is an *m-corroborator* for φ if the disjunction $\varphi\theta_1 \vee \cdots \vee \varphi\theta_m$ is provable. One popular form of the classical Herbrand theorem [27] is this:

> *An existential formula $\exists\vec{x}\varphi(\vec{x})$ is provable if and only if there exist a positive integer m and an m-corroborator for φ in the language of φ.*

The minimal appropriate number m will be called the *multiplicity* of φ. The multiplicity may exceed one. Here is a formula of multiplicity two suggested by Erik Palmgren in a different but similar context; we use '$\approx$' for the formal equality sign.

$$\varphi(x) \quad = \quad (c \approx 0 \Rightarrow x \approx 1) \wedge (c \approx 1 \Rightarrow x \approx 0)$$

The Herbrand theorem plays a fundamental role in automated theorem proving methods known as the *rigid variable methods* [45]. We can identify the following procedure underlying such methods. We call it the *principal procedure* of rigid variable methods. Let $\exists\vec{x}\varphi(\vec{x})$ be a closed formula that we wish to prove.

Step I Choose a positive integer m.

Step II Check if there exists an m-corroborator for φ.

Step III If Step II succeeds then $\exists\vec{x}\varphi(\vec{x})$ is provable, otherwise increase m and return to Step II.

The kernel of the principal procedure is of course Step II or:

The Herbrand Skeleton Problem
Instance: A quantifier free formula φ and a positive integer m.
Question: Is the multiplicity of φ bounded by m ?

We refer the reader to [10] for a detailed discussion of the problem. It is important to us here that the Herbrand Skeleton Problem is intimately related to the Existential Intuitionistic Problem and the Simultaneous Rigid E-Unification Problem [22]. The first of these problems is easy to formulate:

The Existential Intuitionistic Problem
Instance: An existential formula $\exists \vec{x} \varphi(\vec{x})$.
Question: Is the formula provable in intuitionistic logic?

The second requires auxiliary definitions. A *rigid equation* is an expression $E \models_\forall e$ where E is a finite set of equations and e is an equation. A ground substitution θ *solves* a rigid equation $E \models_\forall e$ if $E\theta \vdash e\theta$. A system (that is a finite set) of rigid equations is *solvable* if there is one substitution that solves all rigid equations in the system.

The Simultaneous Rigid E-Unification Problem (SREU)
Instance: A system of rigid equations.
Question: Is the system solvable?

The SREU problem has an interesting history [10]. Several false decidability claims have been published until finally it has been proved undecidable by Degtyarev and Voronkov [12, 14, 16, 17]. Later Plaisted has shown that the fragment of the SREU problem with ground left-hand sides is undecidable [36] (the left-hand side of a rigid equation $E \models_\forall e$ is E).

It is easy to see that SREU is essentially a special case of the Herbrand skeleton problem with Horn formulas and multiplicity one. It follows that the Herbrand skeleton problem is undecidable even in this very special case.

Voronkov suggested the following generalizaton of the Herbrand Skeleton Problem. Let f be a function that assignes a positive integer to every pair (k, φ) where k is a positive integer and φ a formula in our logic. Moreover, it is assumed that $k < l$ implies that $f(k, \varphi) < f(l, \varphi)$. Such a function is called a *strategy* for multiplicity. The intended meaning of the first argument of a strategy is the number of times that Step II of the principal procedure has been executed.

The Herbrand f-Skeleton Problem
Instance: A quantifier free formula φ and a positive integer k.
Question: Is the multiplicity of φ bounded by $f(k, \varphi)$?

In case $f(k, \varphi) = m$ for all φ and k, the Herbrand f-Skeleton Problem will be called the Herbrand m-Skeleton Problem, or simply the m-Skeleton problem. Thus the undecidability result of Degtyarev and Voronkov implies that the 1-Skeleton problem is undecidable. Voda and Komara have proved that, for each fixed m, the m-Skeleton problem is undecidable [42]. One important conclusion for automated theorem proving, drawn in [42], is that there is no m for which there exists an effective decision procedure that

would tell us whether m substitutions suffice to establish the provability of a given quantifier free formula.

Actually, we had hard time to understand the proof of Voda and Komara until, finally, we convinced ourselves that they have a proof. We wondered if there is a way to derive their result from the Degtyarev–Voronkov theorem. It turns out that indeed there is such a way.

In order to formulate our results, we need to recall a few definitions and give definitions of our own. Recall that a *Horn clause* is a disjunction of negated atomic formulas and at most one non-negated atomic formula; a Horn clause is often represented as a set of its disjuncts. Here we restrict attention to Horn clauses that contain exactly one non-negated atom. A *Horn formula* is a conjuction of Horn clauses. Since the equality sign is the only relation symbol in our logic, every Horn clause ψ is equivalent to an implication $E \Rightarrow s \approx t$ where E is a conjunction of equalities.

We say that a collection of formulas is *constant-disjoint* if there is no constant that occurs in two or more of the given formulas. Call a Horn formula φ *guarded* if, for every variable x that occurs in φ, there exists a clause $E \Rightarrow s \approx t$ in φ where E and s are ground and x occurs in t. Finally, call a corroborator of a disjunction φ *partisan* if it corroborates one of the disjuncts of φ. Now we are ready to formulate our first result.

Partisan Corroboration Theorem Every corroborator for a disjunction of constant-disjoint guarded Horn formulas is partisan.

This theorem is proved in Section 3. We believe it is of independent interest. It allows us an easy derivation of the Voda–Komara result from the Degtyarev–Voronkov theorem in Section 4.

In fact, we strengthen the Voda–Komara theorem in several ways. For each m, we effectively reduce SREU to the Herbrand m-Skeleton problem in such a way that the positive-arity part of the signature remains unchanged. In particular, for every m, the monadic (all function symbols are of arity ≤ 1) SREU reduces to the monadic Herbrand m-Skeleton problem; this reduction is of interest because the decidability of monadic SREU is an open problem [26].

In Section 5, we improve upon a construction in Veanes [41] and show the undecidability of a fragment of SREU with only two variables and three rigid equations with ground left-hand sides. Using this fact, we show, for each positive integer m, the undecidability of the m-Skeleton problem where each formula is a conjunction of $3m$ Horn clauses with $2m$ variables and ground negative literals; the negative literals can even be fixed.

In Section 7 we obtain some undecidability results related to the prenex fragment of intuitionistic logic with equality and proof search in intuitionistic

logic with equality. Finally, in Section 8 we describe the current status of SREU and related results and list some open problems.

2 Preliminaries

We will first establish some notation and terminology. We follow Chang and Keisler [4] regarding first order languages and structures. For the purposes of this paper it is enough to assume that the first order languages that we are dealing with are languages with equality and contain only function symbols and constants, so we will assume that from here on. We will in general use Σ, possibly with an index, to stand for a signature, i.e., Σ is a collection of function symbols with fixed arities. A function symbol of arity 0 is called a *constant*. We will always assume that Σ *contains at least one constant*.

2.1 Terms and Formulas

Terms and formulas are defined in the standard manner and are called Σ-*terms* and Σ-*formulas* respectively whenever we want be precise about the language. We refer to terms and formulas collectively as *expressions*. In the following let X be an expression or a set of expressions or a sequence of such.

We write $\Sigma(X)$ for the *signature of* X: the set of all function symbols that occur in X, $\mathcal{V}(X)$ for the set of all free variables in X and $\mathcal{C}(X)$ for the set of all constants in X. We write $X(x_1, x_2, \ldots, x_n)$ to express that $\mathcal{V}(X) \subseteq \{x_1, x_2, \ldots, x_n\}$. Let $t_1, t_2, \ldots, t_n$ be terms, then $X(t_1, t_2, \ldots, t_n)$ denotes the result of replacing each (free) occurence of x_i in X by t_i for $1 \leq i \leq n$. By a *substitution* we mean a function from variables to terms. We will use θ to denote substitutions. We write $X\theta$ for $X(\theta(x_1), \theta(x_2), \ldots, \theta(x_n))$.

We say that X is *closed* or *ground* if $\mathcal{V}(X) = \emptyset$. By $\mathcal{T}_\Sigma$ or simply $\mathcal{T}$ we denote the set of all ground Σ-terms. A substitution is called *ground* if its range consists of ground terms.

A closed formula is called a *sentence*. Since there are no relation symbols all the atomic formulas are *equations*, i.e., of the form $t \approx s$ where t and s are terms and '$\approx$' is the formal equality sign.

Atomic formulas and negated atomic formulas are called *positive* and *negative literals* respectively. A *clause* is a disjunction of literals. By a *Horn clause* we mean a clause with exactly one positive literal.[1] A Horn clause can be written as $E \Rightarrow s \approx t$ where E is a conjunction of equations, and s and t are terms. By a *Horn formula* we understand a conjunction of Horn clauses.

2.2 First Order Structures

First order structures will (in general) be denoted by capital gothic letters like $\mathfrak{A}$ and $\mathfrak{B}$ and their domains by corresponding capital roman letters like

[1] By a Horn clause we mean thus a *strict* Horn clause.

A and B respectively. A first order structure in a signature Σ is called a Σ-*structure*. For $F \in \Sigma$ we write $F^{\mathfrak{A}}$ for the interpretation of F in $\mathfrak{A}$.

If $\mathfrak{A}$ is a Σ-structure and $\Sigma' \subseteq \Sigma$ then $\mathfrak{A}{\restriction}\Sigma'$ is the Σ'-structure that is the reduction of $\mathfrak{A}$ to signature Σ'. Let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures, $\mathfrak{A}$ is a *substructure* of $\mathfrak{B}$, in symbols $\mathfrak{A} \subseteq \mathfrak{B}$, if $A \subseteq B$ and for each n-ary $F \in \Sigma$, $F^{\mathfrak{A}} = F^{\mathfrak{B}}{\restriction}A^n$.

For X a sentence or a set of sentences, $\mathfrak{A} \models X$ means that the structure $\mathfrak{A}$ is a *model of* or *satisfies* X according to Tarski's truth definition. A set of sentences is called *satisfiable* if it has a model. If X and Y are (sets of) sentences then $X \models Y$ means that Y is a *logical consequence* of X, i.e., that every model of X is a model of Y. We write $\models X$ to say that X is *valid*, i.e., true in all models.

One easily establishes, by induction on terms and formulas that, if $\mathfrak{A} \subseteq \mathfrak{B}$ then for all quantifier free sentences φ, $\mathfrak{A} \models \varphi$ iff $\mathfrak{B} \models \varphi$.

By the *free algebra over* Σ we mean the Σ-structure $\mathfrak{A}$, with domain $\mathcal{T}_\Sigma$, such that for each n-ary $F \in \Sigma$ and $t_1, \ldots, t_n \in \mathcal{T}_\Sigma$, $F^{\mathfrak{A}}(t_1, \ldots, t_n) = F(t_1, \ldots, t_n)$. We let $\mathcal{T}_\Sigma$ also stand for the free algebra over Σ.

Let E be a set of ground equations. Define the equivalence relation $=_E$ on $\mathcal{T}$ by $s =_E t$ iff $E \models s \approx t$. By $\mathcal{T}_{\Sigma/E}$ (or simply $\mathcal{T}_{/E}$) we denote the quotient of $\mathcal{T}_\Sigma$ over $=_E$. Thus, for all $s, t \in \mathcal{T}$,

$$\mathcal{T}_{/E} \models s \approx t \quad \Leftrightarrow \quad E \models s \approx t.$$

We call $\mathcal{T}_{/E}$ the *canonical model* of E.

2.3 Term Rewriting

In some cases it is convenient to consider a system of ground equations as a rewrite system. We will assume that the reader is familiar with basic notions regarding ground term rewrite systems [18]. We will only use very elementary properties. In particular, in the next section we will use Birchoff's completeness theorem for equational logic [2]. In the case of ground equations it states simply that, given a ground set of equations E and and a ground equation $s \approx t$, $E \models s \approx t$ iff s can be reduced to t by using the equations in E as rewrite rules in both directions.

In Section 5 we will use the following property of canonical (or convergent) rewrite systems (cf [18, Section 2.4]). Let R be a ground and canonical rewrite system. Then for any two ground terms t and s, the equation $t \approx s$ follows logically from R (seen as a set of equations) iff the normal forms of t and s with respect to R coincide, i.e.,

$$R \models t \approx s \quad \Leftrightarrow \quad t{\downarrow_R} = s{\downarrow_R}.$$

3 Some Logical Tools

In this section we will prove some logical properties that will be used in the next section. The main result is Theorem 5. The following lemma is

actually a consequence of Łoś-Tarski theorem.[2] We say that two (sets of) expressions X and Y are *constant-disjoint* if $\mathcal{C}(X) \cap \mathcal{C}(Y) = \emptyset$.

Lemma 1 *Let φ_i for $i \in I$, be pairwise constant-disjoint quantifier free sentences. Then $\models \bigvee_{i\in I} \varphi_i$ implies $\models \varphi_i$ for some $i \in I$.*

Proof. For $i \in I$, let $\Sigma_i = \Sigma(\varphi_i)$ and let $\Sigma = \bigcup_i \Sigma_i$. Assume by contradiction that $\not\models \varphi_i$ for all $i \in I$. Then there is (for each $i \in I$) a Σ_i-structure $\mathfrak{A}_i$ such that $\mathfrak{A}_i \models \neg\varphi_i$. Without loss of generality, take all the A_i to be pairwise disjoint.

We now construct a Σ-structure $\mathfrak{A}$ such that $\mathfrak{A}_i \subseteq \mathfrak{A}{\restriction}\Sigma_i$ for $i \in I$. First let $A = \bigcup_{i\in I} A_i$. For each $i \in I$ and constant $c \in L_i$ let $c^{\mathfrak{A}} = c^{\mathfrak{A}_i}$. For each n-ary function symbol F in Σ define $F^{\mathfrak{A}}$ as follows. For all $\vec{a} = a_1, \ldots, a_n \in A$,

$$F^{\mathfrak{A}}(\vec{a}) = \begin{cases} F^{\mathfrak{A}_i}(\vec{a}), & \text{if } \vec{a} \in A_i; \\ a_1, & \text{otherwise.} \end{cases}$$

It is clear that $\mathfrak{A}$ is well defined because of the disjointness criteria and that $\mathfrak{A}_i \subseteq \mathfrak{A}{\restriction}\Sigma_i$ for $i \in I$. Hence $\mathfrak{A}{\restriction}\Sigma_i \models \neg\varphi_i$, and thus $\mathfrak{A} \models \neg\varphi_i$ for each $i \in I$. But this contradicts that $\models \bigvee_{i\in I} \varphi_i$. ⊠

If we drop the constant-disjointness criterion in Lemma 1, then of course the lemma is false. A simple counterexample is

$$\models 0 \approx 1 \vee \neg(0 \approx 1).$$

We will state now some other obvious but useful lemmas. Lemma 2 is an easy corollary of Birkhoff's completeness theorem.

Lemma 2 *Let t and s be ground terms and let E and E' be ground sets of equations such that $\mathcal{C}(E') \cap \mathcal{C}(E, s) = \emptyset$. The following is true.*

1. *If $E' \cup E \models t \approx s$ then $E \models t \approx s$.*
2. *If $E \models t \approx s$ then $\Sigma(t) \subseteq \Sigma(E, s)$.*

Proof. Assume that $E' \cup E \models t \approx s$. By Birchoff's completeness theorem we know that s can be rewritten to t by using $E' \cup E$ as a set of rewrite rules. So there is a sequence of terms $s_0, s_1, \ldots, s_{n-1}, s_n$ where $s_0 = s$, $s_n = t$ and s_i is rewritten to s_{i+1} by using some rule in $E' \cup E$, for $0 \leq i < n$. By induction on i (for $i \leq n$) follows that $\Sigma(s_i) \subseteq \Sigma(E, s)$ and only a rule from E can be used to rewrite s_i. Part 1 follows by Birchoff's completeness theorem and part 2 follows immediately (take $E' = \emptyset$). ⊠

[2] Existential sentences are preserved under extensions.

For a finite set E of equations we will write E also for the corresponding conjunction of equations and let the context determine whether a set or a formula is meant.

Lemma 3 *Let t and s be ground terms and E' and E ground sets of equations such that E is finite and $\mathcal{C}(E') \cap \mathcal{C}(E, s) = \emptyset$. Then*

$$\mathcal{T}_{/E' \cup E} \models (E \Rightarrow t \approx s) \quad \Rightarrow \quad \models (E \Rightarrow t \approx s).$$

Proof. From $\mathcal{T}_{/E' \cup E} \models (E \Rightarrow t \approx s)$ follows immediately that $\mathcal{T}_{/E' \cup E} \models t \approx s$ and thus $E' \cup E \models t \approx s$. Hence $E \models t \approx s$ by Lemma 2, i.e., $\models (E \Rightarrow t \approx s)$. ⊠

We will use the following definitions. Let φ be a quantifier free formula and m a positive integer.

- A set of m ground substitutions Θ is an *m-corroborator* for φ if

$$\models \bigvee_{\theta \in \Theta} \varphi\theta.$$

  When $\Theta = \{\theta\}$ we say that θ is a *corroborator* for φ or *corroborates* φ.

The *m-Skeleton problem* is the problem of existence of m-corroborators for given formulas.

- For $x \in \mathcal{V}(\varphi)$, a *guard for x in φ*, if it exists, is a clause

$$E \Rightarrow t \approx s$$

  in φ such that E and s are ground and x occurs in t. We say that

$$\bigwedge_{x \in \mathcal{V}(\varphi)} \psi_x$$

  is a *guard* of φ if each ψ_x is a guard for x in φ; φ is is called *guarded* if it has a guard.

Intuitively, in the light of the second part of Lemma 2, the notion of a Horn formula being guarded is a sufficient condition to guarantee that if there is a corroborator θ for φ then the range of $\theta \restriction \mathcal{V}(\varphi)$ is $\mathcal{T}_{\Sigma(\varphi)}$, i.e., $\Sigma(\varphi\theta) = \Sigma(\varphi)$.

SREU is, by definition, the problem of existence of corroborators for Horn formulas. However, we only need to consider *guarded* Horn formulas. To see that consider a Horn formula φ; let Σ be its signature expanded with a

constant if φ has no constants and let c be a constant in Σ. Let $\varphi'(x)$ be the Horn clause $E_\Sigma \Rightarrow x \approx c$ where

$$E_\Sigma = \{\, f(c, \ldots, c) \approx c \mid f \in \Sigma \,\}.$$

Let now ψ be the guarded Horn formula

$$(\bigwedge_{x \in \mathcal{V}(\varphi)} \varphi'(x)) \wedge \varphi.$$

Clearly, ψ has a corroborator iff φ has one. Note that, for all terms t,

$$\models (E_\Sigma \Rightarrow t \approx c) \quad \Leftrightarrow \quad t \in \mathcal{T}_\Sigma.$$

Example 4 A simple example of a guarded Horn formula is

$$\begin{aligned} \psi \;=\; & (A_1 \Rightarrow c_1' \,.\, x \approx c_1) \wedge \\ & (A_2 \Rightarrow c_2' \,.\, y \approx c_2) \wedge \\ & (\Pi_1 \Rightarrow x \approx y) \wedge \\ & (\Pi_2 \Rightarrow x \approx t \,.\, y) \end{aligned}$$

where A_1, A_2, Π_1, Π_2 and t are ground, c_1, c_1', c_2 and c_2' are constants and $.$ is a binary function symbol. The guard of ψ is

$$(A_1 \Rightarrow c_1' \,.\, x \approx c_1) \wedge (A_2 \Rightarrow c_2' \,.\, y \approx c_2).$$

An example of a Horn formula with a common guard for all variables is

$$\begin{aligned} \varphi \;=\; & (A \Rightarrow x \,.\, y \approx c) \wedge \\ & (\Pi_1 \Rightarrow x \approx y) \wedge \\ & (\Pi_2 \Rightarrow x \approx t \,.\, y), \end{aligned}$$

where A, Π_1, Π_2 and t are ground and c is a constant. The guard of φ is

$$A \Rightarrow x \,.\, y \approx c.$$

Both formulas are of particular interest for us, see Section 5. □

We will use the following definition.

- A corroborator of a disjunction φ is *partisan*, if it corroborates some disjunct of φ.

The main result of this section is the following theorem.

Theorem 5 (Partisan Corroboration Theorem) *Every corroborator of a disjunction of constant-disjoint guarded Horn formulas is partisan.*

Proof. Let $\varphi = \bigvee_{i\in I} \varphi_i$ where all the φ_i's are constant-disjoint guarded Horn formulas. Let θ be a corroborator for φ. We must prove that θ corroborates φ_i for some $i \in I$.

We can assume (without loss of generality) that there exist positive integers m and n such that each φ_i has the following form:

$$\varphi_i \quad = \quad \underbrace{\bigwedge_{1\le k\le m} (E_i^k \Rightarrow s_i^k \approx t_i^k)}_{\psi_i} \quad \wedge \quad \bigwedge_{1\le k\le n} (D_i^k \Rightarrow u_i^k \approx v_i^k),$$

where ψ_i is a guard of φ_i, i.e., each E_i^k and s_i^k is ground and $\mathcal{V}(\varphi_i) = \mathcal{V}(\psi_i)$, for all $i \in I$. Let $C_i = \mathcal{C}(\varphi_i)$ for $i \in I$. We have that

$$C_i \cap C_j = \emptyset \quad (\forall i, j \in I,\ i \neq j). \tag{1}$$

Let $\Sigma = \Sigma(\varphi)$. For $i \in I$ let $\mathcal{K}_i$ denote the class of all Σ-structures that satisfy $\varphi_i\theta$, i.e,

$$\mathcal{K}_i = \{\, \Sigma\text{-structure } \mathfrak{A} \mid \mathfrak{A} \models \varphi_i\theta \,\}.$$

From the validity of $\varphi\theta$ follows that each Σ-structure belongs to some $\mathcal{K}_i$.

Let now J be any subset of I such that

$$\models \psi_i\theta \quad (\forall i \in J). \tag{2}$$

(Take for example $J = \emptyset$.) So

$$\mathcal{C}(\varphi_i\theta) = C_i \quad (\forall i \in J). \tag{3}$$

To see that, suppose (by contradiction) that $\mathcal{C}(\varphi_i\theta)$ contains some $c \notin C_i$. Clearly, c belongs to some $x\theta$ where x occurs in the guard ψ_i. By the second part of Lemma 2, every constant in $x\theta$ belongs to C_i. This gives the desired contradiction.

If $I = J$ then the theorem follows by Lemma 1. Assume that $I \neq J$. Now we prove the following statement:

$$\text{If } \not\models \varphi_i\theta \text{ for all } i \in J \text{ then } \models \psi_i\theta \text{ for some } i \in I \setminus J. \tag{4}$$

Proof of (4) Assume $\not\models \varphi_i\theta$ for all $i \in J$. Form an equation set D as follows.

- If $J = \emptyset$ let $D = \emptyset$.
- If $J \neq \emptyset$ then there is for each $i \in J$ a clause in $\varphi_i\theta$ that is not valid and by (2) this clause is not in $\psi_i\theta$. In other words, there is a mapping $f : J \to \{1, 2, \ldots, n\}$ such that

$$\not\models (D_i^{f(i)} \Rightarrow u_i^{f(i)} \approx v_i^{f(i)})\theta \quad (\forall i \in J). \tag{5}$$

  Let f be fixed and let $D = \bigcup_{i\in J} D_i^{f(i)}\theta$.

For each mapping $g : I \setminus J \to \{1, 2, \ldots, m\}$ let E_g denote the following set of equations:

$$E_g = \bigcup_{i \in I \setminus J} E_i^{g(i)},$$

and let $\mathfrak{A}_g$ be the canonical model of $D \cup E_g$, i.e.,

$$\mathfrak{A}_g = \mathcal{T}_{/E_g \cup D}.$$

We will now prove the following statement.

(*) Fix $g : I \setminus J \to \{1, 2, \ldots, m\}$. There exists $i \in I \setminus J$ such that $\mathfrak{A}_g \in \mathcal{K}_i$.

Proof of (*) Assume that (*) does not hold. (Assume also that $J \neq \emptyset$ or else (*) holds trivially.) Then $\mathfrak{A}_g \in \mathcal{K}_j$ for some $j \in J$. Fix such an appropriate j.

So $\mathfrak{A}_g$ satisfies each clause in $\varphi_j\theta$ and in particular

$$\mathfrak{A}_g \models (D_j^{f(j)} \Rightarrow u_j^{f(j)} \approx v_j^{f(j)})\theta.$$

Let $D' = D_j^{f(j)}\theta$, $u' = u_j^{f(j)}\theta$ and $v' = v_j^{f(j)}\theta$. By (3) follows that

$$\mathcal{C}(D', u', v') \subseteq C_j$$

and

$$\begin{aligned}
\mathcal{C}(E_g, D \setminus D') &= \mathcal{C}(E_g) \cup \mathcal{C}(D \setminus D') \\
&= \mathcal{C}(E_g) \cup \bigcup_{i \in J, i \neq j} \mathcal{C}(D_i^{f(i)}\theta) \\
&\subseteq \bigcup_{i \in I \setminus J} C_i \cup \bigcup_{i \in J, i \neq j} C_i \\
&= \bigcup_{i \in I, i \neq j} C_i.
\end{aligned}$$

So, by (1),

$$\mathcal{C}(D', u', v') \cap \mathcal{C}(E_g, D \setminus D') = \emptyset.$$

It follows, by Lemma 3, that

$$\models (D_j^{f(j)} \Rightarrow u_j^{f(j)} \approx v_j^{f(j)})\theta.$$

But this contradicts (5).

By using (*) we can now prove the following statement

(**) There exists $i \in I \setminus J$ such that $\models \psi_i\theta$.

Proof of (**) Assume that the claim is wrong.

Then there is for each $i \in I \setminus J$ a clause in $\psi_i\theta$ that is not valid, i.e., there is a mapping $g : I \setminus J \to \{1, 2, \ldots, m\}$ such that

$$\not\models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta) \quad (\forall i \in I \setminus J).$$

(Note that only the t_i's can be nonground.) Fix such an appropriate g.

By using (*) we know that $\mathfrak{A}_g \in \mathcal{K}_i$ for some $i \in I \setminus J$. Choose such an i. So $\mathfrak{A}_g$ satisfies each clause in $\varphi_i\theta$ and in particular

$$\mathfrak{A}_g \models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta).$$

But, by (3) and (1), $\mathcal{C}(E_i^{g(i)}, s_i^{g(i)}) \cap \mathcal{C}(E_g \setminus E_i^{g(i)}, D) = \emptyset$. Hence, by Lemma 3,

$$\models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta).$$

So we have contradiction.

This proves statement (4). Let now J be the *maximal* subset of I such that (2) holds. In other words, for all $i \in I \setminus J$, $\not\models \psi_i\theta$. By the contrapositive of (4) we conclude that for some $i \in J$, $\models \varphi_i\theta$ and the theorem follows. $\boxtimes$

Remark Theorem 5, as well as its proof, remain correct if the disjunction is infinite. We will not use this generalization.

The following example illustrates why the conditions of being constant-disjoint and guarded are important and cannot in general be discarded. In each case there is a counterexample to the theorem.

Example 6 Let us first consider an example where the disjuncts are guarded but not constant-disjoint. Let $\varphi(x)$ be the following guarded Horn formula:

$$(c \approx 0 \Rightarrow x \approx 1) \wedge (c \approx 1 \Rightarrow x \approx 0)$$

where c, 0 and 1 are contants, and let $\varphi_1 = \varphi(x_1)$, $\varphi_0 = \varphi(x_0)$ and $\psi = \varphi_1 \vee \varphi_0$ where x_1 and x_0 are distinct variables. Consider now any ground substitution θ such that $\theta(x_1) = 1$ and $\theta(x_0) = 0$. It is easy to show by case analysis that θ corroborates ψ, i.e., that

$$\begin{aligned} \models \quad & ((c \approx 0 \Rightarrow 1 \approx 1) \wedge (c \approx 1 \Rightarrow 1 \approx 0)) \vee \\ & ((c \approx 0 \Rightarrow 0 \approx 1) \wedge (c \approx 1 \Rightarrow 0 \approx 0)). \end{aligned}$$

However, θ corroborates neither φ_1 nor φ_0.

Let us now consider the case when constant-disjointness is not violated but the disjuncts are not guarded. Let $\varphi_1(y, x_1, y_1)$ be the formula

$$((y \approx 0 \Rightarrow x_1 \approx y_1) \wedge (y \approx y_1 \Rightarrow x_1 \approx 0))$$

and let $\varphi_0(x_0, y_0)$ be the formula

$$((c \approx y_0 \Rightarrow x_0 \approx 1) \wedge (c \approx 1 \Rightarrow x_0 \approx y_0))$$

where c, 0 and 1 are constants and x_1, x_0, y_1, y_0, y distinct variables. Let $\psi = \varphi_1 \vee \varphi_0$. Let θ be a ground substitution such that $\theta(x_1) = 1$, $\theta(x_0) = 0$, $\theta(y) = c$, $\theta(y_1) = 1$ and $\theta(y_0) = 0$. Then $\models \psi\theta$ but $\not\models \varphi_1\theta$ and $\not\models \varphi_0\theta$ (the situation is exaclty the same as in the previous case). □

4 Reduction of 1-Skelton Problem to n-Skeleton Problem

The 1-Skeleton problem is undecidable. This follows from the undecidability of SREU by Degtyarev and Voronkov [14, 17]. We can formulate their result in the current setting as follows (cf [17, Theorem 1]).

Theorem 7 (Degtyarev–Voronkov) *The 1-Skeleton problem of guarded Horn formulas is undecidable.*

Under certain restrictions on the language and the structure of formulas, the 1-Skeleton problem becomes decidable. It is known, however, that it is already undecidable in the presence of one binary function symbol (in addition to constants); moreover, two variables suffice for undecidability [41]. For a summary over what is known to be decidable or undecidable under various restrictions see Section 8.

For technical reasons it will be convenient to assume in the following that we have a fixed signature Σ with $\{c_1, c_2, \ldots\}$ as the set of distinct constants in it. Σ may also have other function symbols of arity ≥ 1. Let us also be precise about the variables that we allow in Σ-expressions, by assuming that all variables come from the collection $\{x_1, x_2, \ldots\}$.

For each natural number n, constant c and variable x, let $c^{(n)}$ denote a new constant and let $x^{(n)}$ denote a new variable. We define by induction on any Σ-expression X the corresponding expression $X^{(n)}$ as the one obtained from X by replacing in it each variable x with $x^{(n)}$ and each constant c with $c^{(n)}$. For any substitution θ of Σ-variables with Σ-terms we let $\theta^{(n)}$ denote a substitution that takes the variable $x^{(n)}$ t the term $\theta(x)^{(n)}$. So, for any Σ-expression X and natural number n,

$$(X\theta)^{(n)} = X^{(n)}\theta^{(n)}.$$

The following property is immediate. For any Σ-sentence φ and natural number n,

$$\models \varphi \quad \Leftrightarrow \quad \models \varphi^{(n)}.$$

Theorem 8 *Let φ be a guarded Horn formula and n a positive integer. Then φ has a corroborator iff $\bigwedge_{i=1}^{n} \varphi^{(i)}$ has an n-corroborator.*

Proof. The '$\Rightarrow$' direction is trivial. We prove the '$\Leftarrow$' direction as follows. Let $I = \{1, 2, \ldots, n\}$ and let ψ be the formula $\bigwedge_{i \in I} \varphi^{(i)}$. Assume that ψ has an n-corroborator $\{\, \theta_i \mid i \in I \,\}$. So

$$\models \bigvee_{i \in I} (\bigwedge_{j \in I} \varphi^{(j)} \theta_i).$$

By the distributive law this is equivalent to

$$\models \bigwedge_{f : I \to I} (\bigvee_{i \in I} \varphi^{(f(i))} \theta_i).$$

From this follows in particular that

$$\models \bigvee_{i \in I} \varphi^{(i)} \theta_i.$$

Let $X_i = \mathcal{V}(\varphi^{(i)})$ for $i \in I$. Since all the X_i's are pairwise disjoint we can let θ' be a substitution such that $\theta' \restriction X_i = \theta_i \restriction X_i$ for $i \in I$, and it follows that

$$\models \bigvee_{i \in I} \varphi^{(i)} \theta'.$$

By Theorem 5 follows now that $\models \varphi^{(i)} \theta'$ for some $i \in I$. Fix such an appropriate i. But then, by Lemma 2, the range of $\theta' \restriction X_i$ is $\mathcal{T}_{\Sigma(\varphi^{(i)})}$, and thus there is a substitution θ with range $\mathcal{T}_\Sigma$ such that $\theta^{(i)} \restriction X_i = \theta' \restriction X_i$. Hence $\models \varphi^{(i)} \theta^{(i)}$ and so $\models \varphi\theta$ by above. ⊠

Corollary 9 (Voda–Komara) *For all $n \geq 1$, n-Skeleton problem of guarded Horn formulas is undecidable.*

Proof. The reduction in Theorem 8 is trivially effective. So, if we had a decision procedure (for some n) for finding n-corroborators, we could use it to find corroborators, but this would contradict Theorem 7. ⊠

Assume that we are using an automated theorem proving method that is based on the Herbrand theorem. Roughly, this involves a search for terms, for a given bound m on multiplicity. Corollary 9 (Voda and Komara [42]) tells us that there is no m for which we could effectively decide when to stop our search for such terms in case they do not exist.

By using the fact that SREU is undecidable already with ground left-hand sides [36], (i.e., variables occur only in positive literals in the corresponding Horn formulas) and two variables [40, 41] we obtain a sharper version of the above corollary:

Corollary 10 *For all $n \geq 1$, n-Skeleton problem of guarded Horn formulas is undecidable already if there are $2n$ variables and all variables occur in positive literals.*

The decidability of monadic SREU is currently one of the problems related to SREU that is still open [26]. An effectively equivalent problem is the decidability of the prenex fragment of intuitionistic logic with equality with unary function symbols [15]. Some evidence speaks in favour of that the problem is decidable although with very high computational complexity (e.g., many subcases are decidable, see Section 8). From Theorem 8 follows that:

Corollary 11 *If the 1-Skeleton problem is undecidable in the monadic case then so is the n-Skeleton problem for $n > 1$, or equivalenty, if the n-Skeleton problem is decidable in the monadic case for some $n > 1$ then so is the 1-Skeleton problem.*

5 Undecidability of SREU: Minimal case

W show that *three* rigid equations with *ground left-hand sides* and *two variables* in a signature with one binary function symbol an no other nonconstant function symbols, already imply undecidability. In fact, we give a uniform representation of all the recursively enumerable sets by using just three rigid equations with these properties. As a corollary we get that the undecidability of SREU holds already in very restricted cases. We generalize the construction in Veanes [41] and improve the lower bound on the number of rigid equations from four to three by using finite tree automata techniques. We then use this result to improve the undecidability result of the n-Skeleton problem.

The main idea behind our proof is based on a technique that was used by Plaisted [36] in a similar context, who called the technique *shifted pairing*. The idea is to express repetition explicitly by a sequence of strings (like IDs of a TM). The first string of the sequence fulfills some initial conditions, the last string some final conditions and another sequence is used to check that the consequtive strings of the first sequence satisfy some relationship (like validity of a computation step).

A similar technique was used already by Goldfarb in the proof of the undecidability of second-order unification [24], which is by reduction of Hilbert's tenth problem, and later, adopted from that proof, also in a proof of the undecidability of SREU by Degtyarev and Voronkov [16], which is also by reduction of Hilbert's tenth problem. In this proof the key point is to explicitly represent the "history of a multiplication process".

We note also that shifted pairing bears certain similarities to the technique that is used to prove that any recursively enumerable set of strings is

given by the intersection of two (deterministic) context free languages [28, Lemma 8.6].

Finite Tree Automata Finite tree automata, or simply tree automata from here on, are a generalization of classical automata. Tree automata were introduced, independently, in Doner [19] and Thatcher and Wright [39]. The main motivation was to obtain decidability results for the weak monadic second-order logic of the binary tree. Here we adopt the following definition of tree automata, based on rewrite rules [5, 6].

- A *tree automaton* or *TA* A is a quadruple (Q, Σ, R, F) where
  - Q is a finite set of constants called *states*,
  - Σ is a *signature* that is disjoint from Q,
  - R is a set of *rules* of the form $f(q_1, \ldots, q_n) \to q$, where $f \in \Sigma$ has arity $n \geq 0$ and $q, q_1, \ldots, q_n \in Q$,
  - $F \subseteq Q$ is the set of *final states*.

  A is called a *deterministic* TA or DTA if there are no two different rules in R with the same left-hand side.

Note that if A is deterministic then R is a reduced set of ground rewrite rules and thus canonical [37]. Tree automata as defined above are usually also called *bottom-up* tree automata. Acceptance for tree automata or recognizability is defined as follows.

- The set of terms *recognized* by a TA $A = (Q, \Sigma, R, F)$ is the set

$$T(A) = \{\, \tau \in \mathcal{T}_\Sigma \mid (\exists q \in F)\ \tau \xrightarrow{*}_R q \,\}.$$

  A set of terms is called *recognizable* if it is recognized by some TA.

5.1 Main Idea

We consider a fixed Turing machine

$$M = (Q_M, \Sigma_{\text{in}}, \Sigma_{\text{tape}}, \delta, q_0, \text{ƀ}, \{q_{\text{acc}}\}),$$

and assume, without loss of generality, that the final ID of M is simply q_{acc} i.e., the tape is always empty when M enters the final state, and that $q_0 \neq q_{\text{acc}}$. Let also v be a string over the input alphabet of M. We effectively construct a system $S_v^M(x, y)$ of three rigid equations:

$$S_v^M(x, y) \quad = \quad \{\, S_0(x, y),\ S_1(x, y),\ S_2(x, y) \,\}$$

where

$$\begin{aligned} S_0(x,y) &= E_0 \models_\forall x \,.\, y \approx c_0, \\ S_1(x,y) &= \Pi_1 \models_\forall x \approx y, \\ S_2(x,y) &= \Pi_2 \models_\forall x \approx t_v \,.\, y \end{aligned}$$

where E_0, Π_1 and Π_2 are ground, c_0 is a constant, '.' is the only nonconstant function symbol in the system and t_v is a ground term that represents the initial ID of M with input string v. We prove that M accepts v iff S_v^M is solvable. This establishes the undecidability result because all the steps in the construction are effective.

The main idea behind the rigid equations is roughly as follows. Assume that there is a substitution θ that solves the system.

- From θ being a solution of $S_0(x,y)$, it follows that
  - $x\theta$ represents a sequence
    $$(v_0, v_1, \ldots, v_m)$$
    of IDs of M, and v_m is the final ID of M, and
  - $y\theta$ represents a sequence
    $$((w_0, w_0^+), (w_1, w_1^+), \ldots, (w_n, w_n^+))$$
    of *moves* of M, i.e., $w_i \vdash_M w_i^+$ for $0 \leq i \leq n$.
- From θ being a solution of $S_1(x,y)$ it follows that $n = m$ and $v_i = w_i$ for $0 \leq i \leq m$.
- And finally, from θ being a solution of $S_2(x,y)$ it follows that $v_0 = v$ and $v_i = w_{i-1}^+$ for $1 \leq i \leq m$.

The combination of the last two points is the so-called "shifted pairing" technique. This is illustrated by Figure 1. The outcome of this shifted pairing is that $x\theta$ is a valid computation of M with input v, and thus M accepts v. Conversely, if M accepts v then it is easy to construct a solution of the system. We now give a formal construction of the above idea.

5.2 Words and Trains

Words are certain terms that we choose to represent strings with, and trains are certain terms that we choose to represent sequences of strings with. We use the letters v and w to stand for strings of constants. Let . be a binary function symbol. We write it in infix notation and assume that it associates to the right. For example $t_1 \,.\, t_2 \,.\, t_3$ stands for the term $.(t_1, .(t_2, t_3))$.

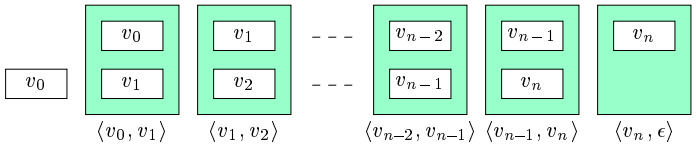


Figure 1: $(\langle v_0, v_1\rangle, \langle v_1, v_2\rangle, \ldots, \langle v_n, \epsilon\rangle)$ is a "shifted pairing" of $(v_0, v_1, \ldots, v_n)$.

▶ We say that a (ground) term t is a *c-word* if it has the form

$$a_1 \centerdot a_2 \centerdot \cdots \centerdot a_n \centerdot c$$

for some $n \geq 0$ where each a_i and c is a constant. A *word* is a c-word for some constant c.

We use the following convenient shorthand notation for words. Let t be the word $a_1 \centerdot a_2 \centerdot \cdots \centerdot a_n \centerdot c$ and v the string $a_1 a_2 \cdots a_n$. We write $v \centerdot c$ for t and say that t *represents* v.

▶ *A* term t is called a *c-train* if it has the form

$$t_1 \centerdot t_2 \centerdot \cdots \centerdot t_n \centerdot c$$

for some $n \geq 0$ where each t_i is a word and c is a constant. If $n = 0$ then t is said to be *empty*. The t_i's are called the *words of* t. A *train* is a c-train for some constant c.

By the *pattern* of a train

$$(v_1 \centerdot c_1) \centerdot (v_2 \centerdot c_2) \centerdot \cdots \centerdot (v_n \centerdot c_n) \centerdot c$$

we mean the string $c_1 c_2 \cdots c_n$. Let $\mathcal{V} = \{V_i\}_{i \in I}$ be a finite family of regular sets of strings over a finite set Σ of constants, where I is a set of constants disjoint from Σ. Let U be a regular set of strings over I and let c be a constant not in Σ or I.

▶ We let $\mathrm{Tn}(\mathcal{V}, U, c)$ denote the set of all c-trains t such that the pattern of t is in U and, for $i \in I$, each i-word of t represents a string in V_i.

Example 12 Consider the set $\mathrm{Tn}(\{V_a, V_b, V_c\}, ab^*c, \Lambda)$. This is the set of all Λ-trains t such that the first word of t is an a-word representing a string in V_a, the last word of t is a c-word representing a string in V_c and the middle ones (if any) are b-words representing strings in V_b. □

We say that a set of trains *has a regular pattern* if it is equal to some set $\mathrm{Tn}(\mathcal{V}, U, c)$ with $\mathcal{V}$, U and c as above. The main result of this section is the following theorem.

Theorem 13 (Train Theorem) *Any set of trains with a regular pattern is recognizable and a DTA that recognizes this set can be obtained effectively.*

As we shall see, the construction of the rigid equation S_0 follows easily from the Train Theorem and some basic properties of tree automata. We believe that this theorem is of independent interest. For example, several theorems that are used in a similar context in Plaisted [36, Theorems 8.2–8.11], can be stated as corollaries of Theorem 13. Before we prove the theorem we state the following simple lemma. This lemma follows from the wellknown fact that all regular sets of strings are recognizable (cf [23]), assuming an appropriate representation of strings.[3] For any string v, we write v^{r} for v *in reverse* and for a set of strings V we let $V^{\mathrm{r}} = \{\, v^{\mathrm{r}} \mid v \in V \,\}$.

Lemma 14 *Let V be a regular set of strings over a set Σ of constants and c a constant not in Σ. Then $\{\, v \mathbin{.} c \mid v \in V \,\}$ is recognizable and a DTA is obtained effectively from V.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts the reverse of V, or V^{r}, (clearly M exists, cf [28, p 281]). For each $a \in \Sigma$ let $\tilde{a}$ be a new state. Let A be the DTA (Q_A, Γ, R_A, F_A) where

$$\begin{aligned} Q_A &= Q \cup \{\, \tilde{a} \mid a \in \Sigma \,\}, \\ \Gamma &= \Sigma \cup \{\mathbin{.}, c\}, \\ R_A &= \{\, \tilde{a} \mathbin{.} q \to p \mid \delta(q,a) = p \,\} \cup \{\, a \to \tilde{a} \mid a \in \Sigma \,\} \cup \{c \to q_0\}, \\ F_A &= F. \end{aligned}$$

We must prove that, for all $t \in \mathcal{T}_\Gamma$,

$$t \xrightarrow{*}_{R_A} q \text{ for some } q \in F \quad \Leftrightarrow \quad t = v \mathbin{.} c \text{ for some } v \in L(M)^{\mathrm{r}}.$$

Let us consider the direction '$\Leftarrow$' first. So assume that

$$v = a_{n-1}a_{n-2}\cdots a_0 \in L(M)^{\mathrm{r}},$$

i.e, $a_0 \cdots a_{n-2}a_{n-1} \in L(M)$. So, there exist $q_1, q_2 \ldots, q_n \in Q$, such that $q_n \in F$ and the following holds:

$$\delta(q_0, a_0) = q_1, \ldots, \delta(q_{n-2}, a_{n-2}) = q_{n-1},\ \delta(q_{n-1}, a_{n-1}) = q_n.$$

But then, by the definition of R_A, we can construct the following reduction:

$$\begin{aligned} v \mathbin{.} c = a_{n-1}a_{n-2}\cdots a_0 \mathbin{.} c &\xrightarrow{*} \tilde{a}_{n-1}\tilde{a}_{n-2}\cdots\tilde{a}_1\tilde{a}_0 \mathbin{.} q_0 \\ &\longrightarrow \tilde{a}_{n-1}\tilde{a}_{n-2}\cdots\tilde{a}_1 \mathbin{.} q_1 \\ &\xrightarrow{*} \tilde{a}_{n-1} \mathbin{.} q_{n-1} \\ &\longrightarrow q_n \in F, \end{aligned}$$

[3] Traditionally a string $a_1a_2\cdots a_n$ is represented by a term $a_n(\cdots a_2(a_1(q_0)))$, i.e., the symbols of the alphabet are treated as unary function symbols, and the term is written using the reverse notation $q_0a_1a_2\cdots a_n$.

which shows that $v \,.\, c \in T(A)$. The direction '$\Rightarrow$' follows also easily. First note that any term t in $\mathcal{T}_\Gamma$ that reduces to a final state q with respect to R_A must be a c-word that represents some string v over Σ. From the definition of R_A follows then, like above, that v must be in V. $\boxtimes$

We now prove the Train Theorem.

Proof. Let $\mathcal{V}$, Σ, U, I, and c be like above. For each $i \in I$, let $\Sigma_i = \Sigma \cup \{\,.\,, i\}$ and let $A_i = (Q_i, \Sigma_i, R_i, F_i)$ be a DTA given by Lemma 14 such that

$$T(A_i) = \{\, v \,.\, i \mid v \in V_i \,\}.$$

Let $\Sigma_c = I \cup \{\,.\,, c\}$ and let $A_c = (Q_c, \Sigma_c, R_c, F_c)$ be a DTA given by Lemma 14 such that

$$T(A_c) = \{\, u \,.\, c \mid u \in U \,\}.$$

Assume, without loss of generality, that all the DTAs have mutually disjoint sets of states, except for the states $\tilde{a}$ for $a \in \Sigma$ that are the same in all the A_i's for $i \in I$. In fact, one can think of any constant $a \in \Sigma$ and the corresponding state $\tilde{a}$ *as being the same element*.

Let now R' be the set of rules obtained from R_c by relpacing, for all $i \in I$, each rule $\tilde{\imath} \,.\, p_1 \to p_2$ in it with the set of rules $\{\, q \,.\, p_1 \to p_2 \mid q \in F_i \,\}$, and discarding the rule $i \to \tilde{\imath}$. Let now R be the following set of rules:

$$R = \bigcup_{i \in I} R_i \cup R'.$$

Note that R is a reduced set of rewrite rules due to the disjointness assumptions and the assumption that the states $\tilde{a}$ for $a \in \Sigma$ are the same in all the DTAs. We are now ready to define A as the DTA (Q, Γ, R, F_c) where $\Gamma = \Sigma \cup I \cup \{\,.\,, c\}$ and

$$Q = \bigcup_{i \in I} Q_i \cup (Q_c \setminus \{\, \tilde{\imath} \mid i \in I \,\}).$$

We can now prove that

$$T(A) = \mathrm{Tn}(\mathcal{V}, U, c).$$

Let use consider the direction '$\subseteq$' first. Assume that $t \in T(A)$, i.e., t reduces to some state q in F_c via the rules in R. This reduction is only possible if it has (in principle) the following form:[4]

$$t \xrightarrow{*}_R q_1 q_2 \cdots q_n \,.\, c \xrightarrow{*}_{R'} q.$$

[4] A formal argument can be given by using induction and proving some lemmas first [40, Chapter 3].

where each q_k is in F_{i_k} for some $i_k \in I$. Furthermore, by definition of R' and A_c, we know that $i_1 i_2 \cdots i_n \in U$. The first part of the reduction is possible only if

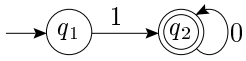
$$t = t_1 \,.\, t_2 \,.\, \cdots \,.\, t_n \,.\, c,$$

where each t_k reduces to q_k. Note that, due to the disjointness properties of the DTAs, only the rules in R_{i_k} can be used in the reduction $t_k \xrightarrow{*} q_k$, and thus $t_k \in T(A_{i_k})$. Hence each t_k has the form $v \,.\, i_k$ for some $v \in V_{i_k}$, and the pattern of t is $i_1 i_2 \cdots i_n$, which we know is in U. This proves that $t \in \mathrm{Tn}(\mathcal{V}, U, c)$.

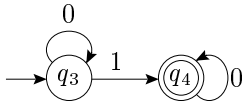
Let us now consider the direction '$\supseteq$'. So assume that $t = t_1 \,.\, t_2 \,.\, \cdots \,.\, t_n \,.\, c$ where each t_k is in $T(A_{i_k})$ for some $i_k \in I$ and $i_1 i_2 \cdots i_n \in U$. It follows that each t_k reduces with R_{i_k} to some $q_k \in F_{i_k}$ and thus t reduces to $q_1 q_2 \cdots q_n \,.\, c$. By definition of R', $q_1 q_2 \cdots q_n \,.\, c$ reduces to some $q \in F_c$. It follows that $t \xrightarrow{*}_R q$ for some $q \in F_c$ and thus $t \in T(A)$. ⊠

The following example illustrates the construction that is used in the proof of the Train Theorem.

Example 15 Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, $I = \{\mathtt{a}, \mathtt{b}\}$ and let Λ be a new constant. Let $\mathcal{V} = \{V_i\}_{i \in I}$ where $V_{\mathtt{a}} = \mathtt{0}^*\mathtt{1}$ and $V_{\mathtt{b}} = \mathtt{0}^*\mathtt{1}\mathtt{0}^*$. Let $U = \mathtt{b}\mathtt{a}\mathtt{b}^*\mathtt{a}$. We construct a DTA that recognizes the set $\mathrm{Tn}(\mathcal{V}, U, \Lambda)$. Consider the following transition diagrams of a DFA for ${V_{\mathtt{a}}}^{\mathrm{r}}$:



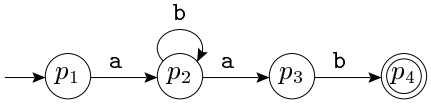
and of a DFA for ${V_{\mathtt{b}}}^{\mathrm{r}}$:



By following the construction in Lemma 14 we get that the rules of $A_{\mathtt{a}}$ and $A_{\mathtt{b}}$ are as follows:

$$\begin{aligned} R_{\mathtt{a}} &= \{\mathtt{1} \to \tilde{\mathtt{1}},\ \mathtt{0} \to \tilde{\mathtt{0}},\ \mathtt{a} \to q_1,\ \tilde{\mathtt{1}} \,.\, q_1 \to q_2,\ \tilde{\mathtt{0}} \,.\, q_2 \to q_2\}, \\ R_{\mathtt{b}} &= \{\mathtt{1} \to \tilde{\mathtt{1}},\ \mathtt{0} \to \tilde{\mathtt{0}},\ \mathtt{b} \to q_3,\ \tilde{\mathtt{0}} \,.\, q_3 \to q_3,\ \tilde{\mathtt{1}} \,.\, q_3 \to q_4,\ \tilde{\mathtt{0}} \,.\, q_4 \to q_4\}. \end{aligned}$$

For the set U^{r} we can consider a DFA with the following transition diagram:



From this we can extract the DTA A_Λ with the following set of rules:

$$\begin{aligned} R_\Lambda \;=\; & \{\mathtt{a} \to \tilde{\mathtt{a}},\ \mathtt{b} \to \tilde{\mathtt{b}},\ \Lambda \to p_1, \\ & \tilde{\mathtt{a}}\,.\,p_1 \to p_2,\ \tilde{\mathtt{b}}\,.\,p_2 \to p_2,\ \tilde{\mathtt{a}}\,.\,p_2 \to p_3,\ \tilde{\mathtt{b}}\,.\,p_3 \to p_4\}. \end{aligned}$$

Now, following the construction in the Train Theorem, we get that the DTA A has the following set of rules. First, a set R' is constructed by removing the first two rules in R_Λ and replacing $\tilde{\mathtt{a}}$ and $\tilde{\mathtt{b}}$ with q_2 and q_4, respectively. Second, R is taken as the union of $R_{\mathtt{a}}$, $R_{\mathtt{b}}$ and R' . So R is the following set of rules:

$$\begin{aligned} R \;=\; & \{\mathtt{1} \to \tilde{\mathtt{1}},\ \mathtt{0} \to \tilde{\mathtt{0}},\ \mathtt{a} \to q_1,\ \tilde{\mathtt{1}}\,.\,q_1 \to q_2,\ \tilde{\mathtt{0}}\,.\,q_2 \to q_2\} \cup \\ & \{\mathtt{b} \to q_3,\ \tilde{\mathtt{0}}\,.\,q_3 \to q_3,\ \tilde{\mathtt{1}}\,.\,q_3 \to q_4,\ \tilde{\mathtt{0}}\,.\,q_4 \to q_4\} \cup \\ & \{\Lambda \to p_1,\ q_2\,.\,p_1 \to p_2,\ q_4\,.\,p_2 \to p_2,\ q_2\,.\,p_2 \to p_3,\ q_4\,.\,p_3 \to p_4\}. \end{aligned}$$

Let us consider a reduction in R. Let us write a Λ-train $t_1\,.\,t_2\,.\cdots.\,t_n\,.\,\Lambda$ as $[t_1, t_2, \ldots, t_n]$. Take for example

$$t = [\mathtt{010\,.\,b},\quad \mathtt{001\,.\,a},\quad \mathtt{1\,.\,b},\quad \mathtt{01\,.\,a}].$$

The pattern of t is $\mathtt{baba}$ which is in U. Let us see how t reduces to p_4.

$$\begin{aligned} t \;&\xrightarrow{*}_R\; [\tilde{\mathtt{0}}\tilde{\mathtt{1}}\tilde{\mathtt{0}}\,.\,q_3,\quad \tilde{\mathtt{0}}\tilde{\mathtt{0}}\tilde{\mathtt{1}}\,.\,q_1,\quad \tilde{\mathtt{1}}\,.\,q_3,\quad \tilde{\mathtt{0}}\tilde{\mathtt{1}}\,.\,q_1] \\ &\xrightarrow{*}_R\; [\tilde{\mathtt{0}}\tilde{\mathtt{1}}\,.\,q_3,\quad \tilde{\mathtt{0}}\tilde{\mathtt{0}}\,.\,q_2,\quad q_4,\quad \tilde{\mathtt{0}}\,.\,q_2] \\ &\xrightarrow{*}_R\; [\tilde{\mathtt{0}}\,.\,q_4,\quad \tilde{\mathtt{0}}\,.\,q_2,\quad q_4,\quad q_2] \\ &\xrightarrow{*}_R\; [q_4,\quad q_2,\quad q_4,\quad q_2] \\ &\longrightarrow_{R'}\; q_4q_2q_4q_2\,.\,p_1 \\ &\xrightarrow{*}_{R'}\; q_4. \end{aligned}$$

□

5.3 Representing IDs and Moves

We show how to construct the rigid equation $S_0(x, y)$. Our main tool in doing so is the Train Theorem. We use also the following simple observation, that relates rigid E-unification with recognizability. Let us, for simplicity, consider a set of rules also as a set of equations.

Lemma 16 *Let $A = (Q, \Sigma, R, \{q\})$ be a DTA. Then, for all θ with range $\mathcal{T}_\Sigma$, θ solves $R \Vdash_\forall x \approx q$ iff $x\theta \in T(A)$.*

Proof. Since R is a canonical rewrite system and q is irreducible in R, we have (for all ground θ) that $R \models x\theta \approx q$ iff $x\theta \xrightarrow{*}_R q$. But for θ with range $\mathcal{T}_\Sigma$, by definition of recognizability, $x\theta \in T(A)$ iff $x\theta \xrightarrow{*}_R q$. ⊠

Let us assign arity 0 to all the tape symbols (Σ_{tape}) and all the states (Q_M) of M. Let Σ be the following signature:

$$\Sigma = \Sigma_{\text{tape}} \cup Q_M \cup \{e_0, e_1, \Lambda, .\},$$

where e_0, e_1 and Λ are new constants.

5.3.1 Representing ID Sequences Recall that an *ID* of M is any string in $\Sigma^*_{\text{tape}} Q_M \Sigma^*_{\text{tape}}$ that does not end with a blank (b). We represent IDs by e-words, where e is one of e_0 or e_1. In particular, the final ID is represented by the word $q_{\text{acc}} \,.\, e_1$ and IDs in general are represented by corresponding e_0-words.

- Any train of the form

$$(v_0 \,.\, e_0) \,.\, (v_1 \,.\, e_0) \,.\, (v_2 \,.\, e_0) \,.\, \cdots \,.\, (v_n \,.\, e_0) \,.\, (q_{\text{acc}} \,.\, e_1) \,.\, \Lambda,$$

  where $n \geq 0$ and each v_i is an ID of M, is called an *ID-train*.

It is clear that the set of all IDs and the set consisting of just the final ID are regular sets. The set of patterns of the ID-trains is given by the regular expression $e_0 e_0^* e_1$. By using the Train Theorem, let

$$A_{\text{id}} = (Q_{\text{id}}, \Sigma, R_{\text{id}}, F_{\text{id}})$$

be a DTA that recognizes the set of all ID-trains.

5.3.2 Representing Move Sequences Let c_{ab} be a new constant for each pair of constants a and b in the set $\Sigma_{\text{tape}} \cup Q_M$. Let also e_2 and Λ' be new constants. Let now Γ be the following signature:

$$\Gamma = \{\, c_{ab} \mid a, b \in \Sigma_{\text{tape}} \cup Q_M \,\} \cup \{e_2, \Lambda', .\}$$

Note that $.$ is the only symbol that occurs in both Σ and Γ.

For and ID w of M we let w^+ denote the successor of w with respect to the transition function of M. For technical reasons it is convenient to let $q^+_{\text{acc}} = \epsilon$, i.e., the successor of the final ID is the empty string. The pair (w, w^+) is called a *move*. Let $w = a_1 a_2 \cdots a_m$ and $w^+ = b_1 b_2 \cdots b_n$ for some $m \geq 1$ and $n \geq 0$. Note that $n \in \{m-1, m, m+1\}$. Let $k = \max(m, n)$. If $m < n$ let $a_k = \text{b}$ and if $n < m$ let $b_k = \text{b}$, i.e., pad the shorter of the two strings with a blank at the end.

- We write $\langle w, w^+ \rangle$ for the string $c_{a_1 b_1} c_{a_2 b_2} \cdots c_{a_k b_k}$ and say that the e_2-word $\langle w, w^+ \rangle \,.\, e_2$ *represents* the move (w, w^+). By a *move-train* we mean any Λ'-train

$$t = t_0 \,.\, t_1 \,.\, \cdots \,.\, t_n \,.\, \Lambda',$$

  such that each t_i represents a move and $n \geq 1$.

Example 17 Take $\Sigma_{\text{in}} = \{\mathtt{0}, \mathtt{1}\}$, and let $\mathtt{q}, \mathtt{p} \in Q_M$. Assume that the transition function δ of M is such that, when the tape head points to a blank and the state is $\mathtt{q}$ then $\mathtt{1}$ is written to the tape, the tape head moves left-and M enters state $\mathtt{p}$, i.e., $\delta(\mathtt{q}, \flat) = (\mathtt{p}, \mathtt{1}, \mathtt{L})$. Imagine that the current ID is $\mathtt{00q}$, i.e., the tape contains the string $\mathtt{00}$ and the tape head points to the bank following the last $\mathtt{0}$. So $(\mathtt{00q}, \mathtt{0p01})$ is a move. This move is represented by the word $c_{\mathtt{00}} \cdot c_{\mathtt{0p}} \cdot c_{\mathtt{q0}} \cdot c_{\flat\mathtt{1}} \cdot e_2 = \langle \mathtt{00q}, \mathtt{0p01} \rangle \cdot e_2$. □

It is straightforward to see that the set of all strings $\langle w, w^+ \rangle$ where w is an ID, is a regular set. The patterns of all move-trains are given by the regular expression $e_2 e_2 e_2^*$. By using the Train Theorem let

$$A_{\text{mv}} = (Q_{\text{mv}}, \Gamma, R_{\text{mv}}, F_{\text{mv}})$$

be a DTA that recognizes the set of all move-trains. Assume also that the states of A_{mv} are new constants.

5.3.3 Construction of S_0 We are now ready to construct S_0. First, let $A_0 = (Q_0, \Sigma_0, R_0, F_0)$ be the following DTA.

$$\begin{aligned}
Q_0 &= Q_{\text{id}} \cup Q_{\text{mv}} \cup \{c_0\}, \\
\Sigma_0 &= \Sigma \cup \Gamma, \\
R_0 &= R_{\text{id}} \cup R_{\text{mv}} \cup \{\, q_1 \cdot q_2 \to c_0 \mid q_1 \in F_{\text{id}},\ q_2 \in F_{\text{mv}} \,\}, \\
F_0 &= \{c_0\}.
\end{aligned}$$

By the disjointness conditions between A_{id} and A_{mv} it follows that A_0 is indeed a deterministic tree automaton. It follows by elementary properties of tree automata that

$$T(A_0) = \{\, t \cdot s \mid t \in T(A_{\text{id}}),\ s \in T(A_{\text{mv}}) \,\}.$$

Let now $E_0 = R_0$ in the rigid equation S_0.

5.4 Final Construction

In this section we finish the construction of S_v^M and prove the undecidability results. The only essential components that we have not defined yet are Π_1 and Π_2. We let Π_1 and Π_2 be the following rewrite systems. The differences between Π_1 and Π_2 are indicated with frames.

$$\begin{aligned}
\Pi_1 &= \{\, c_{ab} \to \boxed{a} \mid a, b \in \Sigma_{\text{tape}} \cup Q_M \,\} \cup \\
&\quad \{\, e_1 \to e_0,\ e_2 \to e_0,\ \Lambda' \to \Lambda,\ \flat \cdot e_0 \to e_0 \,\} \\
\Pi_2 &= \{\, c_{ab} \to \boxed{b} \mid a, b \in \Sigma_{\text{tape}} \cup Q_M \,\} \cup \\
&\quad \{\, e_1 \to e_0,\ e_2 \to e_0,\ \Lambda' \to \Lambda,\ \flat \cdot e_0 \to e_0,\ \boxed{e_0 \cdot \Lambda \to \Lambda} \,\}
\end{aligned}$$

It is easy to see that both sets are in fact reduced sets of ground rewrite rules and thus canonical. For any input string v for M let the term t_v in the system S_v^M be the word $q_0 v \,.\, e_0$, i.e., t_v represents the initial ID of M with input v. We can now state the main theorem of this section.

Theorem 18 *$S_v^M(x,y)$ is solvable iff M accepts v.*

Before proving the theorem we state and prove some useful lemmas.

Lemma 19 *If θ solves $S_1(x,y)$ and $S_2(x,y)$ then $x\theta, y\theta \in \mathcal{T}_{\Sigma\cup\Gamma}$.*

Proof. We prove by induction on the size of $x\theta$ that if θ solves the following system, where t_0 is any term in $\mathcal{T}_{\Sigma\cup\Gamma}$, then $x\theta, y\theta \in \mathcal{T}_{\Sigma\cup\Gamma}$.

$$\{\, \Pi_1 \vdash_\forall x \approx y, \quad \Pi_2 \vdash_\forall x \approx t_0 \,.\, y \,\}$$

The statement follows then by choosing $t_0 = q_0 v \,.\, e_0$.

So consider a fixed t_0 and assume that θ solves the above system. If $x\theta$ is a constant then so is its normal form in Π_2, say $x\theta{\downarrow_{\Pi_2}} = c$, and so $t_0 \,.\, y\theta \xrightarrow{*}_{\Pi_2} c$. But then $c \in \Sigma$ and consequently $x\theta, y\theta \in \mathcal{T}_{\Sigma\cup\Gamma}$. The cases when $x\theta$ is not a constant, but either $x\theta{\downarrow_{\Pi_1}}$ or $x\theta{\downarrow_{\Pi_2}}$ is a constant, are also immediate.

So assume that $x\theta = t_1 \,.\, t$ and $(t_1 \,.\, t){\downarrow_{\Pi_i}} = t_1{\downarrow_{\Pi_i}} \,.\, t{\downarrow_{\Pi_i}}$ for $i \in \{1,2\}$. So $t_1{\downarrow_{\Pi_2}} = t_0{\downarrow_{\Pi_2}}$ and thus $t_1 \in \mathcal{T}_{\Sigma\cup\Gamma}$ since $t_0 \in \mathcal{T}_{\Sigma\cup\Gamma}$; also

$$\Pi_2 \models t \approx y\theta.$$

It follows from $\Pi_1 \models t_1 \,.\, t \approx y\theta$ that $y\theta = s_1 \,.\, s$ for some terms s_1 and s such that

$$\Pi_1 \models t \approx s$$

and $\Pi_1 \models s_1 \approx t_1$. From the latter follows that $s_1 \in \mathcal{T}_{\Sigma\cup\Gamma}$ because $t_1 \in \mathcal{T}_{\Sigma\cup\Gamma}$. Let now θ' be such that $x\theta' = t$ and $y\theta' = s$. So θ' solves the system

$$\{\, \Pi_1 \vdash_\forall x \approx y, \quad \Pi_2 \vdash_\forall x \approx s_1 \,.\, y \,\},$$

and it follows by the induction hypothesis that t and s are in $\mathcal{T}_{\Sigma\cup\Gamma}$, and consequently, so are $t_1 \,.\, t = x\theta$ and $s_1 \,.\, s = y\theta$. ⊠

Lemma 20 *If θ solves $S_v^M(x,y)$ then $x\theta$ is an ID-train and $y\theta$ is a move-train.*

Proof. Assume that θ solves $S_v^M(x,y)$. By Lemma 19, the range of θ is $\mathcal{T}_{\Sigma\cup\Gamma}$. But then, by definition of $S_0(x,y)$ and Lemma 16, $x\theta \,.\, y\theta \in T(A_0)$, and thus $x\theta \in T(A_{\mathrm{id}})$ and $y\theta \in T(A_{\mathrm{mv}})$. ⊠

We can now prove Theorem 18.

Proof. We prove that $S_v^M(x,y)$ is solvable $\Leftrightarrow$ M accepts v.

Proof of '$\Rightarrow$' Let θ be a substitution that solves $S_v^M(x,y)$. By using Lemma 20 we get that $x\theta$ and $y\theta$ have the following form:

$$\begin{aligned} x\theta &= (v_0 \,.\, e_0) \,.\, (v_1 \,.\, e_0) \,.\, \cdots \,.\, (v_{m-1} \,.\, e_0) \,.\, (v_m \,.\, e_1) \,.\, \Lambda \\ y\theta &= (\langle w_0, w_0^+ \rangle \,.\, e_2) \,.\, (\langle w_1, w_1^+ \rangle \,.\, e_2) \,.\, \cdots \,.\, (\langle w_n, w_n^+ \rangle \,.\, e_2) \,.\, \Lambda' \end{aligned}$$

where $m \geq 1$, $n \geq 1$ and all the v_i's and w_i's are IDs of M and $v_m = q_{\mathrm{acc}}$. Since θ solves $S_1(x,y)$, it follows that the normal forms of $x\theta$ and $y\theta$ under Π_1 must coincide. But

$$\begin{aligned} x\theta{\downarrow_{\Pi_1}} &= (v_0 \,.\, e_0) \,.\, (v_1 \,.\, e_0) \,.\, \cdots \,.\, (v_{m-1} \,.\, e_0) \,.\, (v_m \,.\, e_0) \,.\, \Lambda, \\ y\theta{\downarrow_{\Pi_1}} &= (w_0 \,.\, e_0) \,.\, (w_1 \,.\, e_0) \,.\, \cdots \,.\, (w_{n-1} \,.\, e_0) \,.\, (w_n \,.\, e_0) \,.\, \Lambda. \end{aligned}$$

Note that each term $\langle w_i, w_i^+ \rangle \,.\, e_2$ reduces first to $w_i' \,.\, e_0$ where $w_i' = w_i$ or $w_i' = w_i \text{ƀ}$. The extra blank at the end is removed with the rule $\text{ƀ} \,.\, e_0 \to e_0$. So

$$n = m, \quad v_n = q_{\mathrm{acc}}, \quad v_i = w_i \ (0 \leq i \leq n). \tag{6}$$

Since θ solves $S_2(x,y)$ it follows that the normal forms of $x\theta$ and $(q_0 v \,.\, e_0) \,.\, y\theta$ under Π_2 must coincide. But

$$x\theta{\downarrow_{\Pi_2}} = x\theta{\downarrow_{\Pi_1}}$$

because $x\theta$ does not contain any constants from Γ and the rule $e_0 \,.\, \Lambda \to \Lambda$ is not applicable. Moreover, since $w_n = q_{\mathrm{acc}}$, it follows that $w_n^+ = \epsilon$ and thus $\langle w_n, w_n^+ \rangle \,.\, e_0 = c_{q_{\mathrm{acc}}\text{ƀ}} \,.\, e_0$. But

$$(c_{q_{\mathrm{acc}}\text{ƀ}} \,.\, e_0) \,.\, \Lambda \longrightarrow_{\Pi_2} (\text{ƀ} \,.\, e_0) \,.\, \Lambda \longrightarrow_{\Pi_2} e_0 \,.\, \Lambda \longrightarrow\!\!\!\!\!\rightarrow_{\Pi_2} \Lambda.$$

The normal form of $(q_0 v \,.\, e_0) \,.\, y\theta$ under Π_2 is thus

$$(q_0 v \,.\, e_0) \,.\, (w_0^+ \,.\, e_0) \,.\, (w_1^+ \,.\, e_0) \,.\, \cdots \,.\, (w_{n-1}^+ \,.\, e_0) \,.\, \Lambda.$$

It follows that $v_0 = q_0 v$, i.e., v_0 is the initial ID of M with input v, and

$$w_i^+ = v_{i+1} \ (0 \leq i < n). \tag{7}$$

From (6) and (7) follows now that $(v_0, v_1, \ldots, v_n)$ is a valid computation of M, and thus M accepts v.

Proof of '$\Leftarrow$' Assume that M accepts v. So there exists a valid computation $(v_0, v_1, \ldots, v_n)$ of M where $v_0 = q_0 v$, $v_n = q_{\mathrm{acc}}$ and $v_i^+ = v_{i+1}$ for $0 \leq i < n$. Let θ be such that $x\theta$ is the corresponding ID-train and $y\theta$ the corresponding move-train. It follows easily that θ solves $S_M(x,y)$. $\boxtimes$

The *shifted pairing* technique that is used in Theorem 18 is illustrated in Figure 1. The Degtyarev–Voronkov theorem is an immediate consequence of Theorem 18, because all the constructions in it are effective.

Furthermore, the following result due to Plaisted [36] (that we used to prove Corollary 10). is an immediate consequence.

Corollary 21 (Plaisted) *SREU is undecidable even if the left-hand sides are ground.*

Furthermore, we can sharpen this result as follows.

Corollary 22 *SREU is undecidable if the left-hand sides are ground, there are only two variables and three rigid equations and one binary function symbol.*

The undecidability with two variables and three rigid equations may seem like an artifical extra condition, but in fact, it turns out to be an important special case. One implication is that the provability problem for the $\exists\exists$-fragment of intuitionistic logic with equality is undecidable. Another important fact is that two variables are *necessary* to get undecidability. If there is only *one* variable then SREU is decidable [9].

Remark We can also note that one constant suffices. One can easily simulate any number of constants with one constant and a binary function symbol.

5.5 Undecidability Proofs of SREU

The first proof of the udecidability of SREU [14] was by reduction of the monadic semi-unification [1] to SREU. This proof was followed by two alternative (more transparent) proofs by the same authors, first by reducing second order unification to SREU [13, 17], and then by reducing Hilbert's tenth problem to SREU [16]. The undecidability of second order unification was proved by Goldfarb [24]. Reduction of second order unification to SREU is very simple, showing how close these problem are to each other. Plaisted took the Post's Correspondence Problem and reduced it to SREU [36]. From his proof follows that SREU is undecidable already with ground left-hand sides. Veanes improves the construction of Plaisted by using the membership problem for Turing machines and shows that two variables and one binary function symbol is enough to obtain undecidability [40, 41].

6 Minimal Undecidable Case of the n-Skeleton Problem

Let $\varphi_v^M(x, y)$ stand for the following formula:

$$\varphi_v^M(x, y) \quad = \quad (E_0 \Rightarrow x \centerdot y \approx c_0) \wedge$$

$$(\Pi_1 \Rightarrow x \approx y) \wedge$$
$$(\Pi_2 \Rightarrow x \approx t_v \,.\, y).$$

Let $\varphi^M(z, x, y)$ stand for the formula $\varphi_v^M(x, y)$ with the term t_v replaced by the variable z. It is important to note that the construction of E_0, Π_1 and Π_2 is *independent* of v, which justifies this notation. Let M_u be the Turing machine that accepts the universal language L_u,

$$L_u = \{\, \langle M, v\rangle \mid M \text{ is a Turing machine that accepts } v \,\},$$

where $\langle M, v\rangle$ is some encoding of the pair (M, v) that is carried out in some fixed alphabet. We write φ^u for φ^{M_u}. The precise details of the encoding are not relevant here. We get the following result.

Theorem 23 *For all $n \geq 1$, n-Skeleton problem of Horn formulas restricted to $2n$ variables and $3n$ clauses with ground negative literals, is undecidable already for some fixed negative literals.*

Proof. For any Turing machine M and input string v we have that the formula $\varphi^u(t_{\langle M,v\rangle}, x, y)$ has a corroborator iff M accepts v. The statement follows now by Theorem 8. ⊠

7 Relations to Intuitionistic Logic

The decision problems in intuitionistic logic have not been as thoroughly studied as the corresponding problems in classical logic [3]. In particular, new results about the *prenex fragment* of intuitionistic logic (i.e., closed prenex formulas that are intuitionistically provable), have been obtained quite recently by Degtyarev and Voronkov [16, 17, 15] and Voronkov [44]. Some of these results are:

1. Decidability, and in particular PSPACE-completeness, of the prenex fragment of intuitionistic logic *without* equality [15].

2. Prenex fragment of intuitionistic logic *with* equality but *without* function symbols is PSPACE-complete [15]. Decidability of this fragment was proved in Orevkov [35].

3. Prenex fragment of intuitionistic logic with equality in the language with one unary function symbol is decidable [15].

4. $\exists^*$-fragment of intuitionistic logic with equality is undecidable [16, 17].

In some of the above results, the corresponding result has first been obtained for a fragment of SREU with similar restrictions. There are close connections between intuitionistic logic with equality and SREU [44]. By using the undecidability of SREU, the proof of (4) is straightforward. The undecidability of the $\exists^*$-fragment is improved in Veanes [40] by showing that the

5. $\exists\exists$-fragment of intuitionistic logic with equality is undecidable.

We obtain the following uniform characterization of all the recursively enumerable sets in the $\exists\exists$-fragment of intuitionistic logic with equality. Let us consider Turing machines with some fixed tape alphabet and a fixed symbol q_0 for the initial state. Let t_v denote the word that represents $q_0 v$ (the initial ID for input string v).

Theorem 24 *For any Turing machine M and input string v for M,*

$$\vdash_i \exists x \exists y \varphi^M(t_v, x, y) \quad \Leftrightarrow \quad M \text{ accepts } v.$$

Proof. The formula $\exists x \exists y \varphi^M(t_v, x, y)$ is provable intuitionistically iff there exists a corroborator for $\varphi^M(t_v, x, y)$ (cf [17, Proof of Theorem 3]). Use now Theorem 18. ⊠

The following statement is an easy corollary of Theorem 24.

Corollary 25 *The $\exists\exists$-fragment of intuitionistic logic is undecidable already under the following restrictions:*

1. *The signature has two symbols: one constant and one binary function symbol.*
2. *The only connectives are $\wedge$ and at most three $\Rightarrow$'s.*
3. *The antecedents of all implications are closed.*
4. *The antecedents of implications may be fixed.*

If there is only one variable then SREU is decidable [9]. It follows also that the

6. $\forall^*\exists\forall^*$-fragment of intuitionistic logic with equality is decidable [9].

7.1 Proof Search in $\mathrm{LJ}^{\approx}$

Proof search in intuitionistic logic with equality is closely connected with SREU, and, unlike in the classical case, the handling of SREU is in fact *unavoidable* in that context [43, 44]. Voronkov considers a particular sequent calculus based proof system $\mathrm{LJ}^{\approx}$ [43]. In that context a *skeleton* is the structure of a derivation in $\mathrm{LJ}^{\approx}$, and *skeleton instantiation* is the problem of the existence of a derivation with a given skeleton. SREU is polynomially equivalent to skeleton instantiation in $\mathrm{LJ}^{\approx}$ [43]. We get the following result. (See Voronkov [43] for precise definitions.)

Corollary 26 *There is a fixed skeleton with two applications of $(\rightarrow \exists)$ and three applications of $(\rightarrow\Rightarrow)$ for which the skeleton instantiation problem in $\mathrm{LJ}^{\approx}$ is undecidable.*

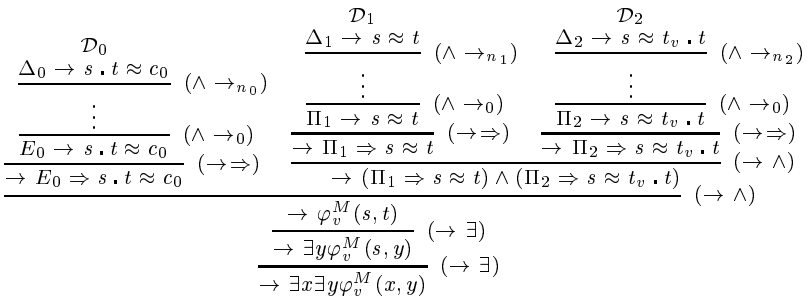


Figure 2: A derivation of $\exists x \exists y \varphi_v^M(x,y)$ in $\mathrm{LJ}^{\approx}$; Δ_0, Δ_1 and Δ_2 are multisets of equations corresponding to E_0, Π_1 and Π_2, respectively; n_0, n_1 and n_2 are the number of $\wedge$'s minus one, in E_0, Π_1 and Π_2, respectively. It is actually the existence of the derivations $\mathcal{D}_0$, $\mathcal{D}_1$ and $\mathcal{D}_2$, that corresponds to the solvability problem of the system S_v^M of rigid equations.

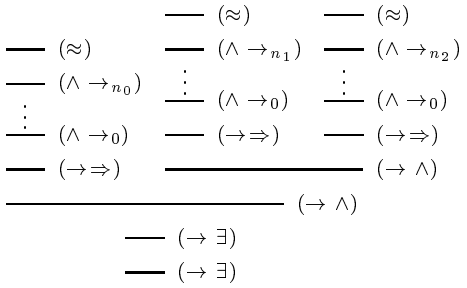


Figure 3: The skeleton of the derivation in Figure 2.

Proof. By using the results proved in Voronkov [43, 44], the sentence $\exists x \exists y \varphi_v^M(x,y)$ is intuitionistically provable iff the sequent $\to \exists x \exists y \varphi_v^M(x,y)$ can be derived in $\mathrm{LJ}^{\approx}$ (see Figure 2) with the skeleton shown in Figure 3. Let $M = M_{\mathrm{u}}$. The statement follows now from Theorem 24. ⊠

7.2 Other Fragments

Decidability problems for other fragments of intuitionistic logic have been studied by Orevkov [34, 35], Mints [33], Statman [38] and Lifschitz [31]. Orevkov proves that the $\neg\neg\forall\exists$-fragment of intuitionistic logic with function symbols is undecidable [34]. Lifschitz proves that intuitionistic logic with equality and without function symbols is undecidable, i.e., that the pure constructive theory of equality is undecidable [31]. Orevkov shows decidability of some fragments (that are close to the prenex fragment) of intuitionistic logic with equality [35]. Statman proves that the intuitionistic propositional logic is PSPACE-complete [38].

8 Current Status of SREU and Open Problems

Here we briefly summarize the current status of SREU and mention some open problems. Many related results are already mentioned above. The first decidability proof of rigid E-unification is given in Gallier, Narendran, Plaisted and Snyder [21]. Recently a simpler proof, without computational complexity considerations, has been given by de Kogel [7]. We start with the **solved cases**:

- Rigid E-unification with ground left-hand side is NP-complete [30]. Rigid E-unification in general is NP-complete and there exist finite complete sets of unifiers [20, 21].
- Rigid E-unification with one variable is P-complete [9]. Or, more generally, SREU with one variable and a bounded number of rigid equations is P-complete [9].
- If all function symbols have arity ≤ 1 (the *monadic* case) then it follows that SREU is PSPACE-hard [25]. If only one unary function symbol is allowed then the problem is decidable [11, 12]. If only constants are allowed then the problem is NP-complete [12] if there are at least two constants.
- About the monadic case it is known that if there are more than 1 unary function symbols then SREU is decidable iff it is decidable with just 2 unary function symbols [12].
- If the left-hand sides are ground then the monadic case is decidable [26]. Monadic SREU with one variable is PSPACE-complete [26].
- The word equation solving [32] (i.e., unification under associativity), which is an extremely hard problem with no interesting known computational complexity bounds, can be reduced to monadic SREU [11].
- Monadic SREU is equivalent to a non-trivial extension of word equations [26].
- Monadic SREU is equivalent to the decidability problem of the prenex fragment of intuitionistic logic with equality with function symbols of arity ≤ 1 [15].
- In general SREU is undecidable [14]. Moreover, SREU is undecidable under the following restrictions:
  - The left-hand sides of the rigid equations are ground [36].
  - Furthermore, there are only two variables [40, 41] and three rigid equations with fixed ground left-hand sides.

- SREU with one variable is decidable, in fact EXPTIME-complete [9]. Moreover, SREU restricted to rigid equations that either contain one variable, or have a ground left-hand side and a right-hand side that is an equality between two variables, is decidable [8].

Note also that SREU is decidable when there are no variables. Actually, the problem is then P-complete because the uniform word problem for ground equations is P-complete [29]. The **unsolved cases** are:

- Decidability of monadic SREU [26].
- Decidability of SREU with *two* rigid equations.

Both problems are highly non-trivial. Another intriguing open problem is to study the Herbrand f-Skeleton problem. In particular, the following open problem is posed in Voronkov [45]:

- Does there exist a computable strategy f for which the f-Skeleton problem is decidable?

Acknowledgements

We wish to thank Andrei Voronkov and Anatoli Degtyarev for many valuable comments and discussions.

References

[1] M. Baaz. Note on the existence of most general semi-unifiers. In *Arithmetic, Proof Theory and Computation Complexity*, volume 23 of *Oxford Logic Guides*, pages 20–29. Oxford University Press, 1993.

[2] G. Birkhoff. On the structure of abstract algebras. *Proc. Cambridge Phil. Soc.*, 31:433–454, 1935.

[3] E. Börger, E. Grädel, and Yu. Gurevich. *The Classical Decision Problem*. Springer Verlag, 1997.

[4] C.C. Chang and H.J. Keisler. *Model Theory*. North-Holland, Amsterdam, third edition, 1990.

[5] J.L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi. Bottom-up tree pushdown automata: classification and connection with rewrite systems. *Theoretical Computer Science*, 127:69–98, 1994.

[6] M. Dauchet. Rewriting and tree automata. In H. Comon and J.P. Jouannaud, editors, *Term Rewriting (French Spring School of Theoretical Computer Science)*, volume 909 of *Lecture Notes in Computer Science*, pages 95–113. Springer Verlag, Font Romeux, France, 1993.

[7] E. De Kogel. Rigid E-unification simplified. In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *Theorem Proving with Analytic Tableaux and Related Methods*, number 918 in Lecture Notes in Artificial Intelligence, pages 17–30, Schloß Rheinfels, St. Goar, Germany, May 1995.

[8] A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. Decidability and complexity of simultaneous rigid E-unification with one variable and related results. Submitted to *Theoretical Computer Science*, 1997.

[9] A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid E-unification with one variable. UPMAIL Technical Report 139, Uppsala University, Computing Science Department, March 1997.

[10] A. Degtyarev, Yu. Gurevich, and A. Voronkov. Herbrand's theorem and equational reasoning: Problems and solutions. UPMAIL Technical Report 128, Uppsala University, Computing Science Department, September 1996. Appears in the Bulletin of the European Association for Theoretical Computer Science (Vol 60, October 1996).

[11] A. Degtyarev, Yu. Matiyasevich, and A. Voronkov. Simultaneous rigid E-unification is not so simple. UPMAIL Technical Report 104, Uppsala University, Computing Science Department, April 1995.

[12] A. Degtyarev, Yu. Matiyasevich, and A. Voronkov. Simultaneous rigid E-unification and related algorithmic problems. In *Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 494–502, New Brunswick, NJ, July 1996. IEEE Computer Society Press.

[13] A. Degtyarev and A. Voronkov. Reduction of second-order unification to simultaneous rigid E-unification. UPMAIL Technical Report 109, Uppsala University, Computing Science Department, June 1995.

[14] A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. UPMAIL Technical Report 105, Uppsala University, Computing Science Department, May 1995.

[15] A. Degtyarev and A. Voronkov. Decidability problems for the prenex fragment of intuitionistic logic. In *Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 503–512, New Brunswick, NJ, July 1996. IEEE Computer Society Press.

[16] A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. In H. Kleine Büning, editor, *Computer Science Logic.*

9th International Workshop, CSL'95, volume 1092 of *Lecture Notes in Computer Science*, pages 178–190, Paderborn, Germany, September 1995, 1996.

[17] A. Degtyarev and A. Voronkov. The undecidability of simultaneous rigid E-unification. *Theoretical Computer Science*, 166(1–2):291–300, 1996.

[18] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 6, pages 243–309. North Holland, Amsterdam, 1990.

[19] J. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.

[20] J. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Rigid E-unification: NP-completeness and applications to equational matings. *Information and Computation*, 87(1/2):129–195, 1990.

[21] J.H. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Rigid E-unification is NP-complete. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, July 1988.

[22] J.H. Gallier, S. Raatz, and W. Snyder. Theorem proving using rigid E-unification: Equational matings. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, 1987.

[23] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiodó, Budapest, 1984.

[24] W.D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[25] J. Goubault. Rigid $\vec{E}$-unifiability is DEXPTIME-complete. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*. IEEE Computer Society Press, 1994.

[26] Y. Gurevich and A. Voronkov. The monadic case of simultaneous rigid E-unification. UPMAIL Technical Report 137, Uppsala University, Computing Science Department, 1997. To appear in Proc. of *ICALP'97*.

[27] J. Herbrand. *Logical Writings*. Harvard University Press, 1972.

[28] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Co., 1979.

[29] D. Kozen. Complexity of finitely presented algebras. In *Proc. of the 9th Annual Symposium on Theory of Computing*, pages 164–177, New York, 1977. ACM.

[30] D. Kozen. Positive first-order logic is NP-complete. *IBM J. of Research and Development*, 25(4):327–332, 1981.

[31] V. Lifschitz. Problem of decidability for some constructive theories of equalities (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 4:78–85, 1967. English Translation in: Seminars in Mathematics: Steklov Math. Inst. 4, Consultants Bureau, NY-London, 1969, p.29–31.

[32] G.S. Makanin. The problem of solvability of equations in free semigroups. *Mat. Sbornik (in Russian)*, 103(2):147–236, 1977. English Translation in American Mathematical Soc. Translations (2), vol. 117, 1981.

[33] G.E. Mints. Choice of terms in quantifier rules of constructive predicate calculus (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 4:78–85, 1967. English Translation in: Seminars in Mathematics: Steklov Math. Inst. 4, Consultants Bureau, NY-London, 1969, p.43–46.

[34] V.P. Orevkov. Unsolvability in the constructive predicate calculus of the class of the formulas of the type $\neg\neg\forall\exists$ (in Russian). *Soviet Mathematical Doklady*, 163(3):581–583, 1965.

[35] V.P. Orevkov. Solvable classes of pseudo-prenex formulas (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 60:109–170, 1976. English translation in: Journal of Soviet Mathematics.

[36] D.A. Plaisted. Special cases and substitutes for rigid E-unification. Technical Report MPI-I-95-2-010, Max-Planck-Institut für Informatik, November 1995.

[37] W. Snyder. Efficient ground completion: An $O(n\log n)$ algorithm for generating reduced sets of ground rewrite rules equivalent to a set of ground equations E. In G. Goos and J. Hartmanis, editors, *Rewriting Techniques and Applications*, volume 355 of *Lecture Notes in Computer Science*, pages 419–433. Springer-Verlag, 1989.

[38] R. Statman. Lower bounds on Herbrand's theorem. *Proc. American Mathematical Society*, 75(1):104–107, 1979.

[39] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.

[40] M. Veanes. Uniform representation of recursively enumerable sets with simultaneous rigid E-unification. UPMAIL Technical Report 126, Uppsala University, Computing Science Department, July 1996.

[41] M. Veanes. The undecidability of simultaneous rigid E-unification with two variables. To appear in *Proc. Kurt Gödel Colloquium KGC'97*, 1997.

[42] P.J. Voda and J. Komara. On Herbrand skeletons. Technical report, Institute of Informatics, Comenius University Bratislava, July 1995. Revised January 1996.

[43] A. Voronkov. On proof-search in intuitionistic logic with equality, or back to simultaneous rigid E-Unification. UPMAIL Technical Report 121, Uppsala University, Computing Science Department, January 1996.

[44] A. Voronkov. Proof search in intuitionistic logic with equality, or back to simultaneous rigid E-unification. In M.A. McRobbie and J.K. Slaney, editors, *Automated Deduction — CADE-13*, volume 1104 of *Lecture Notes in Computer Science*, pages 32–46, New Brunswick, NJ, USA, 1996.

[45] A. Voronkov. Rigid variables considered harmful. UPMAIL Technical Report 134, Uppsala University, Computing Science Department, January 1997. To appear in Proc. *IJCAI'97*.