# Logic with Equality: Partisan Corroboration, and Shifted Pairing*

Yuri Gurevich and Margus Veanes

February 18, 1999

**Yuri Gurevich:** Microsoft Research, One Microsoft Way, Redmond,
WA 98052-6399, USA
gurevich@microsoft.com

**Margus Veanes:** Max-Planck-Institut für Informatik, Im Stadtwald,
66123 Saarbrücken, Germany.
veanes@mpi-sb.mpg.de

*Preliminary versions of this paper have appeared as: UPMAIL Technical Report 138, Uppsala University, 1997, and Research Report MPI-I-98-2-014, Max-Planck-Institut für Informatik, 1998. Yuri Gurevich has been partially supported by the NSF grant CCR 95-04375.

**Proposed running head:** Partisan Corroboration, Shifted Pairing

**Corresponding author:** Margus Veanes,

Max-Planck-Institut für Informatik,

Im Stadtwald,

66123 Saarbrücken,

Germany.

email: `veanes@mpi-sb.mpg.de`

phone: +49-681-9325 218

fax: +49-681-9325 299

# Abstract

Herbrand's theorem plays a fundamental role in automated theorem proving methods based on tableaux. The crucial step in procedures based on such methods can be described as the *corroboration* problem or the *Herbrand skeleton* problem, where, given a positive integer $m$, called *multiplicity*, and a quantifier free formula, one seeks a valid disjunction of $m$ instantiations of that formula. In the presence of *equality*, which is the case in this paper, this problem was recently shown to be undecidable.

The main contributions of this paper are two theorems. The first, the *Partisan Corroboration Theorem*, relates corroboration problems with different multiplicities. The second, the *Shifted Pairing Theorem*, is a finite tree automata formalization of a technique for proving undecidability results through direct encodings of valid Turing machine computations.

These theorems are used in the paper to explain and sharpen several recent undecidability results related to the *corroboration* problem, the *simultaneous rigid E-unification* problem and the *prenex fragment of intuitionistic logic with equality*.

## List of symbols

| **character** | **explanation** |
|---|---|
| ␣ | a “blank” symbol: `\texttt{\char32}` |
| ↾ | arrow symbol from AMS-TeX: `\restriction` |

All other symbols used in this document are standard symbols from LaTeX.

# 1 Introduction

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

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.

Given a positive integer $m$, a set of $m$ ground substitutions $\{\theta_1, \ldots, \theta_m\}$ is an *$m$-corroborator* for $\varphi$ if the disjunction $\varphi\theta_1 \vee \cdots \vee \varphi\theta_m$ is provable. A ground substitution $\theta$ *corroborates* $\varphi$ if $\{\theta\}$ 1-corroborates $\varphi$; such a $\theta$ is called a *corroborator* for $\varphi$.

One popular form of the classical Herbrand theorem [e.g. Herbrand 1972] is this:

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

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

$$(c \approx c_0 \Rightarrow x \approx c_1) \wedge (c \approx c_1 \Rightarrow x \approx c_0)$$

The Herbrand theorem plays a fundamental role in automated theorem proving methods known as the *rigid variable methods* [Voronkov 1997]. We can identify the following procedure underlying such methods. Let $\exists \vec{x} \varphi(\vec{x})$ be a closed formula that we wish to prove.

*The principal procedure of rigid variable methods*

**Step I:** Choose a positive integer $m$.

**Step II:** Check if there exists an $m$-corroborator for $\varphi$.

**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 Corroboration Problem*

**Instance:** A quantifier free formula $\varphi$ and a positive integer $m$.

**Question:** Is the minimum multiplicity for $\varphi$ bounded by $m$?

Corroboration for a fixed $m$ is called *$m$-corroboration.* A detailed discussion of corroboration and related problems is given by Degtyarev, Gurevich & Voronkov [1996]. It is important to us here that corroboration is intimately related to existential intuitionistic provability and simultaneous rigid $E$-unification [Gallier, Raatz & Snyder 1987]. The first of these problems is easy to formulate:

*The Existential Intuitionistic Provability Problem*

**Instance:** An existential formula $\exists \vec{x} \varphi(\vec{x})$.

**Question:** Is the formula provable in intuitionistic logic with equality?

The second requires auxiliary definitions. A *rigid equation* is an expression $E \vdash^{\mathrm{r}} e$ where $E$ is a finite set of equations and $e$ is an equation. A ground substitution $\theta$ *solves* a rigid equation $E \vdash^{\mathrm{r}} e$ if $e\theta$ is a logical consequence of $E\theta$. 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 [e.g. Degtyarev, Gurevich & Voronkov 1996]. Several false decidability claims have been published until, finally, Degtyarev & Voronkov [1995] proved SREU to be undecidable. Moreover, Plaisted [1995] has shown that the fragment of SREU with ground left-hand sides is already undecidable (the *left-hand side* of a rigid equation $E \vdash^{\mathrm{r}} e$ is $E$).

It is easy to see that SREU is essentially a special case of 1-corroboration for Horn formulas. Hence, the result of Degtyarev & Voronkov shows that corroboration is undecidable already in this very special case. Voronkov [1997] has suggested the following generalization of the corroboration problem. Let $f$ be a function that assigns a positive integer to every pair $(k, \varphi)$ where $k$ is a positive integer and $\varphi$ a formula in our logic. Moreover, it is assumed that $k < l$ implies that $f(k, \varphi) \leq f(l, \varphi)$. 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 Corroboration Problem with Strategy f*

**Instance:** A quantifier free formula $\varphi$ and a positive integer $k$.

**Question:** Is the minimum multiplicity for $\varphi$ bounded by $f(k, \varphi)$?

Corroboration with a strategy that does not depend on it arguments, i.e., takes a constant value $m$ for all arguments, is simply $m$-corroboration. Voda & Komara [1995] have proved that, for each positive integer $m$, the $m$-corroboration problem is undecidable. One important conclusion for automated theorem proving, drawn by Voda & Komara, is that there is no $m$ for which one can effectively determine whether $m$ bounds the minimum multiplicity for a given formula. The proof of Voda & Komara is very technical, and 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 conjunction of Horn clauses. Since the equality sign is the only relation symbol in our logic, every Horn clause $\psi$ 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 $\varphi$ *guarded* if, for every variable $x$ that occurs in $\varphi$, there exists a clause $E \Rightarrow s \approx t$ in $\varphi$ where $E$ and $s$ are ground and $x$ occurs in $t$. Finally, call a corroborator $\theta$ of a disjunction $\varphi$ *partisan* if already $\theta$ corroborates one of the disjuncts of $\varphi$. 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 Voda & Komara's [1995] result from Degtyarev & Voronkov's [1995] theorem in Section 4. Moreover, we strengthen the theorem of Voda & Komara in several ways. For each $m$, we effectively reduce SREU to the m-corroboration 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 at most one) SREU reduces to monadic m-corroboration; this reduction is of interest because the decidability of monadic SREU is an open problem.

In Section 5 we use *finite tree automata* theory to describe a powerful technique, named *shifted pairing* by Plaisted [1995], for proving undecidability results via encodings of valid Turing machine computations. The main components are two finite tree automata $A_{\mathrm{mv}}$, $A_{\mathrm{id}}$ and two ground term rewrite systems $\Pi_1$ and $\Pi_2$ that are obtained (effectively) from a given Turing machine $M$ and are used to check the existence of a *valid computation of* $M$. See Figure 1.

**Shifted Pairing Theorem**

*There are two finite tree automata* $A_{\mathrm{mv}}$ *and* $A_{\mathrm{id}}$ *and two ground rewrite systems* $\Pi_1$ *and* $\Pi_2$ *such that, it is undecidable whether, given a ground term* $t_0$, $A_{\mathrm{mv}}$ *recognizes a term* $s$ *and* $A_{\mathrm{id}}$ *recognizes a term* $t$, *such that* $s$ *reduces in* $\Pi_1$ *to* $t$ *and* $f(t_0, s)$ *reduces in* $\Pi_2$ *to* $t$.

A more precise version of the theorem is stated in Section 5. The shifted pairing technique, and in particular the Shifted Pairing Theorem, that is an improvement upon [Plaisted 1995, Veanes 1997], has recently been applied successfully to settle several open decidability questions [Ganzinger, Jacquemard

& Veanes 1998, Levy & Veanes 1998, Veanes 1997, Veanes 1998].

In Section 6, we use the Shifted Pairing Theorem to show the undecidability of a fragment of SREU with only two variables and three rigid equations with ground left-hand sides, which constitutes the currently known *least undecidable fragment* of SREU. Using this result and the Partisan Corroboration Theorem, we show, for each positive integer $m$, the undecidability of $m$-corroboration when each formula is a conjunction of $3m$ Horn clauses with $2m$ variables and ground negative literals.

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 mention some open problems.

## 2 Preliminaries

We will first establish some notation and terminology. We follow Chang & Keisler [1990] 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 $\Sigma$, possibly with an index, to stand for a signature, i.e., $\Sigma$ is a collection of function symbols with fixed arities. A function symbol of arity 0 is called a *constant.* We will always assume that $\Sigma$ *contains at least one constant.*

## 2.1 Terms and formulas

Terms and formulas are defined in the standard manner and are called $\Sigma$-*terms* and $\Sigma$-*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 $X$, $FV(X)$ for the set of all free variables in $X$ and $Con(X)$ for the set of all constants in $X$. We write $X(x_1, x_2, \ldots, x_n)$ to express that $FV(X) \subseteq \{x_1, x_2, \ldots, x_n\}$. 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) occurrence 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 $\theta$ 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 $FV(X) = \emptyset$. By $\mathcal{T}_\Sigma$ or simply $\mathcal{T}$ we denote the set of all ground $\Sigma$-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 (i.e., a *strict* Horn clause). A Horn clause can be written as $E \Rightarrow s \approx t$ where $E$ is a (possibly empty) 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 $\mathcal{A}$ and $\mathcal{B}$. A first-order structure in a signature $\Sigma$ is called a $\Sigma$-*structure*. For $f \in \Sigma$ we write $f^{\mathcal{A}}$ for the interpretation of $f$ in $\mathcal{A}$.

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

For $X$ a sentence or a set of sentences, $\mathcal{A} \models X$ means that the structure $\mathcal{A}$ 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.

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

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

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

## 2.3 Term rewriting

In most cases we 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 [e.g. Dershowitz & Jouannaud 1990]. We will only use very elementary properties. In particular, in the next section we will use Birkhoff's [1935] completeness theorem for equational logic in the case of ground equations.

**Theorem 1 (Birkhoff)** *Given a ground set of equations $E$ and a ground equation $s \approx t$, $E \models s \approx t$ if and only if $s$ can be reduced to $t$ by using the equations in $E$ as rewrite rules in both directions.*

Let $R$ be a ground rewrite system. We write $R^{\approx}$ for the corresponding set of equations:

$$R^{\approx} = \{\, s \approx t \mid s \to t \in R \,\}.$$

In Section 6 we will use the following property of ground canonical (or convergent) rewrite systems $R$ [e.g. Dershowitz & Jouannaud 1990, Section 2.4]. For any two ground terms $t$ and $s$, the equation $t \approx s$ follows logically from $R^{\approx}$ if and only if the normal forms of $t$ and $s$ in $R$ coincide, i.e.,

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

Snyder [1989] has given a very simple but useful condition for showing that a ground rewrite system $R$ is canonical, namely that it is *reduced*: for each rule $s \to t$ in $R$, $s$ is irreducible in $R \setminus \{s \to t\}$ and $t$ is irreducible in $R$. We will use this test on several occasions, to show that a ground rewrite system is canonical.

## 2.4 Finite tree automata

A *finite tree automaton* or *TA* is a quadruple $(Q, \Sigma, R, F)$, where

- $Q$ is a finite set of constants called *states*,
- $\Sigma$ 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 TA is called *deterministic* or a *DTA* if there are no two different rules in it with the same left-hand side. Terms are also called *trees* and a *forest* is a set of trees. The forest *recognized* by a TA $A = (Q, \Sigma, R, F)$ is the following set that is denoted by $\mathcal{L}(A)$:

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

A forest is *recognizable* or *regular* if it is recognized by some TA. A well-known fact is that every regular forest is recognized by a DTA. Two finite tree automata are called *constant-disjoint* if there is no constant that occurs in both of them.

**Example 2** Let $A = (\{q\}, \Sigma, R, \{q\})$ be a TA, where

$$\begin{aligned} R = \ & \{\, c \to q \mid c \text{ is a constant in } \Sigma \,\} \cup \\ & \{\, f(q,\ldots,q) \to q \mid f \text{ is a function symbol in } \Sigma \,\}. \end{aligned}$$

This DTA recognizes the forest $\mathcal{T}_\Sigma$. □

## 3 Partisan Corroboration Theorem

The following lemma is used in the Partisan Corroboration Theorem. We say that two (sets of) expressions $X$ and $Y$ are *constant-disjoint* if $Con(X) \cap Con(Y) = \emptyset$.

**Lemma 3** *Let $\varphi_i$ 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 $\Sigma_i$-structure $\mathcal{A}_i$ such that $\mathcal{A}_i \models \neg\varphi_i$. Without loss of generality, take the universes of all the models to be pairwise disjoint.

We now construct a $\Sigma$-structure $\mathcal{A}$ such that $\mathcal{A}_i \subseteq \mathcal{A}\restriction\Sigma_i$ for $i \in I$. First, let the universe of $\mathcal{A}$ be the union of the universes of the $\mathcal{A}_i$'s. Next, for each $i \in I$ and constant $c \in \Sigma_i$ let $c^{\mathcal{A}} = c^{\mathcal{A}_i}$. For each $n$-ary function symbol $f$ in $\Sigma$ define $f^{\mathcal{A}}$ as follows. For all individuals $\vec{a} = a_1, \ldots, a_n$ in $\mathcal{A}$,

$$f^{\mathcal{A}}(\vec{a}) = \begin{cases} f^{\mathcal{A}_i}(\vec{a}), & \text{if } a_1, \ldots, a_n \text{ are in } \mathcal{A}_i; \\ a_1, & \text{otherwise.} \end{cases}$$

It is clear that $\mathcal{A}$ is well-defined because of the disjointness criteria and that $\mathcal{A}_i \subseteq \mathcal{A}\restriction\Sigma_i$ for $i \in I$. One easily establishes, by induction on terms and formulas that, if $\mathcal{B} \subseteq \mathcal{A}$ then for all quantifier free sentences $\varphi$, $\mathcal{B} \models \varphi$ if and only if $\mathcal{A} \models \varphi$. Hence $\mathcal{A}\restriction\Sigma_i \models \neg\varphi_i$ for $i \in I$, and thus $\mathcal{A} \models \neg\varphi_i$ for $i \in I$. But this contradicts that $\models \bigvee_{i \in I} \varphi_i$. ⊠

Note that Lemma 3 can be seen as a particular case of Łoś-Tarski theorem (existential sentences are preserved under extensions).

If we drop the constant-disjointness criterion in Lemma 3, then of course the lemma is false. A simple counterexample is

$$\models c_0 \approx c_1 \vee \neg(c_0 \approx c_1).$$

We will state now some other obvious but useful lemmas. Lemma 4 is an easy corollary of Birkhoff's completeness theorem.

**Lemma 4** *Let $t$ and $s$ be ground terms and let $E$ and $E'$ be ground sets of equations such that $Con(E') \cap (Con(E) \cup Con(s)) = \emptyset$. Then $E' \cup E \models t \approx s$ implies $E \models t \approx s$.*

**Proof.** Let $E$, $E'$, $s$ and $t$ be given and assume that $E' \cup E \models t \approx s$. By Birkhoff's [1935] 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 $\Sigma(s_i) \subseteq \Sigma(E) \cup \Sigma(s)$ and only a rule from $E$ can be used to rewrite $s_i$. The statement follows from the completeness theorem of Birkhoff. ⊠

**Lemma 5** *Let $t$ and $s$ be ground terms and let $E$ be a ground set of equations. Then $E \models t \approx s$ implies $\Sigma(t) \subseteq \Sigma(E) \cup \Sigma(s)$.*

**Proof.** Take $E' = \emptyset$ in the proof of Lemma 4. ⊠

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

**Lemma 6** *Let $t$ and $s$ be ground terms and $E'$ and $E$ ground sets of equations such that $E$ is finite and $Con(E') \cap (Con(E) \cup Con(s)) = \emptyset$. Then*

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

**Proof.** Let $E$, $E'$, $s$ and $t$ be given. 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 4, i.e., $\models (E \Rightarrow t \approx s)$. ⊠

We will use the following definitions. Let $\varphi$ be a quantifier free formula and $m$ a positive integer. A set of $m$ ground substitutions $\Theta$ is an *$m$-corroborator* for $\varphi$ if

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

When $\Theta = \{\theta\}$ we say that $\theta$ is a *corroborator* for $\varphi$ or *corroborates* $\varphi$. The $m$-*corroboration* problem is the problem of determining whether a given quantifier free formula has an $m$-corroborator.

For $x \in FV(\varphi)$, a *guard for* $x$ *in* $\varphi$, if it exists, is a clause

$$E \Rightarrow t \approx s$$

in $\varphi$ such that $E$ and $s$ are ground and $x$ occurs in $t$. We say that

$$\bigwedge_{x \in FV(\varphi)} \psi_x$$

is a *guard* of $\varphi$ if each $\psi_x$ is a guard for $x$ in $\varphi$; $\varphi$ is is called *guarded* if it has a guard.

Intuitively, in the light of Lemma 5, the notion of a Horn formula being guarded is a sufficient condition to guarantee that if there is a corroborator $\theta$ for $\varphi$ then $\Sigma(\varphi\theta) = \Sigma(\varphi)$.

*SREU* is, by definition, the 1-corroboration problem for Horn formulas. However, we only need to consider *guarded* Horn formulas. To see that, consider a Horn formula $\varphi$; let $\Sigma$ be its signature and let $c$ be a constant in $\Sigma$. For each variable $x$ in $\varphi$, let $\mathrm{Gr}_\Sigma(x)$ denote the following Horn clause:

$$\begin{array}{l} \{\, c' \approx c \mid c' \text{ is a constant in } \Sigma \setminus \{c\} \,\} \cup \\ \{\, f(c, \ldots, c) \approx c \mid f \text{ is a function symbol in } \Sigma \,\} \quad \Rightarrow x \approx c. \end{array}$$

This is a very simple but useful construction that was first used by Degtyarev & Voronkov to enforce certain solutions to be within a given signature. It is easy to see that, for all terms $t$,

$$\models \mathrm{Gr}_\Sigma(t) \quad \Leftrightarrow \quad t \in \mathcal{T}_\Sigma.$$

Let now $\psi$ be the guarded Horn formula

$$(\bigwedge_{x\in FV(\varphi)} \mathrm{Gr}_\Sigma(x)) \wedge \varphi.$$

From Herbrand's theorem follows that one only needs to consider corroborators in the language of $\varphi$, therefore $\psi$ has a corroborator if and only if $\varphi$ has one.

**Example 7** A simple example of a guarded Horn formula is this

$$\begin{aligned}\psi \quad = \quad & (E_1 \Rightarrow x \approx c_1) \wedge \\ & (E_2 \Rightarrow y \approx c_2) \wedge \\ & (\Pi_1 \Rightarrow x \approx y) \wedge \\ & (\Pi_2 \Rightarrow x \approx t \centerdot y)\end{aligned}$$

where $E_1$, $E_2$, $\Pi_1$, $\Pi_2$ and $t$ are ground, $c_1$, $c_2$ are constants, and '$\centerdot$' is a binary function symbol written in infix notation. A guard of $\psi$ is

$$(E_1 \Rightarrow x \approx c_1) \wedge (E_2 \Rightarrow y \approx c_2).$$

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

$$\begin{aligned}\varphi \quad = \quad & (E \Rightarrow x \centerdot y \approx c) \wedge \\ & (\Pi_1 \Rightarrow x \approx y) \wedge \\ & (\Pi_2 \Rightarrow x \approx t \centerdot y),\end{aligned}$$

where $E$, $\Pi_1$, $\Pi_2$ and $t$ are ground and $c$ is a constant. The guard of $\varphi$ is

$$E \Rightarrow x \centerdot y \approx c.$$

These formulas are of particular interest for us, see Section 6. □

We say that a corroborator of a disjunction $\varphi$ is *partisan*, if it corroborates some disjunct of $\varphi$. The main result of this section is the following theorem.

**Theorem 8 (Partisan Corroboration)** *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 $\varphi_i$'s are constant-disjoint guarded Horn formulas. Let $\theta$ be a corroborator for $\varphi$. We must prove that $\theta$ corroborates $\varphi_i$ for some $i \in I$.

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

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

where $\psi_i$ is a guard of $\varphi_i$, i.e., each $E_i^k$ and $s_i^k$ is ground and $FV(\varphi_i) = FV(\psi_i)$, for all $i \in I$. Let $C_i = Con(\varphi_i)$ 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 $\Sigma$-structures that satisfy $\varphi_i\theta$, i.e,

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

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

Let now $J$ be any subset of $I$ such that

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

So

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

To see that, suppose (by contradiction) that $Con(\varphi_i\theta)$ contains some $c \notin C_i$. Clearly, $c$ belongs to some $x\theta$ where $x$ occurs in the guard $\psi_i$. By Lemma 5, every constant in $x\theta$ belongs to $C_i$. This gives the desired contradiction.

If $I = J$ then the theorem follows by (1), (3) and Lemma 3. Assume that $I \neq J$. Below we prove the following statement:

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

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.

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

$$D = \bigcup_{i \in J} D_i^{f(i)}\theta.$$

(Note that $D = \emptyset$ if $J = \emptyset$.) 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

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

We will now prove the following statement.

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

> **Proof.** Suppose, by contradiction, that (6) does not hold. (Assume also that $J \neq \emptyset$ or else (6) holds trivially.) Then $\mathcal{A}_g \in \mathcal{K}_j$ for some $j \in J$. Fix such an appropriate $j$.

So $\mathcal{A}_g$ satisfies each clause in $\varphi_j\theta$ and in particular the following holds, call it (†):

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

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

$$Con(D', u', v') \subseteq C_j$$

and

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

So, by (1),

$$Con(D', u', v') \cap Con(E_g, D'') = \emptyset.$$

It follows, from Lemma 6 and (†), that

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

But this contradicts (5). ⊠

By using (6) we can now complete the proof of (4). Suppose, by contradiction, that there is no $i \in I \setminus J$ such that $\models \psi_i\theta$. 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 (6) we know that $\mathcal{A}_g \in \mathcal{K}_i$ for some $i \in I \setminus J$. Choose such an $i$. So $\mathcal{A}_g$ satisfies each clause in $\varphi_i\theta$, and in particular the following holds, call it (‡):

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

But, by (3),

$$Con(E_i^{g(i)}, s_i^{g(i)}) \subseteq C_i$$

and

$$\begin{array}{rcl} Con(\bigcup_{j \in I \setminus J, j \neq i} E_j^{g(j)}) \cup Con(D) & \subseteq & Con(\bigcup_{j \in I \setminus J, j \neq i} C_j) \cup Con(\bigcup_{j \in J} C_j) \\ & = & Con(\bigcup_{j \in I, j \neq i} C_j), \end{array}$$

and thus, by (1),

$$Con(E_i^{g(i)}, s_i^{g(i)}) \cap Con(\bigcup_{j \in I \setminus J, j \neq i} E_j^{g(j)}, D) = \emptyset.$$

Hence, by Lemma 6 and (‡),

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

which contradicts our choice of $g$. ☒

**Remark** Theorem 8, 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 9** 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 constants, 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 $\theta$ such that $\theta(x_1) = 1$ and $\theta(x_0) = 0$. It is easy to show by case analysis that $\theta$ corroborates $\psi$, 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, $\theta$ corroborates neither $\varphi_1$ nor $\varphi_0$.

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 $x_1, x_0, y_1, y_0, y$ are distinct variables. Let $\psi = \varphi_1 \vee \varphi_0$. Let $\theta$ 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 exactly the same as in the previous case). □

## 4 From corroboration to $m$-corroboration

As Degtyarev & Voronkov [1995] have shown, the corroboration problem is undecidable. Shortly after, Voda & Komara [1995] have shown that $m$-corroboration is undecidable for all multiplicities $m$. We show that the latter result follows easily from the former result by using the Partisan Corroboration Theorem.

**Theorem 10 (Degtyarev–Voronkov)** *Corroboration of guarded Horn formulas is undecidable.*

For technical reasons it will be convenient to use the following definitions. Given a set $V$ of variables, by a $(\Sigma, V)$-expression we mean a $\Sigma$-expression all of whose variables are in $V$. We write $\Sigma^{(n)}$ for the constant-disjoint copy of $\Sigma$ where each constant $c$ has been replaced by a unique constant $c^{(n)}$ not in $\Sigma$. Similarly, $V^{(n)}$ is a disjoint copy of $V$ where each variable $x$ has been replaced by a unique variable $x^{(n)}$ not in $V$. It is also assumed that $c^{(m)} \neq c^{(n)}$ and $x^{(m)} \neq x^{(n)}$ for $m \neq n$.

We define by induction on any $(\Sigma, V)$-expression $X$ the $(\Sigma^{(n)}, V^{(n)})$-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 $\theta$ mapping variables in $V$ to ground $\Sigma$-terms we let $\theta^{(n)}$ denote the substitution that takes the variable $x^{(n)}$ to the term $(x\theta)^{(n)}$, for $x \in V$. So, for any $(\Sigma, V)$-expression $X$ and natural number $n$,

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

The following property holds trivially. For any $(\Sigma, V)$-sentence $\varphi$ and natural number $n$,

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

**Theorem 11** *Let $\varphi$ be a guarded Horn $(\Sigma, V)$-formula and $n$ a positive integer. Then $\varphi$ has a corroborator if and only if $\bigwedge_{i=1}^{n} \varphi^{(i)}$ has an $n$-corroborator.*

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

$n$-corroborator $\{\, \theta_i \mid i \in I \,\}$. So

$$\models \bigvee_{i \in I} (\varphi^{(1)}\theta_i \wedge \cdots \wedge \varphi^{(i)}\theta_i \wedge \cdots \wedge \varphi^{(n)}\theta_i).$$

By using the distributive laws we can construct an equivalent formula in conjunctive normal form, including as one of the conjuncts the formula $\bigvee_{i \in I} \varphi^{(i)}\theta_i$. Hence

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

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

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

From the Partisan Corroboration Theorem 8 follows now that $\models \varphi^{(i)}\theta'$ for some $i \in I$. Fix such an appropriate $i$. But then, by the fact that $\varphi^{(i)}$ is guarded and using Lemma 5, it follows that the range of $\theta' \restriction V_i$ is $\mathcal{T}_{\Sigma(\varphi^{(i)})}$, and thus there is a substitution $\theta$ such that $\theta^{(i)} \restriction V_i = \theta' \restriction V_i$. Hence $\models \varphi^{(i)}\theta^{(i)}$ and so $\models \varphi\theta$. ⊠

**Theorem 12 (Voda–Komara)** *For all $n \geq 1$, $n$-corroboration is undecidable.*

**Proof.** Given $n$ and $\varphi$, the construction of $\psi$ in Theorem 11 is trivially effective. So, if we had a decision procedure (for some $n$) for deciding the existence of $n$-corroborators, we could use it to decide the existence of corroborators, but this would contradict Theorem 10. ⊠

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 multiplicity $m$. Voda–Komara theorem 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 with ground left-hand sides [Plaisted 1995], (i.e., variables occur only in positive literals in the corresponding Horn formulas), and already in the guarded case with two variables [Veanes 1996], we can sharpen the Voda-Komara theorem as follows.

**Corollary 13** *For all $n \geq 1$, $n$-corroboration is undecidable for guarded Horn formulas with $2n$ variables and ground negative literals.*

By a *monadic* signature or language we mean a signature or language where all function symbols have arity at most one. By *monadic* SREU or corroboration we understand the restriction of that decision problem to monadic languages. The decidability of monadic SREU is currently one of the difficult open problems related to SREU [Gurevich & Voronkov 1997]. An effectively equivalent problem is the decidability of the prenex fragment of intuitionistic logic with equality in monadic languages [Degtyarev & Voronkov 1996*a*]. Some evidence speaks in favor of that the problem is decidable (e.g., many subcases are decidable, see Section 8). From Theorem 11 follows that:

**Corollary 14** *If monadic corroboration is undecidable, then so is monadic $n$-corroboration for any $n > 1$, or equivalently, if monadic $n$-corroboration is decidable for some $n > 1$ then so is monadic corroboration.*

# 5 Shifted pairing

Shifted pairing is a general technique for proving undecidability results. The term shifted pairing was introduced by Plaisted [1995]. A variant of shifted pairing was used already by Hopcroft & Ullman [1979] in establishing the undecidability of the problem of testing nonemptiness of the intersection of two context free languages. Goldfarb's [1981] proof of the undecidability of second-order unification uses also similar ideas. Finite tree automata provide a suitable abstraction level for our purposes, for formalizing this technique as a decision problem of finite tree automata.

The main result of this section is the Shifted Pairing theorem. In this section we use a binary function symbol '$\centerdot$', and we write it for better readability using infix notation and assume that it associates to the right. For example, if $t_1$, $t_2$, and $t_3$ are terms, then the term $\centerdot(t_1, \centerdot(t_2, t_3))$ is written unambiguously as $t_1 \centerdot t_2 \centerdot t_3$.

**Theorem 15 (Shifted Pairing)** *One can effectively construct two constant-disjoint tree automata*

$$A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\}), \quad A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\}),$$

*and two ground and canonical rewrite systems*

$$\Pi_1 \subseteq \mathcal{T}_{\Sigma_{\mathrm{mv}}} \times \mathcal{T}_{\Sigma_{\mathrm{id}}}, \quad \Pi_2 \subseteq \mathcal{T}_{\Sigma_{\mathrm{mv}}} \times \mathcal{T}_{\Sigma_{\mathrm{id}}},$$

*such that, it is undecidable whether, given $t_0 \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, there exists $s \in \mathcal{L}(A_{\mathrm{mv}})$ and $t \in \mathcal{L}(A_{\mathrm{id}})$ such that $s \xrightarrow{*}_{\Pi_1} t$ and $t_0 \centerdot s \xrightarrow{*}_{\Pi_2} t$.*

The rest of this section is devoted to the proof of this theorem. We start by proving some lemmas. The proof of the theorem itself is given in Section 5.3.2.

We consider a fixed deterministic Turing machine $M$ with *initial state* $q_0$, *final state* $q_{\mathrm{f}}$, a *blank* symbol $\sqcup$. By $\Sigma(M)$ we denote the union of the states and tape symbols of $M$ including the blank symbol. All characters in $\Sigma(M)$ are considered to be *constants*. Moreover, $M$ is only allowed to write a blank when it erases the *last* nonblank symbol on the tape. This means that IDs do not include blanks. However, overwriting the last nonblank symbol on the tape by a blank, means erasing of the last input symbol on the tape. For such a TM $M$ we can assume, without loss of generality, that when $M$ enters the final state then its tape is empty. Given an ID $v$, we let $v^+$ denote the following string:

$$v^+ = \begin{cases} \text{successor of } v, & \text{if } v \text{ is nonfinal;} \\ \epsilon, & \text{otherwise.} \end{cases}$$

Note that the final ID of $M$ is the unique one character string $q_{\mathrm{f}}$ and $q_{\mathrm{f}}^+ = \epsilon$.

## 5.1 Words and trains

We use certain nonmonadic terms to represent strings, we call such terms words. Similarly, we use certain terms, that we call trains, to represent sequences of strings. Let $c$ and $d$ be constants.

- A term $s$ is called a *$c$-word* if either $s = c$, or $s = c_1 \centerdot s'$ for some constant $c_1$ and $c$-word $s'$.

  The *empty $c$-word* is simply the constant $c$.

- A term $t$ is called a *$d$-train of $c$-words* if either $t = d$, or $t = s \centerdot t'$ for some $c$-word $s$ and $d$-train $t'$.

  The *empty $d$-train* is simply the constant $d$.

We adopt the following convenient notation for words and trains. A $c$-word

$$c_1 \centerdot c_2 \centerdot \cdots \centerdot c_n \centerdot c$$

is written simply as

$$c_1 c_2 \cdots c_n \centerdot c$$

and is said to *represent* the string $c_1 c_2 \ldots c_n$. When we say that a *$c$-word is in a set $V$ of strings*, we mean that the string represented by that $c$-word is in $V$.

Similarly, a $d$-train

$$(v_1 \centerdot c) \centerdot (v_2 \centerdot c) \centerdot \cdots \centerdot (v_n \centerdot c) \centerdot d$$

is said to *represent* the string sequence

$$(v_1, v_2, \ldots, v_n).$$

By representing strings by words as above, one can of course easily represent arbitrary regular sets of strings by corresponding regular forests of words. We use this fact in the Train Lemma, that is our key tool in constructing the two tree automata $A_{\mathrm{mv}}$ and $A_{\mathrm{id}}$.

**Lemma 16 (Train Lemma)** *Let $V$ be a regular set of strings over a signature $\Sigma$ of constants. Let $c$ and $d$ be distinct constants not in $\Sigma$. Then the set of all $d$-trains of $c$-words in $V$ is recognized by a DTA with one final state.*

**Proof.** To begin with, let $A_1 = (Q_1, \Sigma_1, R_1, F_1)$, where $\Sigma_1 = \Sigma \cup \{\centerdot, c\}$, be a DTA that recognizes the set of all $c$-words in $V$. Next, let $p$ be a new state, $\Sigma_2 = \Sigma_1 \cup \{d\}$, and

$$A = (Q_1 \cup \{p\}, \Sigma_2, R, \{p\})$$

where

$$R = R_1 \cup \{d \to p\} \cup \{\, q \centerdot p \to p \mid q \in F_1 \,\}.$$

We prove that $A$ is a DTA satisfying the claim. Clearly, it *is* a DTA. First, we prove the equivalence of statements 1 and 2.

1. $t \in \mathcal{L}(A)$ ($t \in \mathcal{T}_{\Sigma_2}$ and $t \xrightarrow{*}_R p$)

2. $t \in \mathcal{T}_{\Sigma_2}$ and there exist $n \geq 0$ and states $q_1, q_2, \ldots, q_n \in F_1$ such that

$$t \qquad \xrightarrow{*}_{R_1} \qquad q_1 \centerdot q_2 \centerdot \cdots \centerdot q_n \centerdot d$$

$$\begin{aligned} &\longrightarrow_{\{d\to p\}} && q_1 \centerdot q_2 \centerdot \cdots \centerdot q_n \centerdot p \\ &\xrightarrow{*}_{\{\, q\centerdot p\to p \mid q\in F_1 \,\}} && p \end{aligned}$$

The direction from Statement 2 to Statement 1 is immediate. To prove the converse direction consider a reduction of $t$ in $R$ to $p$. A classical permutation argument on reductions, using the fact that $p \notin Q_1$ and $d \notin \Sigma_1$, shows that there exists a reduction where all the rules from $R_1$ appear first:

$$t \xrightarrow{*}_{R_1} t' \xrightarrow{*}_{R\setminus R_1} p.$$

It follows now, by induction on the number of rewrite steps, that any reduction of $t'$ in $R \setminus R_1$ to $p$ must be of the desired form, proving Statement 2.

Since no rule in $R$ can introduce a '$\centerdot$', the first part of the reduction in Statement 2 is equivalent to saying that there exist terms $s_1, \ldots, s_n \in \mathcal{T}_{\Sigma_2}$ such that $t = s_1 \centerdot \cdots \centerdot s_n \centerdot d$ and $s_i \xrightarrow{*}_{R_1} q_i$. From $s_i \xrightarrow{*}_{R_1} q_i$ follows that $d$ cannot occur in $s_i$, i.e., $s_1, \ldots, s_n \in \mathcal{T}_{\Sigma_1}$, and thus $s_1, \ldots, s_n \in \mathcal{L}(A_1)$. Consequently, Statement 2 is tantamount to saying that $t$ is a $d$-train of $c$-words in $V$, and the claim follows. $\boxtimes$

Let $c_{\mathrm{id}}$ and $d_{\mathrm{id}}$ be two fixed distinct constants not in $\Sigma(M)$. A *train of IDs* is a $d_{\mathrm{id}}$-train of $c_{\mathrm{id}}$-words representing IDs of $M$.

**Lemma 17** *There is a DTA* $A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\})$ *that recognizes the set of all trains of IDs, where* $\Sigma_{\mathrm{id}} = \Sigma(M) \cup \{\centerdot, c_{\mathrm{id}}, d_{\mathrm{id}}\}$.

**Proof.** The set of all IDs of $M$ is regular. Use Lemma 16. $\boxtimes$

## 5.2 Trains of moves

We now want to represent moves of $M$ in such a way that we can obtain a statement corresponding to Lemma 17 for moves. A naive encoding of a move

$(v, v^+)$ as a term $(v \mathbin{.} c) \mathbin{.} (v^+ \mathbin{.} c)$ does of course not work for several reasons, to mention one: such terms are not recognizable.

Instead, we exploit the following information. Let $(v, v^+)$ be a move, $m$ the length of $v$ and $n$ the length of $v^+$. We know that either $n = m$, $n = m + 1$ ($M$ adds a new symbol at the end of the tape contents), or $n = m - 1$ ($M$ erases the last nonblank symbol on the tape). We encode moves by strings of new characters where the $i$'th character encodes the $i$'th characters in the components of the move. We now proceed with the formal definition.

Two new constants, denoted by $\langle a, b\rangle$ and $\langle a, b\rangle'$, respectively, are introduced for every pair of constants $a$ and $b$ in $\Sigma(M)$. All these new constants are assumed to be pairwise distinct. Let $v$ be any ID of $M$ and $v^+$ its successor, say

$$\begin{aligned} v &= a_1 a_2 \cdots a_m, \\ v^+ &= b_1 b_2 \cdots b_n. \end{aligned}$$

We define $\langle v, v^+\rangle$ as the following string.

$$\langle v, v^+\rangle = \begin{cases} \langle a_1, b_1\rangle\langle a_2, b_2\rangle \cdots \langle a_{n-1}, b_{n-1}\rangle\langle \sqcup, b_n\rangle', & \text{if } m = n - 1; \\ \langle a_1, b_1\rangle\langle a_2, b_2\rangle \cdots \langle a_{m-1}, b_{m-1}\rangle\langle a_m, \sqcup\rangle', & \text{if } m = n + 1; \\ \langle a_1, b_1\rangle\langle a_2, b_2\rangle \cdots \langle a_{m-1}, b_{m-1}\rangle\langle a_m, b_m\rangle', & \text{if } m = n. \end{cases}$$

we call such a string a *move* also.

Let $c_{\mathrm{mv}}$ and $d_{\mathrm{mv}}$ be fixed distinct new constants. A *train of moves* is a $d_{\mathrm{mv}}$-train of $c_{\mathrm{mv}}$-words that represent moves.

**Lemma 18** *There is a DTA* $A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\})$ *that recognizes the set of all trains of moves, where*

$$\Sigma_{\mathrm{mv}} = \{\, \langle a, b\rangle, \langle a, b\rangle' \mid a, b \in \Sigma(M)\,\} \cup \{\mathbin{.}, c_{\mathrm{mv}}, d_{\mathrm{mv}}\}.$$

**Proof.** Follows from the Train Lemma 16 and the fact that the set of moves is regular. The important property that is exploited here is that only a fixed size substring of an ID is changed by a move.

For example, the set of all moves corresponding to computation steps that do not change the last tape symbol can be described by the following regular set of strings:

$$V^* V_\delta V^* V'$$

where $V_\delta$ is is a certain finite set of three-character or two-character strings constructed from the transition function of $M$, e.g., if $M$ upon reading the symbol $a$ in state $q$ writes the symbol $a'$, moves right, and enters state $q'$, then $\langle q, a'\rangle\langle a, q'\rangle$ is in $V_\delta$. The set $V$ consists all constants $\langle a, a\rangle$ such that $a$ is an input symbol of $M$, and $V'$ is the set of all constants $\langle a, a\rangle'$ such that $a$ is an input symbol of $M$. The other cases are similar. ⊠

## 5.3 Main construction

Given a nonempty train $t$ of moves, say

$$t = (\langle v_1, v_1^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot (\langle v_2, v_2^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot \cdots \centerdot (\langle v_{k-1}, v_{k-1}^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot (\langle v_k, v_k^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot d_{\mathrm{mv}}$$

define the *first projection* of $t$ as the following train of IDs

$$\pi_1(t) = (v_1 \centerdot c_{\mathrm{id}}) \centerdot (v_2 \centerdot c_{\mathrm{id}}) \centerdot \cdots \centerdot (v_{k-1} \centerdot c_{\mathrm{id}}) \centerdot (v_k \centerdot c_{\mathrm{id}}) \centerdot d_{\mathrm{id}}$$

and the *second projection* of $t$ as the following train

$$\pi_2(t) = \begin{cases} (v_1^+ \centerdot c_{\mathrm{id}}) \centerdot (v_2^+ \centerdot c_{\mathrm{id}}) \centerdot \cdots \centerdot (v_{k-1}^+ \centerdot c_{\mathrm{id}}) \centerdot d_{\mathrm{id}}, & \text{if } v_k = q_{\mathrm{f}}; \\ (v_1^+ \centerdot c_{\mathrm{id}}) \centerdot (v_2^+ \centerdot c_{\mathrm{id}}) \centerdot \cdots \centerdot (v_{k-1}^+ \centerdot c_{\mathrm{id}}) \centerdot (v_k^+ \centerdot c_{\mathrm{id}}) \centerdot d_{\mathrm{id}}, & \text{otherwise.} \end{cases}$$

We say that $t$ is a *shifted pairing train* if $t$ is a train of moves such that

$$\pi_1(t) = (v_1 \centerdot c_{\mathrm{id}}) \centerdot \pi_2(t)$$

and we refer to $v_1$ as the *first ID* of $t$. Recall that $q_0$ is the initial state of $M$.

**Lemma 19** *Let $v_0$ be an input string for $M$. Then there exists a shifted pairing train $t$ with first ID $q_0v_0$ if and only if $M$ accepts $v_0$.*

**Proof.** Let $v_0$ be given and $t$ a train of moves as above, with $v_1 = q_0v_0$. If $t$ is a shifted pairing train then the second projection must be shorter than the first one, and thus $v_k = q_{\mathrm{f}}$ and

$$\begin{array}{ccccccccccccc} ( & v_1 & , & v_2 & , & v_3 & , & \ldots & , & v_{k-1} & , & v_k & ) \; = \\ ( & q_0v_0 & , & v_1^+ & , & v_2^+ & , & \ldots & , & v_{k-2}^+ & , & v_{k-1}^+ & ) \end{array}$$

which is tantamount to saying that the first projection of $t$ represents a valid computation of $M$ with input $v_0$, i.e., $M$ accepts $v_0$. The proof of the converse direction is similar. ⊠

### 5.3.1 The rewrite systems $\Pi_1$ and $\Pi_2$

The system $\Pi_1$ contains all the following rules:

1. For all $a, b \in \Sigma(M)$, the rule $\langle a, b\rangle \to a$.

2. For all $a, b \in \Sigma(M)$ such that $a \neq \sqcup$, the rule $\langle a, b\rangle' \centerdot c_{\mathrm{mv}} \to a \centerdot c_{\mathrm{id}}$.

3. For all $b \in \Sigma(M)$, the rule $\langle \sqcup, b\rangle' \centerdot c_{\mathrm{mv}} \to c_{\mathrm{id}}$.

4. The rule $d_{\mathrm{mv}} \to d_{\mathrm{id}}$.

**Lemma 20** *The rewrite system $\Pi_1$ is canonical and $\Pi_1 \subseteq \mathcal{T}_{\Sigma_{\mathrm{mv}}} \times \mathcal{T}_{\Sigma_{\mathrm{id}}}$.*

**Proof.** It is easy to check that the rules in $\Pi_1$ form a reduced set of rules and $\Pi_1$ is therefore canonical. ⊠

Hence, we have the following relation between $\Pi_1$ and the notion of first projection.

**Lemma 21** *For all trains $s$ of moves and all trains $t$ of IDs, $s \xrightarrow{*}_{\Pi_1} t$ if and only if $t = \pi_1(s)$.*

**Proof.** Let $s$ and $t$ be given. By Lemma 20 $t$ is irreducible in $\Pi_1$ because $\Sigma_{\mathrm{mv}}$ and $\Sigma_{\mathrm{id}}$ are constant-disjoint. So, $s \xrightarrow{*}_{\Pi_1} t$ if and only if $s{\downarrow}_{\Pi_1} = t$. It remains to be checked that indeed $s{\downarrow}_{\Pi_1} = \pi_1(s)$, which is straightforward. ☒

The system $\Pi_2$ contains all the following rules:

1. For all $a, b \in \Sigma(M)$, the rule $\langle a, b\rangle \to b$.

2. For all $a, b \in \Sigma(M)$ such that $b \neq \sqcup$, the rule $\langle a, b\rangle' \centerdot c_{\mathrm{mv}} \to b \centerdot c_{\mathrm{id}}$.

3. For all $a \in \Sigma(M)$ such that $a \neq q_{\mathrm{f}}$, the rule $\langle a, \sqcup\rangle' \centerdot c_{\mathrm{mv}} \to c_{\mathrm{id}}$.

4. The rule $(\langle q_{\mathrm{f}}, \sqcup\rangle' \centerdot c_{\mathrm{mv}}) \centerdot d_{\mathrm{mv}} \to d_{\mathrm{id}}$.

Again, one can easily check that the rules in 1–4 form a reduced rule set.

**Lemma 22** *The rewrite system $\Pi_2$ is canonical and $\Pi_2 \subseteq \mathcal{T}_{\Sigma_{\mathrm{mv}}} \times \mathcal{T}_{\Sigma_{\mathrm{id}}}$.*

**Lemma 23** *For all trains $s$ of moves and all IDs $v$, $(v \centerdot c_{\mathrm{id}}) \centerdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$ if and only if $s$ is a shifted pairing train with first ID $v$.*

**Proof.** Let $s$ and $v$ be given and Assume that $(v \centerdot c_{\mathrm{id}}) \centerdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$. Say

$$s = (\langle v_1, v_1^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot \cdots \centerdot (\langle v_{k-1}, v_{k-1}^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot (\langle v_k, v_k^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot d_{\mathrm{mv}}.$$

So

$$\pi_1(s) = (v_1 \centerdot c_{\mathrm{id}}) \centerdot (v_2 \centerdot c_{\mathrm{id}}) \centerdot \cdots \centerdot (v_{k-1} \centerdot c_{\mathrm{id}}) \centerdot (v_k \centerdot c_{\mathrm{id}}) \centerdot d_{\mathrm{id}}.$$

By Lemma 22, $((v \centerdot c_{\mathrm{id}}) \centerdot s){\downarrow}_{\Pi_2} = \pi_1(s)$, and thus $v_1 = v$, and

$$s{\downarrow}_{\Pi_2} = (v_2 \centerdot c_{\mathrm{id}}) \centerdot \cdots \centerdot (v_{k-1} \centerdot c_{\mathrm{id}}) \centerdot (v_k \centerdot c_{\mathrm{id}}) \centerdot d_{\mathrm{id}}.$$

On the other hand, from the definition of $\Pi_2$ follows that all the rules can be applied simultaneously, which implies that

$$(\langle v_i, v_i^+\rangle \centerdot c_{\mathrm{mv}}){\downarrow}_{\Pi_2} = v_{i+1} \centerdot c_{\mathrm{id}} \quad \text{for } 1 \leq i < k, \tag{7}$$

and the last word of $s$ must be removed:

$$((\langle v_k, v_k^+\rangle \centerdot c_{\mathrm{mv}}) \centerdot d_{\mathrm{mv}}){\downarrow}_{\Pi_2} = d_{\mathrm{id}}. \tag{8}$$

From (7) follows that $v_i^+ = v_{i+1}$ for $1 \leq i < k$ and from (8) follows that the rule $(\langle q_{\mathrm{f}}, \sqcup\rangle' \centerdot c_{\mathrm{mv}}) \centerdot d_{\mathrm{mv}} \to d_{\mathrm{id}}$ is used, and thus $v_k = q_{\mathrm{f}}$. Hence, $s$ is a shifted pairing train with first ID $v$.

The proof of the converse direction is analogous. ⊠

### 5.3.2 Proof of the Shifted Pairing Theorem

**Proof.** Let $M$ in the above construction be a universal Turing machine. Let $A_{\mathrm{id}}$ and $A_{\mathrm{mv}}$ be constant-disjoint DTAs given by Lemma 17 and Lemma 18, respectively. The claim in Theorem 15 is a consequence of the equivalence of the following statements. The additional conditions on the rewrite systems $\Pi_1$ and $\Pi_2$ follow from Lemma 20 and Lemma 22.

1. $M$ accepts $v_0$.
2. There exists $s \in \mathcal{L}(A_{\mathrm{mv}})$ such that $(q_0 v_0 \centerdot c_{\mathrm{id}}) \centerdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$.
3. There exist $s \in \mathcal{L}(A_{\mathrm{mv}})$ and $t \in \mathcal{L}(A_{\mathrm{id}})$, such that $s \xrightarrow{*}_{\Pi_1} t$ and $(q_0 v_0 \centerdot c_{\mathrm{id}}) \centerdot s \xrightarrow{*}_{\Pi_2} t$.

The statements 1 and 2 are equivalent by Lemma 19 and Lemma 23. The statements 2 and 3 are equivalent by Lemma 21. ⊠

See Figure 2 for a concrete example.

# 6 Applications of Partisan Corroboration Theorem and Shifted Pairing Theorem

The Shifted Pairing Theorem is used here to give a very elementary undecidability proof of SREU. The latter result is then used, in combination with the Partisan Corroboration Theorem to improve upon the undecidability result of $n$-corroboration for arbitrary $n$.

## 6.1 Undecidability of SREU: minimal case

Consider fixed constant-disjoint DTAs $A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\})$ and $A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\})$, a binary function symbol $\centerdot$, and ground canonical rewrite systems $\Pi_1$ and $\Pi_2$ given by the Shifted Pairing Theorem 15. Let $q$ be a new state and $A$ the tree automaton $(Q, \Sigma, R, F)$, where

$$\begin{aligned} Q &= Q_{\mathrm{mv}} \cup Q_{\mathrm{id}} \cup \{q\}, \\ \Sigma &= \Sigma_{\mathrm{mv}} \cup \Sigma_{\mathrm{id}}, \\ R &= R_{\mathrm{mv}} \cup R_{\mathrm{id}} \cup \{q_{\mathrm{mv}} \centerdot q_{\mathrm{id}} \to q\}, \\ F &= \{q\}. \end{aligned}$$

Obviously, $A$ is still a *deterministic* tree automaton, because $A_{\mathrm{mv}}$ and $A_{\mathrm{id}}$ are constant-disjoint and deterministic. We have the following property as a direct consequence of the constant-disjointness of $A_{\mathrm{id}}$ and $A_{\mathrm{mv}}$.

**Lemma 24** *For all ground terms $s$ and $t$, $s \centerdot t \xrightarrow{*}_R q$ if and only if $s \xrightarrow{*}_{R_{\mathrm{mv}}} q_{\mathrm{mv}}$ and $t \xrightarrow{*}_{R_{\mathrm{id}}} q_{\mathrm{id}}$.*

We can now prove the following result. Recall that a *rigid equation* is an expression $E \vdash^{\mathrm{r}} s \approx t$, where $E$ is a finite set of equations and $s$ and $t$ are terms.

A ground substitution $\theta$ *solves* $E \vdash^{\mathrm{r}} s \approx t$ if $\theta$ corroborates $E \Rightarrow s \approx t$. *SREU* is the problem of deciding if there exists a $\theta$ that solves all members in a given finite set of rigid equations.

**Theorem 25** *There is an integer $n$, such that SREU is undecidable under the following restrictions:*

1. **(Plaisted)** *the left-hand sides are ground,*
2. *the left-hand sides have at most $n$ symbols,*
3. *there are at most two variables each occurring at most three times, and*
4. *there are at most three rigid equations.*

**Proof.** Let $S_{t_0}(x, y)$ be the following system of rigid equations where $t_0$ is a given ground term over $\Sigma_{\mathrm{id}}$.

$$S_{t_0}(x,y) = \begin{cases} R^{\approx} & \vdash^{\mathrm{r}} \quad x \mathbin{\centerdot} y \approx q \\ \Pi_1^{\approx} & \vdash^{\mathrm{r}} \quad x \approx y \\ \Pi_2^{\approx} & \vdash^{\mathrm{r}} \quad t_0 \mathbin{\centerdot} x \approx y \end{cases}$$

Let $\theta$ be a ground substitution with $x\theta = s$ and $y\theta = t$. Since all the left-hand sides are canonical rewrite systems, by using Birkhoff's theorem, we get that $\theta$ solves $S_{t_0}(x, y)$ if and only if

$$(s \mathbin{\centerdot} t){\downarrow_R} = q{\downarrow_R}, \quad s{\downarrow_{\Pi_1}} = t{\downarrow_{\Pi_1}}, \quad t_0 \mathbin{\centerdot} s{\downarrow_{\Pi_2}} = t{\downarrow_{\Pi_2}}.$$

By using Lemma 24 and that $q$ is irreducible in $R$, this is equivalent to

$$s{\downarrow_{R_{\mathrm{mv}}}} = q_{\mathrm{mv}}, \quad t{\downarrow_{R_{\mathrm{id}}}} = q_{\mathrm{id}}, \quad s{\downarrow_{\Pi_1}} = t{\downarrow_{\Pi_1}}, \quad t_0 \mathbin{\centerdot} s{\downarrow_{\Pi_2}} = t{\downarrow_{\Pi_2}}. \qquad (9)$$

The first two facts in (9) imply that $s \in \mathcal{T}_{\Sigma_{\mathrm{mv}} \cup Q_{\mathrm{mv}}}$ and $t \in \mathcal{T}_{\Sigma_{\mathrm{id}} \cup Q_{\mathrm{id}}}$. In particular, $s$ and $t$ are constant-disjoint. At the same time, $\Pi_1^{\approx} \models s \approx t$ implies (see Lemma 5) that $Con(s) \subseteq Con(\Pi_1, t)$ and $Con(t) \subseteq Con(\Pi_1, s)$. Hence $Con(s, t) \subseteq Con(\Pi_1) \subseteq \Sigma_{\mathrm{id}} \cup \Sigma_{\mathrm{mv}}$. So (9) implies that $s \in \mathcal{T}_{\Sigma_{\mathrm{mv}}}$ and $t \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, and therefore (9) is equivalent to

$$s \in \mathcal{L}(A_{\mathrm{mv}}), \quad t \in \mathcal{L}(A_{\mathrm{id}}), \quad s{\downarrow_{\Pi_1}} = t{\downarrow_{\Pi_1}}, \quad t_0 \centerdot s{\downarrow_{\Pi_2}} = t{\downarrow_{\Pi_2}}. \tag{10}$$

But $t$ is irreducible in both $\Pi_1$ and $\Pi_2$, so (10) is equivalent to

$$s \in \mathcal{L}(A_{\mathrm{mv}}), \quad t \in \mathcal{L}(A_{\mathrm{id}}), \quad s \xrightarrow{*}_{\Pi_1} t, \quad t_0 \centerdot s \xrightarrow{*}_{\Pi_2} t. \tag{11}$$

By the Shifted Pairing Theorem 15, the problem of existence of such $s$ and $t$ for a given $t_0$ is undecidable, and thus, so is the solvability of $S_{t_0}(x, y)$ for a given $t_0$. The additional conditions are simply properties of $S_{t_0}(x, y)$, and $n$ can be chosen to be any integer greater than the number of symbols in the left-hand sides of the rigid equations in $S_{t_0}(x, y)$. ⊠

### Undecidability proofs of SREU

Degtyarev & Voronkov's [1995] original proof of the undecidability of SREU was by reduction of Baaz's [1993] monadic semi-unification problem. This proof was followed by other proofs by Degtyarev & Voronkov, first by reducing second-order unification to SREU [1996*c*], and then by reducing Hilbert's tenth problem to SREU [1996*b*]. The undecidability of second-order unification was proved by Goldfarb [1981]. Plaisted [1995] reduced Post's Correspondence Problem to SREU. From his proof follows that SREU is undecidable already with ground left-hand sides. Veanes [1996] improved that construction by using the halting problem for Turing machines and showed that two variables and one binary function symbol is enough to obtain undecidability. Here we have shown that, in addition, already three rigid equations suffice for the undecidability.

### 6.2 Undecidability of $m$-corroboration: minimal case

Consider the system $S_{t_0}(x, y)$ of rigid equations in the proof of Theorem 25 and let $\varphi_{t_0}$ denote the corresponding guarded Horn formula:

$$(R^{\approx} \Rightarrow x \cdot y \approx c) \wedge (\Pi_1^{\approx} \Rightarrow x \approx y) \wedge (\Pi_2^{\approx} \Rightarrow t_0 \cdot x \approx y).$$

A formula is *ground negative* if all negatively occurring atoms in it are ground. For example $\varphi_{t_0}$ is ground negative.

**Theorem 26** *For all $m \geq 1$, $m$-corroboration is undecidable for ground negative guarded Horn formulas with at most $2m$ variables, and at most $3m$ clauses.*

**Proof.** Given $m$ and $t_0$, construct the formula $\psi = \bigwedge_{1 \leq i \leq m} \varphi_{t_0}^{(i)}$. By Theorem 11, $\psi$ has an $m$-corroborator if and only if $\varphi_{t_0}$ has a corroborator. The rest follows from Theorem 25. $\boxtimes$

## 7 Relations to intuitionistic logic

The decision problems in intuitionistic logic have not been as thoroughly studied as the corresponding problems in classical logic [Börger, Grädel & Gurevich 1997]. In particular, new results about the *prenex fragment* of intuitionistic logic (i.e., closed prenex formulas that are intuitionistically provable), have been obtained recently by Degtyarev & Voronkov in [1996*b*, 1996*c*, 1996*a*] and Voronkov [1996]. Some of these results are:

1. Decidability, and in particular PSPACE-completeness, of the prenex fragment of intuitionistic logic *without* equality [Degtyarev & Voronkov 1996*a*].

2. Prenex fragment of intuitionistic logic *with* equality but *without* function

symbols is PSPACE-complete [Degtyarev & Voronkov 1996*a*]. Decidability of this fragment was proved by Orevkov [1976].

3. Prenex fragment of intuitionistic logic with equality in the language with one unary function symbol is decidable [Degtyarev & Voronkov 1996*a*].

4. $\exists^*$-fragment of intuitionistic logic with equality is undecidable [Degtyarev & Voronkov 1996*b*, Degtyarev & Voronkov 1996*c*].

In some of the above results, the corresponding result has first been obtained for a fragment of SREU with similar restrictions. The undecidability of the $\exists^*$-fragment is improved by Veanes [1996] by showing that already the

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

Given a rigid equation $E \vdash^{\mathrm{r}} s \approx t$, let $\mathcal{F}(E \vdash^{\mathrm{r}} s \approx t)$ denote the following implication in intuitionistic logic with equality:

$$(\bigwedge_{e\in E} e) \Rightarrow s \approx t$$

Given a system $S$ of rigid equations, let $\mathcal{F}(S)$ denote the conjunction:

$$\bigwedge_{E\vdash^{\mathrm{r}} s\approx t\in S} \mathcal{F}(E \vdash^{\mathrm{r}} s \approx t).$$

Provability in intuitionsitic logic with equality is related to SREU through the following lemma [Degtyarev & Voronkov 1996*c*].

**Lemma 27** *A system $S(\vec{x})$ of rigid equations is solvable if and only if the formula $\exists\vec{x}\mathcal{F}(S(\vec{x}))$ is provable in intuitionistic logic with equality.*

By using Theorem 25 and Lemma 27, we obtain the following sharpening of the result in [Veanes 1996].

**Corollary 28** *There is an integer $n$ such that the $\exists\exists$-fragment of intuitionistic logic with equality is undecidable already under the following restrictions:*

1. *The only logical connectives are $\wedge$ and at most three $\Rightarrow$'s.*

2. *The antecedents of all implications are ground and have less than $n$ symbols.*

In contrast, Degtyarev, Gurevich, Narendran, Veanes & Voronkov [1998*b*] have shown that the

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

Note that the statements 5 and 6 imply a complete classification of the decidability of the prenex fragment of intuitionistic logic with equality in terms of the quantifier prefix.

## 7.1 Other fragments

Decidability problems for other fragments of intuitionistic logic have been studied by Orevkov in [1965, 1976], Mints [1967], Statman [1979], and Lifschitz [1967]. Orevkov [1965] proves that the $\neg\neg\forall\exists$-fragment of intuitionistic logic with function symbols is undecidable. Lifschitz [1967] proves that intuitionistic logic with equality and without function symbols is undecidable, i.e., that the pure constructive theory of equality is undecidable. Orevkov [1976] shows decidability of some fragments (that are close to the prenex fragment) of intuitionistic logic with equality. Statman [1979] proves that the intuitionistic propositional logic is PSPACE-complete.

### 7.2 A remark about proof search

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, which is clearly illustrated by Lemma 27. Voronkov considers in [1996] a particular sequent calculus based proof system $\mathrm{LJ}^{\approx}$. A part of that system is shown in Figure 3. A *proof skeleton* in $\mathrm{LJ}^{\approx}$ is obtained from a proof in $\mathrm{LJ}^{\approx}$ by erasing all sequents and keeping only a tree decorated with rule names. See Figure 4.

*Skeleton instantiation* is the decision problem of the existence of a proof of a given formula with a given (proof) skeleton. Voronkov [1996] shows that SREU is polynomial time equivalent to skeleton instantiation in $\mathrm{LJ}^{\approx}$. So in particular, the skeleton instantiation problem in $\mathrm{LJ}^{\approx}$ is undecidable. Lemma 27 and the system of rigid equations constructed in the proof of Theorem 25 can be used to exhibit a *fixed* skeleton for which the skeleton instantiation problem is undecidable. Such a "universal" skeleton is illustrated in Figure 4.

## 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 by Gallier, Narendran, Plaisted & Snyder [1988]. De Kogel [1995] has presented a simpler proof, without computational complexity considerations. We start with the **solved cases**:

- Rigid $E$-unification with ground left-hand side is NP-complete [Kozen 1981]. Rigid $E$-unification in general is NP-complete and there exist finite complete sets of unifiers [Gallier, Narendran, Plaisted & Snyder 1990, Gallier et al. 1988]. (Here completeness has a very special meaning, differing from its meaning in the context of (non-rigid) $E$-unification.)

- Rigid $E$-unification with one variable, or, more generally, SREU with one variable and a *fixed* number of rigid equations is P-complete [Degtyarev et al. 1998*b*].

- If all function symbols have arity $\leq 1$ (the *monadic* case) then it follows that SREU is PSPACE-hard [Goubault 1994]. If only one unary function symbol is allowed then the problem is decidable [Degtyarev, Matiyasevich & Voronkov 1996]. If only constants are allowed then the problem is NP-complete [Degtyarev, Matiyasevich & Voronkov 1996] assuming that 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 if and only if it is decidable with just 2 unary function symbols [Degtyarev, Matiyasevich & Voronkov 1996].

- If the left-hand sides are ground then the monadic case is decidable [Gurevich & Voronkov 1997], and in fact PSPACE-complete [Cortier, Ganzinger, Jacquemard & Veanes 1999]. A more general problem is shown to be decidable in [Ganzinger et al. 1998]. Monadic SREU with one variable is PSPACE-complete [Gurevich & Voronkov 1997].

- The word equation solving [Makanin 1977], which is an extremely hard problem, can be reduced to monadic SREU [Degtyarev, Matiyasevich & Voronkov 1996].

- Monadic SREU is equivalent to a non-trivial extension of word equations [Gurevich & Voronkov 1997].

- Monadic SREU is equivalent to the decidability problem of the prenex fragment of intuitionistic logic with equality with function symbols of arity $\leq 1$ [Degtyarev & Voronkov 1996*a*].

- In general SREU is undecidable [Degtyarev & Voronkov 1995]. Moreover, SREU is undecidable under the following restrictions:

  - The left-hand sides of the rigid equations are ground [Plaisted 1995].

  - Furthermore, there are only two variables [Veanes 1996] and three rigid equations with ground left-hand sides of bounded size.

- SREU with one variable is decidable, in fact EXPTIME-complete [Degtyarev et al. 1998*b*]. Further decidable cases are proved in [Degtyarev, Gurevich, Narendran, Veanes & Voronkov 1998*a*] and [Cortier et al. 1999].

- SREU is polynomial time equivalent with second-order unification [Levy 1998, Veanes 1998].

The **unsolved cases** are:

- Decidability of monadic SREU.

- Decidability of SREU with two rigid equations.

Both problems are highly non-trivial. An intriguing problem is also the corroboration problem with a given strategy. In particular, the following open problem was posed by Voronkov [1997]:

- Does there exist a computable strategy $f$ with which the corroboration problem is decidable?

Further problems related to SREU and the Herbrand theorem are discussed in [Voronkov 1998*b*, Voronkov 1998*a*].

## Acknowledgements

We wish to thank Andrei Voronkov and Anatoli Degtyarev for many valuable discussions. We thank Florent Jacquemard for useful comments on a preliminary version of this paper. The comments and suggestions of an anonymous referee greatly improved the final manuscript.

# References

Baaz, M. (1993), Note on the existence of most general semi-unifiers, *in* 'Arithmetic, Proof Theory and Computation Complexity', Vol. 23 of *Oxford Logic Guides*, Oxford University Press, pp. 20–29.

Birkhoff, G. (1935), 'On the structure of abstract algebras', *Proc. Cambridge Phil. Soc.* **31**, 433–454.

Börger, E., Grädel, E. & Gurevich, Y. (1997), *The Classical Decision Problem*, Springer Verlag.

Chang, C. & Keisler, H. (1990), *Model Theory*, third edn, North-Holland, Amsterdam.

Cortier, V., Ganzinger, H., Jacquemard, F. & Veanes, M. (1999), Decidable fragments of simultaneous rigid reachability, submitted to *ICALP'99*.

De Kogel, E. (1995), Rigid $E$-unification simplified, *in* P. Baumgartner, R. Hähnle & J. Posegga, eds, 'Theorem Proving with Analytic Tableaux and Related Methods', number 918 *in* 'Lecture Notes in Artificial Intelligence', Schloß Rheinfels, St. Goar, Germany, pp. 17–30.

Degtyarev, A. & Voronkov, A. (1995), Simultaneous rigid $E$-unification is undecidable, UPMAIL Technical Report 105, Uppsala University, Computing Science Department.

Degtyarev, A. & Voronkov, A. (1996*a*), Decidability problems for the prenex fragment of intuitionistic logic, *in* 'Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)', IEEE Computer Society Press, New Brunswick, NJ, pp. 503–512.

Degtyarev, A. & Voronkov, A. (1996*b*), Simultaneous rigid $E$-unification is undecidable, *in* H. Kleine Büning, ed., 'Computer Science Logic. 9th International Workshop, CSL'95', Vol. 1092 of *Lecture Notes in Computer Science*, Paderborn, Germany, September 1995, pp. 178–190.

Degtyarev, A. & Voronkov, A. (1996*c*), 'The undecidability of simultaneous rigid $E$-unification', *Theoretical Computer Science* **166**(1–2), 291–300.

Degtyarev, A., Gurevich, Y. & Voronkov, A. (1996), Herbrand's theorem and equational reasoning: Problems and solutions, *in* 'Bulletin of the European Association for Theoretical Computer Science', Vol. 60. The "Logic in Computer Science" column.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M. & Voronkov, A. (1998*a*), 'Decidability and complexity of simultaneous rigid $E$-unification with one variable and related results', *Theoretical Computer Science*. To appear.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M. & Voronkov, A. (1998*b*), The decidability of simultaneous rigid $E$-unification with one variable, *in* T. Nipkow, ed., 'Rewriting Techniques and Applications', Vol. 1379 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 181–195.

Degtyarev, A., Matiyasevich, Y. & Voronkov, A. (1996), Simultaneous rigid $E$-unification and related algorithmic problems, *in* 'Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)', IEEE Computer Society Press, New Brunswick, NJ, pp. 494–502.

Dershowitz, N. & Jouannaud, J.-P. (1990), Rewrite systems, *in* J. Van Leeuwen, ed., 'Handbook of Theoretical Computer Science', Vol. B: Formal Methods and Semantics, North Holland, Amsterdam, chapter 6, pp. 243–309.

Gallier, J., Narendran, P., Plaisted, D. & Snyder, W. (1988), Rigid $E$-unification is NP-complete, *in* 'Proc. IEEE Conference on Logic in Computer Science (LICS)', IEEE Computer Society Press, pp. 338–346.

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

Gallier, J., Raatz, S. & Snyder, W. (1987), Theorem proving using rigid $E$-unification: Equational matings, *in* 'Proc. IEEE Conference on Logic in Computer Science (LICS)', IEEE Computer Society Press, pp. 338–346.

Ganzinger, H., Jacquemard, F. & Veanes, M. (1998), Rigid reachability, *in* J. Hsiang & A. Ohori, eds, 'Advances in Computing Science – ASIAN'98, 4th Asian Computing Science Conference, Manila, The Philippines, December 1998, Proceedings', Vol. 1538 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 4–21.

Goldfarb, W. (1981), 'The undecidability of the second-order unification problem', *Theoretical Computer Science* **13**, 225–230.

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

Gurevich, Y. & Voronkov, A. (1997), Monadic simultaneous rigid $E$-unification and related problems, *in* P. Degano, R. Corrieri & A. Marchetti-Spaccamella, eds, 'Automata, Languages and Programming, 24th International Colloquium, ICALP'97', Vol. 1256 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 154–165.

Herbrand, J. (1972), *Logical Writings*, Harvard University Press.

Hopcroft, J. E. & Ullman, J. D. (1979), *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Publishing Co.

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

Levy, J. (1998), Decidable and undecidable second-order unification problems, *in* T. Nipkow, ed., 'Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March/April 1998, Proceedings', Vol. 1379 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 47–60.

Levy, J. & Veanes, M. (1998), On unification problems in restricted second-order languages, *in* 'Annual Conference of the European Association for Computer Science Logic (CSL'98), Brno, Czech Republic'.

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

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

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

Orevkov, V. (1965), '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.

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

Plaisted, D. (1995), Special cases and substitutes for rigid $E$-unification, Technical Report MPI-I-95-2-010, Max-Planck-Institut für Informatik.

Snyder, W. (1989), 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 & J. Hartmanis, eds, 'Rewriting Techniques and Applications', Vol. 355 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 419–433.

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

Veanes, M. (1996), Uniform representation of recursively enumerable sets with simultaneous rigid $E$-unification, UPMAIL Technical Report 126, Uppsala University, Computing Science Department.

Veanes, M. (1997), The undecidability of simultaneous rigid $E$-unification with two variables, *in* 'Proc. Kurt Gödel Colloquium KGC'97', Vol. 1289 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 305–318.

Veanes, M. (1998), The relation between second-order unification and simultaneous rigid $E$-unification, *in* 'Proc. Thirteenth Annual IEEE Symposium on Logic in Computer Science, June 21–24, 1998, Indianapolis, Indiana (LICS'98)', IEEE Computer Society Press, pp. 264–275.

Voda, P. & Komara, J. (1995), On Herbrand skeletons, Technical report, Institute of Informatics, Comenius University Bratislava. Revised January 1996.

Voronkov, A. (1996), Proof search in intuitionistic logic with equality, or back to simultaneous rigid $E$-unification, *in* M. McRobbie & J. Slaney, eds, 'Automated Deduction — CADE-13', Vol. 1104 of *Lecture Notes in Computer Science*, New Brunswick, NJ, USA, pp. 32–46.

Voronkov, A. (1997), Strategies in rigid-variable methods, *in* M. Pollack, ed., 'Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)', Vol. 1, Nagoya, Japan, pp. 114–119.

Voronkov, A. (1998*a*), Herbrand's theorem, automated reasoning and semantic tableaux, *in* 'Proc. Thirteenth Annual IEEE Symposium on Logic in Computer Science, June 21–24, 1998, Indianapolis, Indiana (LICS'98)', IEEE Computer Society Press, pp. 252–263.

Voronkov, A. (1998*b*), 'Simultaneous rigid $E$-unification and other decision problems related to Herbrand's theorem', *Theoretical Computer Science*. Article after invited talk at *LFCS'97*.

# Legends of figures

**Figure 1:** Shifted pairing. Each term $t$ recognized by $A_{\mathrm{id}}$ represents a sequence of *IDs* of $M$. Each term $s$ recognized by $A_{\mathrm{mv}}$ represents a sequence of *moves* of $M$. If $s$ reduces in $\Pi_1$ to $t$ then the *first projection* of $s$ coincides with $t$. Moreover, if $s$ reduces in $\Pi_2$ to the *tail* of $t$, then the *second projection* of $s$ coincides with the tail of $t$.

**Figure 2:** Example of shifted pairing. Consider a Turing machine $M$ that, given an empty input string, writes two 0's and then simply erases them. A valid computation of $M$ can have the form $(q_0, 0q_1, 00q_2, 0q_30, q_40, q_{\mathrm{f}})$. The corresponding train of moves is the middle tree in the figure, say $s$, with the tree above $s$ as the first projection of $s$, and the tree below $s$ as the second projection of $s$. The arrows illustrate in a precise way, how $\Pi_1$ and $\Pi_2$ reduce $s$ to its first projection and second projection, respectively.

**Figure 3:** The propositional and quantifier inference rules of $\mathrm{LJ}^{\approx}$. Here $\Gamma$ and $\Delta$ are multisets of (side) formulas. In the rules $(\exists \rightarrow)$ and $(\rightarrow \forall)$ the variable $y$ does not occur free in the conclusions of the rules.

**Figure 4:** The upper part of the figure shows a possible proof tree in $\mathrm{LJ}^{\approx}$ of the formula $\exists x\, \exists y\, (\varphi_1 \Rightarrow e_1(x,y)) \wedge (\varphi_2 \Rightarrow e_2(x,y)) \wedge (\varphi_3 \Rightarrow e_3(x,y))$. This formula denotes the formula $\exists x\, \exists y\, \mathcal{F}(S_{t_0}(x,y))$ where $S_{t_0}(x,y)$ is the system of rigid equations taken from the proof of Theorem 25. Here the $\Delta_i$'s are multisets of equations, the $\mathcal{D}_i$'s are subproofs consisting of inference rules involving equality only, and each $n_i$ is the size of $\varphi_i$.

The corresponding proof skeleton is shown in the lower part of the figure.

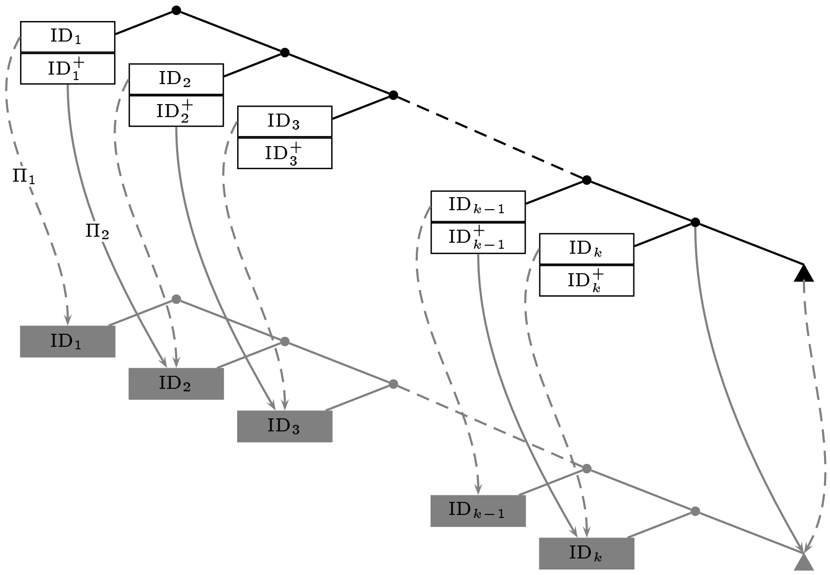
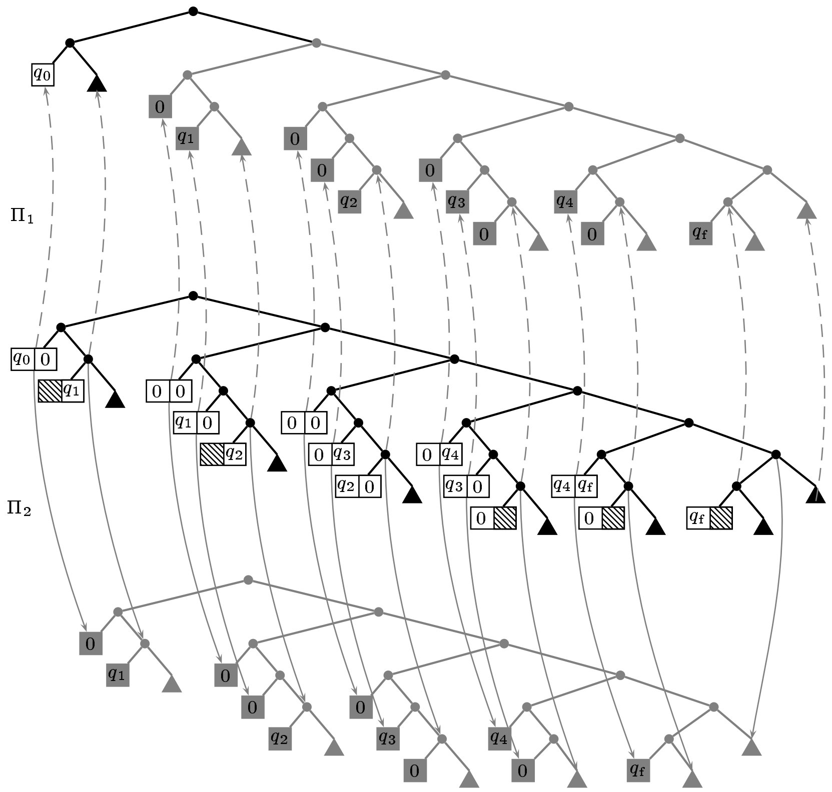

$\mathrm{ID}_1$
$\mathrm{ID}_1^+$
$\mathrm{ID}_2$
$\mathrm{ID}_2^+$
$\mathrm{ID}_3$
$\mathrm{ID}_3^+$
$\mathrm{ID}_{k-1}$
$\mathrm{ID}_{k-1}^+$
$\mathrm{ID}_k$
$\mathrm{ID}_k^+$
$\Pi_1$
$\Pi_2$
$\mathrm{ID}_1$
$\mathrm{ID}_2$
$\mathrm{ID}_3$
$\mathrm{ID}_{k-1}$
$\mathrm{ID}_k$

Figure 1:



$q_0$
0
$q_1$
0
$q_2$
0
$q_3$
0
$q_4$
0
$q_f$
$\Pi_1$
$q_0$ 0
▨ $q_1$
0 0
$q_1$ 0
▨ $q_2$
0 0
0 $q_3$
$q_2$ 0
0 $q_4$
$q_3$ 0
0 ▨
$q_4$ $q_f$
0 ▨
$q_f$ ▨
$\Pi_2$
0
$q_1$
0
0
$q_2$
0
$q_3$
0
$q_4$
0
$q_f$

Figure 2:

$$\frac{\Gamma, \phi, \psi, \Delta \rightarrow \chi}{\Gamma, \phi \wedge \psi, \Delta \rightarrow \chi}\ (\wedge \rightarrow) \qquad \frac{\Gamma \rightarrow \phi \quad \Gamma \rightarrow \psi}{\Gamma \rightarrow \phi \wedge \psi}\ (\rightarrow \wedge)$$

$$\frac{\Gamma, \phi, \Delta \rightarrow \chi \quad \Gamma, \psi, \Delta \rightarrow \chi}{\Gamma, \phi \vee \psi, \Delta \rightarrow \chi}\ (\vee \rightarrow) \qquad \frac{\Gamma \rightarrow \phi}{\Gamma \rightarrow \phi \vee \psi}\ (\rightarrow \vee_1)$$

$$\frac{\Gamma, \psi, \Delta \rightarrow \chi \quad \Gamma, \phi \Rightarrow \psi, \Delta \rightarrow \phi}{\Gamma, \phi \Rightarrow \psi, \Delta \rightarrow \chi}\ (\Rightarrow\rightarrow) \qquad \frac{\Gamma \rightarrow \psi}{\Gamma \rightarrow \phi \vee \psi}\ (\rightarrow \vee_2)$$

$$\frac{\phi, \Gamma \rightarrow \psi}{\Gamma \rightarrow \phi \Rightarrow \psi}\ (\rightarrow\Rightarrow)$$

$$\frac{\Gamma, \phi\{x \mapsto t\}, \forall x \phi, \Delta \rightarrow \chi}{\Gamma, \forall x \phi, \Delta \rightarrow \chi}\ (\forall \rightarrow) \qquad \frac{\Gamma \rightarrow \phi\{x \mapsto y\}}{\Gamma \rightarrow \forall x \phi}\ (\rightarrow \forall)$$

$$\frac{\Gamma, \phi\{x \mapsto y\}, \Delta \rightarrow \chi}{\Gamma, \exists x \phi, \Delta \rightarrow \chi}\ (\exists \rightarrow) \qquad \frac{\Gamma \rightarrow \phi\{x \mapsto t\}}{\Gamma \rightarrow \exists x \phi}\ (\rightarrow \exists)$$

Figure 3:

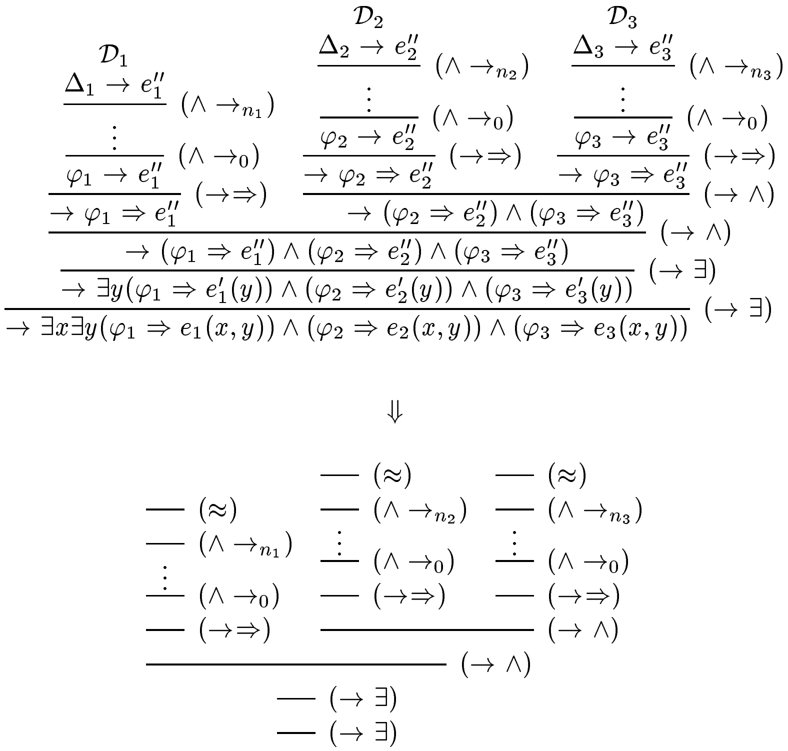

$\mathcal{D}_1$
$\Delta_1 \to e_1''$
$(\wedge \to_{n_1})$
$(\wedge \to_0)$
$\varphi_1 \to e_1''$
$(\to\Rightarrow)$
$\to \varphi_1 \Rightarrow e_1''$
$\mathcal{D}_2$
$\Delta_2 \to e_2''$
$(\wedge \to_{n_2})$
$(\wedge \to_0)$
$\varphi_2 \to e_2''$
$(\to\Rightarrow)$
$\to \varphi_2 \Rightarrow e_2''$
$\mathcal{D}_3$
$\Delta_3 \to e_3''$
$(\wedge \to_{n_3})$
$(\wedge \to_0)$
$\varphi_3 \to e_3''$
$(\to\Rightarrow)$
$\to \varphi_3 \Rightarrow e_3''$
$(\to \wedge)$
$\to (\varphi_2 \Rightarrow e_2'') \wedge (\varphi_3 \Rightarrow e_3'')$
$(\to \wedge)$
$\to (\varphi_1 \Rightarrow e_1'') \wedge (\varphi_2 \Rightarrow e_2'') \wedge (\varphi_3 \Rightarrow e_3'')$
$(\to \exists)$
$\to \exists y(\varphi_1 \Rightarrow e_1'(y)) \wedge (\varphi_2 \Rightarrow e_2'(y)) \wedge (\varphi_3 \Rightarrow e_3'(y))$
$(\to \exists)$
$\to \exists x \exists y(\varphi_1 \Rightarrow e_1(x,y)) \wedge (\varphi_2 \Rightarrow e_2(x,y)) \wedge (\varphi_3 \Rightarrow e_3(x,y))$
$\Downarrow$
$(\approx)$
$(\wedge \to_{n_1})$
$(\wedge \to_0)$
$(\to\Rightarrow)$
$(\approx)$
$(\wedge \to_{n_2})$
$(\wedge \to_0)$
$(\to\Rightarrow)$
$(\approx)$
$(\wedge \to_{n_3})$
$(\wedge \to_0)$
$(\to\Rightarrow)$
$(\to \wedge)$
$(\to \wedge)$
$(\to \exists)$
$(\to \exists)$

Figure 4: