UPMAIL Technical Report No. 152
February 16, 1998
ISSN 1100-0686

Simultaneous rigid E-unification and other decision problems related to the Herbrand theorem

Andrei Voronkov*

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

email: voronkov@csd.uu.se

*Supported by a TFR grant.

Abstract

Recently, a number of results have been published related to simultaneous rigid E-unification and Herbrand's theorem for logic with equality. The aim of this article is to overview these results, fill in some proofs that have only been sketched before, and present some new results.[2]

Contents

1 Introduction

Herbrand's theorem is one of the major tools in automated deduction in classical logic. The tableau method, the connection method and model elimination are based on a direct use of this theorem. Variants of Herbrand's theorem are used in completeness proofs for nearly all methods of automated deduction, including variants of resolution.

Recently, a number of results about Herbrand's theorem and proof-search for logic with equality have been published. This article aim at presenting results in this area and filling in proofs of some results that have been announced by not formally justified.

We also give more precise formulations of results with respect to the signatures in which problems are formulated and provide a table of known results.

2 Preliminaries

The equality predicate is denoted $\approx$. The symbol $\vdash$ denotes provability in classical logic. By $[x_1 \mapsto t_1, \dots, x_n \mapsto t_n]$ we denote the substitution that replaces variables x_i by terms t_i. If θ is a substitution and t is a term, then $t\theta$ denotes the term obtained from t by applying θ. An expression (e.g. a term or a set of terms) is called *ground* if it contains no variables.

We only consider first-order formulas. We assume that every signature considered in this article contains at least one constant. The *signature of a formula* φ is the signature consisting of all symbols occurring in φ, plus one fixed constant if φ contains no constants. If all variables of a formula φ are $\mathbf{x}$, we denote by $\exists\varphi$ the formula $\exists\mathbf{x}\varphi$.

Some of results depend on the signature and logic in which formulas are written. We consider two logics: with and without equality. If a signature Σ consists of k constants, l unary function symbols, and m function symbols of arity ≥ 2, we shall denote such a signature by (k, l, m). We shall also use ordinals and wildcards in the notation for signatures. For example, $(\omega, \_, 3)$ denotes any signature with infinitely many constants, any number of unary function symbols, and 3 function symbols of arity ≥ 2. Similarly, $(< \omega, 0, 0)$ denotes a signature with any finite number of constants and no function symbols of arity ≥ 1. Signatures without function symbols of arity ≥ 2, i.e., $(\_, \_, 0)$ are called *monadic*.

A *literal* is either an atomic formula or the negation of an atomic formula. Literals that are atomic formulas are called *positive*, and literals that are negations of atomic formulas *negative*. A formula is in *negation normal form* if it is constructed from literals using $\wedge, \vee, \forall, \exists$.

We use the terms "unsatisfiable" and "inconsistent" as synonyms.

A *sequent* is an expression $\Gamma \to \Delta$, where Γ and Δ are multisets of formulas. In this article, we shall use several sequent calculi. Let us fix some terminology related to derivations in such calculi.

An *inference rule* is an n-ary relation on the set of sequents, where $n \geq 1$. The elements of such a relation are called *inferences* and usually written as

$$\frac{S_1 \quad \ldots \quad S_{n-1}}{S} \ .$$

We call the sequents $S_1, \ldots, S_{n-1}$ *premises*, and the sequent S the *conclusion*, of this inference. A *calculus* is a set of inference rules. An *axiom* of a calculus is any conclusion of the rule with 0 premises. A *derivation* of a sequent S in a calculus is a tree of sequents with the root S formed by inferences in the calculus. A *proof* is any finite derivation whose leaves are axioms. We shall speak about derivations of formulas φ is sequent calculi, meaning derivations of the sequents $\to \varphi$.

3 Herbrand's theorem and related problems

3.1 Herbrand's theorem

Most automated reasoning methods for classical logic reduce, by means of skolemization, a given formula to a prenex form $\exists\mathbf{x}\varphi(\mathbf{x})$, where $\varphi(\mathbf{x})$ is quantifier free. For such formulas, provability is characterized by Herbrand's theorem.

Theorem 3.1 (Herbrand [22]) A sentence $\exists\mathbf{x}\varphi(\mathbf{x})$, where $\varphi(\mathbf{x})$ is quantifier-free, is provable if and only if so is a particular disjunction $\varphi(\mathbf{t}_1) \vee \ldots \vee \varphi(\mathbf{t}_n)$.

It is well-known that it suffices to search for terms $\mathbf{t}_1, \ldots, \mathbf{t}_n$ among ground terms of the signature of φ. Thus, we can reformulate Herbrand's theorem as follows.

Theorem 3.2 Let $\exists\mathbf{x}\varphi(\mathbf{x})$ be a sentence, where $\varphi(\mathbf{x})$ is quantifier-free. This sentence is provable if and only if so is a particular disjunction $\varphi(\mathbf{t}_1) \vee \ldots \vee \varphi(\mathbf{t}_n)$, where $\mathbf{t}_1, \ldots, \mathbf{t}_n$ are ground terms in the signature of φ.

Example 3.3 It is known that $n = 1$ may not suffice. For example, consider

$$\exists x((\neg P(a) \wedge \neg P(b)) \vee P(x)).$$

Then no formula of the form

$$(\neg P(a) \land \neg P(b)) \lor P(t).$$

is provable, but the disjunction

$$\begin{array}{l}(\neg P(a) \land \neg P(b)) \lor P(a) \lor \\ (\neg P(a) \land \neg P(b)) \lor P(b)\end{array}$$

is provable.

Herbrand's theorem holds both for logic with equality and logic without equality. It also holds for any first-order theory axiomatized by universally quantified sentences.

Herbrand's theorem is directly used in several methods of automated deduction by means of the following loop.

1. guess a number n;
2. find particular terms $\mathbf{t}_1, \ldots, \mathbf{t}_n$;
3. if such terms are not found, increase n and repeat step 2.

In the sequel we shall refer to methods using this loop as *tableau-like methods*.

Step 2 of this procedure gives rise to several decision problems.

3.2 Formula Instantiation

The first relevant problem is the following.

Decision problem 1 (Herbrand Skeleton [36]) Given a quantifier-free formula $\varphi(\mathbf{x})$ and a positive natural number n, do there exist term sequences $\mathbf{t}_1, \ldots, \mathbf{t}_n$ such that the formula $\varphi(\mathbf{t}_1) \lor \ldots \lor \varphi(\mathbf{t}_n)$ is provable?

We denote the Herbrand Skeleton problem for a signature Σ by $\mathsf{HS}(\Sigma)$, and by $\mathsf{HS}(\Sigma)^{\approx}$ for logic with equality.

When $n = 1$, we have a special case of this problem.

Decision problem 2 (Formula Instantiation) Given a quantifier-free formula $\varphi(\mathbf{x})$, does there exist a single term sequence $\mathbf{t}$ such that the formula $\varphi(\mathbf{t})$ is provable?

In view of Theorem 3.2, both for Herbrand Skeleton and Formula Instantiation, term sequences $\mathbf{t}_1, \ldots, \mathbf{t}_n$ and $\mathbf{t}$ may be searched among ground terms of the signature of φ.

We denote the Formula Instantiation problem for a signature Σ by $\mathsf{FI}(\Sigma)$, and by $\mathsf{FI}(\Sigma)^{\approx}$ for logic with equality.

Theorem 3.4 For every signature Σ, the problem $\mathsf{HS}(\Sigma)$ (respectively $\mathsf{HS}(\Sigma)^{\approx}$) is decidable if and only if so is $\mathsf{FI}(\Sigma)$ (respectively $\mathsf{FI}(\Sigma)^{\approx}$).

Proof. Formula Instantiation is a special case of Herbrand Skeleton, when $n = 1$. To reduce Herbrand Skeleton to Formula Instantiation, for every instance $(n, \varphi(\mathbf{x}))$ of Herbrand Skeleton, take the instance $\varphi(\mathbf{x}_1) \lor \ldots \lor \varphi(\mathbf{x}_n)$ of Formula Instantiation. □

The following theorem characterizes complexity of Formula Instantiation for logic without equality.

Theorem 3.5 For every signature with at least two symbols, $\mathsf{FI}(\Sigma)$ is Σ_2^p-complete.

Proof. First, we show that $\mathsf{FI}(\Sigma)$ is in Σ_2^p. Let $\varphi(\mathbf{x})$ be an instance of Formula Instantiation. Let us call a *split* of $\varphi(\mathbf{x})$ any equivalence relation on the set of atomic subformulas of $\varphi(\mathbf{x})$. We shall identify such equivalence relations with the sets of corresponding equivalence classes. Thus, any split can be considered as a set of sets of atomic subformulas of $\varphi(\mathbf{x})$.

Given a split S, we obtain from $\varphi(\mathbf{x})$ a propositional formula φ_S as follows. Replace all atomic subformulas by propositional variables so that subformulas belonging to the same equivalence class of S are replaced by the same symbol, and subformulas belonging to different equivalence classes are replaced by different subformulas.

A split is called a *justification* of $\varphi(\mathbf{x})$ if

1. φ_S is a tautology;
2. there exists a substitution θ such that for every equivalence class $E \in S$ and for every two formulas $A, B \in E$ we have $A\theta = B\theta$.

Consider an example, developed from Example 3.3. Suppose $\varphi(x_1, x_2)$ is the formula

$$(\neg P(a) \land \neg P(b)) \lor P(x_1) \lor \\ (\neg P(a) \land \neg P(b)) \lor P(x_2).$$

Consider the split S with two equivalence classes:

$$\{P(a), P(x_1)\} \quad \text{and} \quad \{P(b), P(x_2)\}.$$

This split is a justification. Indeed, by replacing $P(a), P(x_1)$ by the propositional letter A, and $P(b), P(x_2)$ by the propositional letter B, we obtain the tautology

$$\varphi_S = (\neg A \land \neg B) \lor A \lor (\neg A \land \neg B) \lor B.$$

The corresponding substitution θ is $[x_1 \mapsto a, x_2 \mapsto b]$.

Is not hard to argue that the instance $\varphi(\mathbf{x})$ of Formula Instantiation holds if and only if there exists a justification of $\varphi(\mathbf{x})$. Let us now prove that the problem of the existence of a justification is in Σ_2^p. This follows from the following observations:

1. all splits can be found by a nondeterministic polynomial-time algorithm;
2. the problem of checking, given a split S, whether φ_S is a tautology, is co-NP-complete;
3. the problem of finding the appropriate substitution θ is polynomial-time (it is the so-called *simultaneous unification problem* known to be P-complete [13, 44, 14]).

It remains to prove that Formula Instantiation is Σ_2^p-hard. We prove this for the signature with two constants $0, 1$. We use reduction of the QBF_2 problem known to be Σ_2^p-complete, see e.g. [30]. This problem can be formulated as follows. Consider the algebra $\mathbf{2}$ with two elements $0, 1$ in the signature with two constants $0, 1$ representing these elements. Given a sentence of this signature

$$\exists \mathbf{x} \forall \mathbf{y} \Phi(\mathbf{x}, \mathbf{y}), \tag{1}$$

whose atomic formulas are of the form $x_i = 0, x_i = 1, y_j = 0, y_j = 1$, is this formula true in $\mathbf{2}$?

Given an instance (1) of QBF_2, we construct the corresponding instance of Formula Instantiation as follows. The instance uses a unary predicate symbol P and propositional symbols $Q_1, \ldots, Q_m$. Denote by Φ' the formula obtained from Φ by replacing

- each occurrence of $x_i = 0$ by $\neg P(x_i)$;
- each occurrence of $x_i = 1$ by $P(x_i)$;
- each occurrence of $y_i = 0$ by $\neg Q$;
- each occurrence of $y_i = 1$ by Q.

(Think of $P(x)$ as saying $x = 1$, then $\neg P(x)$ says $x = 0$.) Consider the instance of Formula Instantiation

$$\neg P(0) \wedge P(1) \supset \Phi'. \tag{2}$$

We claim that (1) is a yes-instance if and only if so is (2).

It is not hard to argue that if (2) is satisfied by a substitution, then it is also satisfied by a substitution θ such that θ maps each variable to 0 or 1. Let θ be such a substitution.

Also, it is not hard to argue that θ satisfies (2) if and only if

$$\mathbf{2} \models \forall \mathbf{y} \Phi(\mathbf{x}\theta, \mathbf{y}).$$

This proves the equivalence of (1) and (2). This proof also works for any other signature with at least two symbols, by replacing constants 0 and 1 by any two different ground terms. □

It is easy to see from this proof that Formula Instantiation in the signature with one constant is co-NP-complete.

The discussion of Herbrand Skeleton and Formula Instantiation for logic with equality is postponed until section 4.

The Herbrand Skeleton problem has been defined in [36] as a series of problems: for any fixed n we have the Herbrand Skeleton (n) problem whose instances are (n, φ). [36] prove that for all n the problem Herbrand Skeleton (n) is undecidable for logic with equality. [19] show that there is a general method of reducing the Herbrand Skeleton (n) problems to simultaneous rigid E-unification, and prove stronger undecidability results. Some other formulations of Herbrand Skeleton, where instead of n we consider a function depending on n and φ, are discussed in [40, 42]. We do not discuss the complexity of Herbrand Skeleton, again for a discussion see [42].

3.3 Matrix Instantiation

In the method of matings [1] or the connection method [4] provability of formulas is defined in terms of *matrices*. We call these methods *matrix-based methods*. A matrix is defined by induction as follows.

1. Any literal is a matrix;

2. If $M_1, \ldots, M_n$ are matrices, then so are

$$\left[\begin{array}{ccc} M_1 & \ldots & M_n \end{array} \right] \tag{3}$$

and

$$\left[\begin{array}{c} M_1 \\ \vdots \\ M_n \end{array} \right] \tag{4}$$

A *vertical path* in a matrix is a conjunction of literals defined as follows.

1. A matrix consisting of one literal L has one vertical path L.
2. If p is a vertical path in a matrix M_i, where $1 \leq i \leq n$, then p is also a vertical path in (3).
3. If p_i are vertical paths in matrices M_i, where $1 \leq i \leq n$, then $p_1 \wedge \ldots \wedge p_n$ is a vertical path in (4).

Matrix-based methods try to instantiate matrices so that every vertical path becomes inconsistent. This gives rise the following decision problem.

Decision problem 3 (Matrix Instantiation) Given a matrix M, does there exist a substitution σ such that every vertical path in $M\sigma$ is inconsistent?

We denote Matrix Instantiation for a signature Σ by $\mathsf{MI}(\Sigma)$, and by $\mathsf{MI}(\Sigma)^{\approx}$ for logic with equality.

Consider, for example, the following matrix

$$\left[\begin{array}{cc} \left[\begin{array}{c} P(f(y)) \\ \neg P(x) \end{array} \right] & \left[\begin{array}{c} Q(y) \\ \neg Q(g(x)) \end{array} \right] \\ \multicolumn{2}{c}{\neg Q(f(z))} \end{array} \right] \tag{5}$$

It has two vertical paths:

$$P(f(y)) \wedge \neg P(x) \wedge \neg Q(f(z)) \qquad \text{and} \qquad Q(y) \wedge \neg Q(g(x)) \wedge \neg Q(f(z)).$$

After applying the substitution $[x \mapsto f(f(z)), y \mapsto f(z)]$, the paths become

$$P(f(f(z))) \wedge \neg P(f(f(z))) \wedge \neg Q(f(z)) \qquad \text{and} \qquad Q(f(z)) \wedge \neg Q(g(f(f(z)))) \wedge \neg Q(f(z)).$$

Now each vertical path is inconsistent.

Theorem 3.6 For every signature Σ, the problem $\mathsf{MI}(\Sigma)$ (respectively, $\mathsf{MI}(\Sigma)^{\approx}$) is polynomial-time equivalent to $\mathsf{FI}(\Sigma)$ (respectively, $\mathsf{FI}(\Sigma)^{\approx}$).

Proof. First, we prove that Formula Instantiation is polynomial-time reducible to Matrix Instantiation. Let ψ be an instance of Formula Instantiation. Construct in polynomial time a formula ψ' in negation normal form such that $\vdash \psi' \equiv \neg\psi$.

For every formula φ in negation normal form define the matrix M_φ by induction as follows. If φ is a literal, then M_φ is φ. If φ has the form $\varphi_1 \wedge \ldots \wedge \varphi_n$ (respectively, $\varphi_1 \vee \ldots \vee \varphi_n$), then M_φ is the matrix

$$\begin{bmatrix} M_{\varphi_1} \\ \vdots \\ M_{\varphi_n} \end{bmatrix} \qquad \text{(respectively, the matrix} \qquad \begin{bmatrix} M_{\varphi_1} & \ldots & M_{\varphi_n} \end{bmatrix} \quad \text{).}$$

We prove that ψ is a yes-instance of Formula Instantiation if and only if $M_{\psi'}$ is a yes-instance of Matrix Instantiation.

Let $p_1, \ldots, p_m$ be all vertical paths in $M_{\psi'}$. It is not hard to argue that

$$\vdash \psi' \equiv \bigvee_{1 \leq i \leq m} p_i. \tag{6}$$

Suppose ψ is a yes-instance of Formula Instantiation. Then there exists a substitution θ such that $\psi\theta$ is ground and provable. Hence, $\psi'\theta$ is unsatisfiable. In view of (6), $\bigvee_{1\leq i\leq m} p_i\theta$ is unsatisfiable, hence every $p_i\theta$ is inconsistent. Therefore, $M_{\psi'}$ is a yes-instance of Matrix Instantiation. In a similar way we can prove that ψ is a yes-instance of Formula Instantiation, whenever $M_{\psi'}$ is a yes-instance of Matrix Instantiation. Since $M_{\psi'}$ can be constructed in polynomial time, Formula Instantiation is polynomial-time reducible to Matrix Instantiation.

Then, we prove that Matrix Instantiation is polynomial-time reducible to Formula Instantiation. Given any matrix M, construct a quantifier-free formula φ_M as follows. If M is a literal L, then φ_M is L. If M is matrix (4), then φ_M is the conjunction of φ_{M_i}. If M is matrix (3), then φ_M is the disjunction of φ_{M_i}. Reasoning as above, it is not hard to argue that M is a yes-instance of Matrix Instantiation if and only if $\neg\varphi_M$ is a yes-instance of Formula Instantiation. It remains to note that $\neg\varphi_M$ can be constructed in polynomial time. □

3.4 Skeleton Instantiation

There exists another relevant problem: instantiation of a derivation skeleton to a valid derivation. One can define different notions of a derivation skeleton. In our case derivation skeletons are obtained from sequent-style derivations by removing all sequents (so that only names of inference rules remain), and omitting all applications of equality rules. Such skeletons and related decision problems are discussed in [38, 37] for logic without equality and in [39, 43] for intuitionistic logic with equality.

We refer the reader to one of the above mentioned papers for thorough discussion and the precise definition of skeletons. Here we shall only give examples illustrating the notion of a skeleton. We consider a standard sequent-style cut-free calculus with invertible rules[3].

[3]The results presented in this section remain true for all standard cut-free formalizations of sequent calculi, invertibility of rules is not essential.

Examples of inference rules in this calculus are given below.

$$\frac{\Gamma \to \Delta, \varphi, \psi}{\Gamma \to \Delta, \varphi \vee \psi}\;(\to \vee) \qquad \frac{\Gamma, \varphi \to \Delta \quad \Gamma, \psi \to \Delta}{\Gamma, \varphi \vee \psi \to \Delta}\;(\vee \to)$$

$$\frac{\Gamma \to \Delta, \exists x \varphi(x), \varphi(t)}{\Gamma \to \Delta, \exists x \varphi(x)}\;(\to \exists)$$

In the case of logic with equality, we also have three equality rules:

$$\frac{}{\Gamma \to \Delta, t \approx t}\;(\approx)$$

$$\frac{\Gamma(s), s \approx t \to \Delta(s)}{\Gamma(t), s \approx t \to \Delta(t)}\;(\approx_l) \qquad \frac{\Gamma(s), t \approx s \to \Delta(s)}{\Gamma(t), t \approx s \to \Delta(t)}\;(\approx_r)$$

The *skeleton* of a derivation is the figure obtained from the derivation by removing all sequents and all applications of rules $(\approx_l)$ and $(\approx_r)$. For example, consider the derivation

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{a \approx b, x \approx a \to a \approx a}\;(\approx)}{a \approx b, x \approx a \to x \approx a}\;(\approx_r)}{a \approx b \to x \approx a \supset x \approx a}\;(\to\supset)}{a \approx b \to x \approx a \supset x \approx b}\;(\approx_l)}{a \approx b \to \forall x (x \approx a \supset x \approx b)}\;(\to \forall)}{\to a \approx b \supset \forall x (x \approx a \supset x \approx b)}\;(\to\supset)$$

It has the following skeleton.

$$\cfrac{\cfrac{\cfrac{\cfrac{}{}\;(\approx)}{}\;(\to\supset)}{}\;(\to \forall)}{}\;(\to\supset)$$

Decision problem 4 (Skeleton Instantiation) Given a formula φ and a derivation skeleton $\mathcal{S}$, does there exist a derivation of φ with the skeleton $\mathcal{S}$?

We denote the Skeleton Instantiation problem for a signature Σ by $\mathsf{SI}(\Sigma)$, and by $\mathsf{SI}(\Sigma)^{\approx}$ for logic with equality.

For logic without equality, [38] proves that Skeleton Instantiation is NP-complete. However, the proof of NP-hardness in [38] essentially uses the presence of a binary symbol in the signature. We give a new proof here, which is valid for any signature.

Theorem 3.7 $\mathsf{SI}(\_,\_,\_)$ is NP-complete.

Proof. For a proof of containment in NP see [38]. The proof is not very complex, but requires the consideration of many small details. The main idea is that, given a skeleton, we can nondeterministically guess the principal formulas of inferences, and then to find a substitution making this skeleton a valid derivation. Checking the existence of a substitution can be done in polynomial time by simultaneous unification plus solving so-called nonoccurrence constraints.

We now prove NP-hardness. The proof will use a polynomial-time reduction of the propositional satisfiability problem known to be NP-complete and defined as follows.

We call a *clause* any disjunction of literals. Instances of the propositional satisfiability problem are conjunctions of clauses. An instance is a yes-instance if there exists a truth assignment that makes the conjunction true.

Let

$$\varphi = C_1 \wedge \ldots \wedge C_n$$

be an instance of the propositional satisfiability problem. Let $p_1, \ldots, p_m$ be all propositional variables of φ. Consider the signature consisting of two predicate symbols T and F. Define a first-order formula $\varphi'(x_1, \ldots, x_m)$ obtained from φ by replacing

- every negative literal $\neg p_i$ by $F(x_i)$;
- every positive literal p_i by $T(x_i)$.

Define the formula

$$\psi = \forall y \forall z (T(y) \wedge F(z) \supset \exists x_1 \ldots \exists x_m \varphi').$$

Consider also the skeleton $\mathcal{S}$ shown in Figure 1. We claim that the pair $(\psi, \mathcal{S})$ is a yes-instance of Skeleton Instantiation if and only if φ is a yes-instance of the propositional satisfiability problem.

It is not hard to argue that any derivation of $\rightarrow \varphi$ with this skeleton has the form shown in Figure 2, where σ is a substitution for variables $x_1, \ldots, x_m$ and $0, 1$ are constants. Evidently, we can restrict σ to be of the form $[x_1 \mapsto t_1, \ldots, x_m \mapsto t_m]$, where $\{t_1, \ldots, t_m\} \subseteq \{0, 1\}$.

Consider an arbitrary substitution σ of this form and define the truth value assignment τ for $p_1, \ldots, p_m$ by $\tau(p_i) = x_i\sigma$. We leave it to the reader to check that the derivation of Figure 2 is a valid proof if and only if τ satisfies each clause C_i. $\Box$

As for logic with equality, the following result is proved in [43].

Theorem 3.8 $\mathsf{SI}(\_, y, z)^{\approx}$ is decidable if and only if so is $\mathsf{FI}(\omega, y, z)^{\approx}$. $\Box$

In fact, this theorem is stated in [43] for skeletons of intuitionistic derivations, but the proof works literally for classical derivations as well.

3.5 Simultaneous rigid E-unification

In this section we introduce a decision problem that can be considered as a combinatorial analogue of the problems we have discussed. This decision problem was introduced in [17] as *simultaneous rigid E-unification*, or SREU for short.

Decision problem 5 (Simultaneous Rigid E-unification) Given finite sets of equations E_i and equations $s_i \approx t_i$, where $i \in \{1, \ldots, n\}$, does there exist a substitution θ such that $\vdash \bigwedge_{e \in E_i} e\theta \supset s_i\theta \approx t_i\theta$ for all i?

We denote the Simultaneous Rigid E-unification problem for a signature Σ by $\mathsf{SREU}(\Sigma)$. Note that we define $\mathsf{SREU}(\Sigma)$ only for logic with equality.

We introduce some terminology related to simultaneous rigid E-unification. We call a *rigid equation* any expression of the form $E \vdash_{\forall} s \approx t$, where E is a finite set of equations and s, t are terms. In such a rigid equation, E is called its *left-hand side* and the equation $s \approx t$ its *right-hand side*. A *system of rigid equations* is a finite set of rigid equations. A *solution* to a rigid equation

$$
\begin{array}{c}
\dfrac{\mathcal{S}_1 \qquad \begin{array}{c} \dfrac{\mathcal{S}_{n-1} \quad \mathcal{S}_n}{\qquad\qquad}\ (\to \wedge) \\ \vdots \end{array}}{\qquad\qquad\qquad}\ (\to \wedge) \\
\overline{\qquad}\ (\to \exists) \\
\vdots \\
\overline{\qquad}\ (\to \exists) \\
\overline{\qquad}\ (\wedge \to) \\
\overline{\qquad}\ (\to \supset) \\
\overline{\qquad}\ (\to \forall) \\
\overline{\qquad}\ (\to \forall)
\end{array}
$$

This skeleton has $n-1$ applications of $(\to \wedge)$ and m applications of $(\to \exists)$. Each skeleton $\mathcal{S}_i$ has the form

$$
\left.
\begin{array}{c}
\overline{\qquad}\ (\to \vee) \\
\vdots \\
\overline{\qquad}\ (\to \vee)
\end{array}
\right\}
\begin{array}{l} k-1 \text{ applica-} \\ \text{tions of } (\to \vee) \end{array}
$$

where k is the number of literals in the clause C_i.

Figure 1: A skeleton for Theorem 3.7

$$
\begin{array}{c}
\dfrac{\begin{array}{c}\Pi_{n-1} \\ T(1), F(0) \to D_{n-1}\sigma\end{array} \quad \begin{array}{c}\Pi_{n} \\ T(1), F(0) \to D_{n}\sigma\end{array}}{T(1), F(0) \to (D_{n-1} \wedge D_n)\sigma}\ (\to \wedge) \\
\vdots \\
\dfrac{\dfrac{\begin{array}{c}\Pi_1 \\ T(1), F(0) \to D_1\sigma\end{array} \qquad T(1), F(0) \to (D_2 \wedge \ldots \wedge D_n)\sigma}{T(1), F(0) \to (D_1 \wedge \ldots \wedge D_n)\sigma}\ (\to \wedge)}{T(1), F(0) \to (\exists x_m (D_1 \wedge \ldots \wedge D_n))\sigma}\ (\to \exists) \\
\vdots \\
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{T(1), F(0) \to (\exists x_2 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n))\sigma}{T(1), F(0) \to \exists x_1 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n)}\ (\to \exists)}{T(1) \wedge F(0) \to \exists x_1 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n)}\ (\wedge \to)}{\to T(1) \wedge F(0) \supset \exists x_1 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n)}\ (\to \supset)}{\to \forall z (T(1) \wedge F(z) \supset \exists x_1 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n))}\ (\to \forall)}{\to \forall y \forall z (T(y) \wedge F(z) \supset \exists x_1 \ldots \exists x_m (D_1 \wedge \ldots \wedge D_n))}\ (\to \forall)
\end{array}
$$

Here each derivation Π_i has the form

$$
\begin{array}{c}
\dfrac{T(1), F(0) \to A_1\sigma, \ldots, A_k\sigma}{T(1), F(0) \to A_1\sigma, \ldots, A_{k-2}\sigma, (A_{k-1} \vee A_k)\sigma}\ (\to \vee) \\
\vdots \\
\dfrac{T(1), F(0) \to A_1\sigma, (A_2 \vee \ldots \vee A_k)\sigma}{T(1), F(0) \to (A_1 \vee \ldots \vee A_k)\sigma}\ (\to \vee)
\end{array}
$$

where k is the number of literals in the clause C_i.

Figure 2: A derivation for Theorem 3.7

$E \vdash_\forall s \approx t$ is any substitution θ such that $\vdash \bigwedge_{e \in E_i} e\theta \supset s\theta \approx t\theta$. A *solution to a system of rigid equations* is any substitution that solves all rigid equations in the system.

Consider an example. In this example we denote applications of unary function symbols to arguments without parentheses, for example we write fx instead of $f(x)$. Consider the following system of two rigid equations:

$$\begin{array}{c} hx_1 \approx a, hx_2 \approx a, ha \approx y \;\vdash_\forall\; y \approx ghy; \\ hx_1 \approx a, hx_2 \approx a, hb \approx fy \;\vdash_\forall\; y \approx gfy. \end{array}$$

This system has a solution $[x_1 \mapsto a, x_2 \mapsto b, y \mapsto ga]$, because

$$\begin{array}{c} \vdash ha \approx a, hb \approx a, ha \approx ga \;\supset\; ga \approx ghga; \\ \vdash ha \approx a, hb \approx a, hb \approx fga \;\supset\; ga \approx gfga. \end{array}$$

We can equivalently reformulate simultaneous rigid E-unification as the problem of checking solvability of systems of rigid equations.

We shall use one technical property of first-order logic.

Lemma 3.9 Let $r_1 \approx u_1, \ldots, r_l \approx u_l$ be equations, $s_1 \not\approx t_1, \ldots, s_p \not\approx t_p$ be disequations and $A_1, \ldots, A_m$ and $B_1, \ldots, B_n$ be atomic formulas different from equations. Then the disjunction

$$r_1 \approx u_1 \vee \ldots \vee r_l \approx u_l \vee s_1 \not\approx t_1 \vee \ldots \vee s_p \not\approx t_p \vee A_1 \vee \ldots \vee A_m \vee \neg B_1 \vee \ldots \vee \neg B_n \qquad (7)$$

is provable if and only if one of the following conditions hold:

1. there exists $i \in \{1, \ldots, l\}$ such that

$$\vdash s_1 \approx t_1 \wedge \ldots \wedge s_p \approx t_p \supset r_i \approx u_i;$$

2. there exist $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$ such that A_i has the form $P(v_1, \ldots, v_q)$, B_j has the form $P(w_1, \ldots, w_q)$ for the same predicate symbol P, and

$$\vdash s_1 \approx t_1 \wedge \ldots \wedge s_p \approx t_p \supset v_1 \approx w_1 \wedge \ldots \wedge v_q \approx w_q.$$

□

Let φ be a disjunction of literals (7). Define the set R_φ of systems of rigid equations as the smallest set with the following properties:

1. For every $i \in \{1, \ldots, l\}$ the set R_φ contains the system consisting of one equation

$$s_1 \approx t_1 \wedge \ldots \wedge s_p \approx t_p \vdash_\forall r_i \approx u_i.$$

2. For every $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$ such that A_i has the form $P(v_1, \ldots, v_q)$ and B_j has the form $P(w_1, \ldots, w_q)$ for the same predicate symbol P, the set R_φ contains the system

$$\begin{array}{c} s_1 \approx t_1 \wedge \ldots \wedge s_p \approx t_p \ \vdash_\forall \ v_1 \approx w_1; \\ \ldots \\ s_1 \approx t_1 \wedge \ldots \wedge s_p \approx t_p \ \vdash_\forall \ v_q \approx w_q. \end{array}$$

Lemma 3.9 immediately yields

Lemma 3.10 Let φ be a disjunction of literals. Then φ is a yes-instance of Formula Instantiation if and only if some system in R_φ is a yes-instance of Simultaneous Rigid E-unification. □

Theorem 3.11 Let Σ be a signature. $\mathsf{FI}(\Sigma)^\approx$ is decidable if and only if so is $\mathsf{SREU}(\Sigma)$.

Proof. First, we show an obvious reduction of $\mathsf{SREU}(\Sigma)$ to $\mathsf{FI}(\Sigma)^\approx$. Let $\{E_i \vdash_\forall s_1 \approx t_i\}_{i=1\ldots n}$ be an instance of $\mathsf{SREU}(\Sigma)$. It is easy to see that this instance has a solution if and only if the following formula is a yes-instance of $\mathsf{FI}(\Sigma)^\approx$:

$$\bigwedge_{i=1}^{n} (\bigwedge_{e \in E_i} e \supset s_i \approx t_i).$$

Next, we show how to reduce $\mathsf{FI}(\Sigma)^\approx$ to $\mathsf{SREU}(\Sigma)$. Let φ be a quantifier-free formula. Without loss of generality we assume that φ is in conjunctive normal form:

$$\varphi = \bigwedge_{i=1}^{k} \varphi_i,$$

where φ_i are disjunctions of literals. Constructs sets of systems of rigid equations $R_i = R_{\varphi_i}$ and define a set R of systems of rigid equations as follows: a system S belongs to R if and only if there exists $S_1 \in R_1, \ldots, S_k \in R_k$ such that $S = S_1 \cup \ldots \cup S_k$. Using Lemma 3.10, it is not hard to argue that φ is a yes-instance of Formula Instantiation if and only if some $S \in R$ is a yes-instance of Simultaneous Rigid E-unification. □

Since Simultaneous Rigid E-unification is equivalent (with respect to decidability) to all other problems we study in this paper, we shall overview results about Simultaneous Rigid E-unification.

A lemma on equational logic. Let us now formulate a lemma on equational logic that appears to be very useful in the study of Simultaneous Rigid E-unification.

Let Σ be a signature and c be a constant in Σ. Denote by $sig_\Sigma(x)$ the formula

$$\bigwedge_{f \in \Sigma} f(c, \ldots, c) \approx c \supset x \approx c.$$

Lemma 3.12 Let Σ be a signature, c be a constant in Σ and t be a term. Then the formula $sig_\Sigma(t)$ is provable in $\mathbf{LJ}^\approx$ if and only if t is a ground term in Σ. □

Essentially, this lemma says that using rigid equations we can express the property *"t is a ground term in a given signature* Σ*"*. For every signature Σ and term t, denote by $R_\Sigma(t)$ the following rigid equation:

$$\{f(c, \dots , c) \approx c \mid f \in \Sigma\} \vdash_\forall t \approx c,$$

where c is any constant in Σ. Lemma 3.12 immediately implies the following.

Lemma 3.13 A substitution θ is a solution to $R_\Sigma(t)$ if and only if $t\theta$ is a ground term in Σ. $\square$

3.6 Encodings of signatures

In this section we give encodings of some signatures in other signatures. These encodings will reduce signature-depending instances of simultaneous rigid E-unification to a finite number of signatures.

Binary signatures. We show how to encode terms in an arbitrary finite signature by terms in the signature $(1, 0, 1)$. Without loss of generality we assume that the signature $(1, 0, 1)$ consists of a constant a and a binary function symbol f.

First, we define an encoding of natural numbers. For every natural number n define the term $\overline{n}$ in $(1, 0, 1)$ as follows:

$$\begin{array}{rcl} \overline{0} & = & f(f(a, a), a); \\ \overline{n+1} & = & f(\overline{n}, a). \end{array}$$

Now, let Σ be an arbitrary signature with function symbols (including constants) $g_1, \dots , g_n$. For every term t in Σ (respectively, tuple of terms $t_1, \dots , t_n$), their *binary encodings* in $(0, 1, 0)$, denoted $B(t)$ (respectively, $\langle t \rangle$) are defined by the following recursive definitions:

$$\begin{array}{rcl} B(x) & = & x, \text{ where } x \text{ is a variable;} \\ \langle\rangle & = & a; \\ \langle t_1, \dots , t_m\rangle & = & f(t_1, \langle t_2, \dots , t_m\rangle); \\ B(g_i(t_1, \dots , t_m)) & = & f(a, f(\overline{m}, \langle t_1, \dots , t_m\rangle)) \end{array}$$

The main property of $B(t)$ is the following.

Lemma 3.14 Let s, t be terms.

1. s is subterm of t if and only if $B(s)$ is a subterm of $B(t)$.

2. Let a term t' be obtained by replacing an occurrence of s in t by a term r. Then $B(t')$ can be obtained by replacing an occurrence of $B(s)$ in $B(t)$ by the term $B(r)$.

3. Let the term t'' be obtained by replacing an occurrence of $B(s)$ in $B(t)$ by a term $B(r)$. Then there exists a term t' such that $t'' = B(t')$ and t' is obtained by replacing an occurrence of s in t by r. $\square$

Using this lemma, it is not hard to prove an analogue of Lemma 3.13. Let c be a constant in and g be a function symbol, both in the signature Σ. Denote $R'_\Sigma t$ the following rigid equation in the signature $(1, 0, 1)$:

$$\{B(f(c, \dots, c)) \approx B(c) \mid f \in \Sigma\} \vdash_\forall B(t) \approx B(c),$$

Lemma 3.15 A substitution θ is a solution to $R'_\Sigma(t)$ if and only if $t\theta = B(s)$ for some ground term s in Σ. □

Using the binary encoding, we can reduce Simultaneous Rigid E-unification to Simultaneous Rigid E-unification in the signature with one constant and one binary function symbol.

Lemma 3.16 Let Σ be an arbitrary signature. Then $\mathsf{SREU}(\Sigma)$ is polynomial-time reducible to $\mathsf{SREU}(1, 0, 1)$.

Proof. (Sketch.) The proof will use the binary encoding introduced above. Let $\mathbf{R} = (R_1, \dots, R_m)$ be any system of rigid equations in Σ. Let $x_1, \dots, x_n$ be all variables of $\mathbf{R}$ and Σ' be the smallest subsignature of Σ containing all symbols occurring in $\mathbf{R}$. Consider the system $\mathbf{R}'$ of rigid equations obtained from $\mathbf{R}$ be replacement of all equations $s \approx t$ by $B(s) \approx B(t)$ and by adding the rigid equations $R'_{\Sigma'}(x_i)_{i=1,\dots,n}$. It is not hard to argue that $\mathbf{R}$ is solvable if and only if so is $\mathbf{R}'$. □

Since $(1, 0, 1)$ can be regarded as a subsignature of any signature of the form $(\_, \_, \geq 1)$, we also have that $\mathsf{SREU}(1, 0, 1)$ is polynomial-time reducible to $\mathsf{SREU}(\_, \_, \geq 1)$. Therefore, we have the following theorem.

Theorem 3.17 $\mathsf{SREU}(\_, \_, \geq 1)$ is polynomial-time equivalent to $\mathsf{SREU}(1, 0, 1)$. □

Monadic signatures. We define the *unary encoding* $U(t)$ of each term t in an arbitrary monadic signature with constants $a_1, a_2, \dots$ and unary function symbols $f_1, f_2, \dots$, as a term in the signature $(1, 2, 0)$ consisting of one constant a and two unary function symbols g, h as follows

$$\begin{aligned} U(a_i) &= g^i(a), \\ U(f_i(t)) &= g^i(h(\overline{t})). \end{aligned}$$

By using $U(t)$ instead of $B(t)$ and monadic signatures instead of arbitrary signatures, we can prove the analogue of Theorem 3.17:

Theorem 3.18 $\mathsf{SREU}(\_, \_, 0)$ is polynomial-time reducible to $\mathsf{SREU}(1, 2, 0)$. $\mathsf{SREU}(\_, \geq 2, 0)$ is polynomial-time equivalent to $\mathsf{SREU}(1, 2, 0)$.

One can also use the encodings $B(t)$ and $U(t)$ to prove statements similar to Theorems 3.17 and 3.18. For example, Theorem 3.11 implies that $\mathsf{FI}(\_, \geq 2, 0)$ is polynomial-time equivalent to $\mathsf{FI}(1, 2, 0)$.

3.7 The prenex fragment of intuitionistic logic

The problems mentioned above have relation to the prenex fragment of intuitionistic logic. A formula is called *prenex* if it has the form $Q_1x_1 \dots Q_nx_n\varphi$, where $Q_1, \dots, Q_n$ are quantifiers and φ is a quantifier-free formula. Such a formula is called *existencial* if, in addition, all quantifiers $Q_1, \dots, Q_n$ are $\exists$. In classical logic, there is an effective procedure (so-called *prenexing*) transforming each formula to an equivalent prenex formula. For this reason, the decision problem for prenex formulas of classical logic has been intensively studied and the complete classification of prenex fragments in terms of the signature, quantifier prefix and the presence of equality is known [5]. In intuitionistic logic it is not the case, so the value of prenex formulas is not evident.

Formally, we study the following problem.

Decision problem 6 (Prenex Intuitionistic) Is a given prenex sentence provable in intuitionistic logic?

We denote the Prenex Intuitionistic problem for a signature Σ by $\mathsf{PI}(\Sigma)$, and by $\mathsf{PI}(\Sigma)^{\approx}$ for logic with equality.

We also introduce a special case of Prenex Intuitionistic:

Decision problem 7 (Existencial Intuitionistic) Is a given existencial sentence provable in intuitionistic logic?

We denote the Existencial Intuitionistic problem for a signature Σ by $\mathsf{EI}(\Sigma)$, and by $\mathsf{EI}(\Sigma)^{\approx}$ for logic with equality.

In classical logic, the prenex fragment (i.e. the set of all provable prenex formulas) is undecidable. For intuitionistic logic without equality we have the following result proved in [38, 37].

Theorem 3.19 $\mathsf{PI}(\Sigma)$ and $\mathsf{EI}(\Sigma)$ are PSPACE-complete for every signature Σ. $\square$

Since intuitionistic propositional logic is already PSPACE-complete, this results also holds for subsets of $\mathsf{PI}(\Sigma)$ characterized by a restricted quantifier prefix.

For intuitionistic logic with equality, some results were sketched in [11], but without complete proofs. Here we give proofs of these results.

Our first aim is to introduce a cut-free sequent calculus $\mathbf{LJ}_2^{\approx}$ for intuitionistic logic in which all propositional derivations have a finite (and polynomial-size) depth. To this end, we first define the "usual" calculus $\mathbf{LJ}^{\approx}$ for intuitionistic logic with equality, and then one additional calculus $\mathbf{LJ}_1^{\approx}$.

The *calculus* $\mathbf{LJ}^{\approx}$ is given in Figure 3. The *calculus* $\mathbf{LJ}_1^{\approx}$ is given in Figure 4. The latter calculus (at least without equality rules) is a standard multi-succedent version of intuitionistic sequent calculus.

Though $\mathbf{LJ}^{\approx}$ is introduced as a calculus without cut rules, admissibility of cut can be proved by standard syntactic proofs of cut-eliminations, for example as in [28, 27].

We call *inessential* the rules $(\approx_l)$, $(\approx_r)$, $(\approx)$, (Ax) and $(\bot)$, and *essential* all other rules. We call an *equality derivation* any derivation consisting of only applications of inessential rules and an *essential derivation* any derivation consisting of only applications of essential rules.

We call a derivation in any of these calculi *regular* if it satisfies the following restrictions: above any application of $(\approx_l)$ and $(\approx_r)$ there may only be applications of inessential rules. $\mathbf{LJ}^{\approx}$ is a standard axiomatization for intuitionistic logic with equality. Thus, we can identify provability of a formula φ in intuitionistic logic with equality with provability of $\rightarrow \varphi$ in $\mathbf{LJ}^{\approx}$.

Lemma 3.20 Every provable formula of $\mathbf{LJ}^{\approx}$ has a regular proof. $\square$

$$\overline{\Gamma, A \to A}\;(Ax) \qquad \overline{\Gamma, \bot \to \Delta}\;(\bot) \qquad \overline{\Gamma \to t \approx t}\;(\approx)$$

$$\frac{\Gamma(s), s \approx t \to \Delta(s)}{\Gamma(t), s \approx t \to \Delta(t)}\;(\approx_2) \qquad \frac{\Gamma(t), s \approx t \to \Delta(t)}{\Gamma(s), s \approx t \to \Delta(s)}\;(\approx_r)$$

$$\frac{\Gamma, \varphi, \psi \to \Delta}{\Gamma, \varphi \wedge \psi \to \Delta}\;(\wedge \to) \qquad \frac{\Gamma \to \varphi \quad \Gamma \to \psi}{\Gamma \to \varphi \wedge \psi}\;(\to \wedge)$$

$$\frac{\Gamma, \varphi \to \Delta \quad \Gamma, \psi \to \Delta}{\Gamma, \varphi \vee \psi \to \Delta}\;(\vee \to) \qquad \frac{\Gamma \to \varphi, \psi}{\Gamma \to \varphi \vee \psi}\;(\to \vee)$$

$$\frac{\Gamma, \psi \to \Delta \quad \Gamma, \varphi \supset \psi \to \Delta, \varphi}{\Gamma, \varphi \supset \psi \to \Delta}\;(\supset\to) \qquad \frac{\Gamma, \varphi \to \psi}{\Gamma \to \varphi \supset \psi}\;(\to\supset)$$

$$\frac{\Gamma, \forall x\varphi(x), \varphi(t) \to \Delta}{\Gamma, \forall x\varphi(x) \to \Delta}\;(\forall \to) \qquad \frac{\Gamma \to \varphi(c)}{\Gamma \to \forall x\varphi(x)}\;(\to \forall)$$

$$\frac{\Gamma, \varphi(c) \to \Delta}{\Gamma, \exists x\varphi(x) \to \Delta}\;(\exists \to) \qquad \frac{\Gamma \to \varphi(t)}{\Gamma \to \exists x\varphi(x)}\;(\to \exists)$$

In this figure, A ranges over atomic formulas, φ, ψ, χ range over arbitrary formulas, Γ, Δ range over finite multisets of formulas, and Δ consists of at most one formula. In the rules $(\to \forall)$ and $(\exists \to)$ the constant c has no occurrences in the conclusions of the rules.

Figure 3: Calculus $\mathbf{LJ}^{\approx}$

$$\overline{\Gamma, A \to \Delta, A}\;(Ax) \qquad \overline{\Gamma, \bot \to \Delta}\;(\bot) \qquad \overline{\Gamma \to \Delta, t \approx t}\;(\approx)$$

$$\frac{\Gamma(s), s \approx t \to \Delta(s)}{\Gamma(t), s \approx t \to \Delta(t)}\;(\approx_2) \qquad \frac{\Gamma(t), s \approx t \to \Delta(t)}{\Gamma(s), s \approx t \to \Delta(s)}\;(\approx_r)$$

$$\frac{\Gamma, \varphi \wedge \psi, \varphi, \psi \to \Delta}{\Gamma, \varphi \wedge \psi \to \Delta}\;(\wedge \to) \qquad \frac{\Gamma \to \Delta, \varphi \wedge \psi, \varphi \quad \Gamma \to \Delta, \varphi \wedge \psi, \psi}{\Gamma \to \Delta, \varphi \wedge \psi}\;(\to \wedge)$$

$$\frac{\Gamma, \varphi \vee \psi, \varphi \to \Delta \quad \Gamma, \varphi \vee \psi, \psi \to \Delta}{\Gamma, \varphi \vee \psi \to \Delta}\;(\vee \to) \qquad \frac{\Gamma \to \Delta, \varphi \vee \psi, \varphi, \psi}{\Gamma \to \Delta, \varphi \vee \psi}\;(\to \vee)$$

$$\frac{\Gamma, \varphi \supset \psi, \psi \to \Delta \quad \Gamma, \varphi \supset \psi \to \Delta, \varphi}{\Gamma, \varphi \supset \psi \to \Delta}\;(\supset\to) \qquad \frac{\Gamma, \varphi \to \psi}{\Gamma \to \Delta, \varphi \supset \psi}\;(\to\supset)$$

$$\frac{\Gamma, \forall x\varphi(x), \varphi(t) \to \Delta}{\Gamma, \forall x\varphi(x) \to \Delta}\;(\forall \to) \qquad \frac{\Gamma \to \varphi(c)}{\Gamma \to \Delta, \forall x\varphi(x)}\;(\to \forall)$$

$$\frac{\Gamma, \varphi(c) \to \Delta}{\Gamma, \exists x\varphi(x) \to \Delta}\;(\exists \to) \qquad \frac{\Gamma \to \Delta, \varphi(t)}{\Gamma \to \Delta, \exists x\varphi(x)}\;(\to \exists)$$

In this figure, A ranges over atomic formulas, φ, ψ, χ range over arbitrary formulas, and Γ, Δ range over finite multisets of formulas. In the rules $(\to \forall)$ and $(\exists \to)$ the constant c has no occurrences in the conclusions of the rules.

Figure 4: Calculus $\mathbf{LJ}_1^{\approx}$

The proof is based on "down-permutability" of any inference with inferences by $(\approx_l)$ and $(\approx_r)$ and may be found in [29] or [43].

For finite multisets Γ, Γ' we write $\Gamma \dot{\subseteq} \Gamma'$ to denote that Γ' can be obtained from Γ by adding zero or more elements. Let $S_1 = \Gamma_1 \rightarrow \Delta_1$ and $S_2 = \Gamma_2 \rightarrow \Delta_2$ be sequents. We say that S_2 is an *extension* of S_1, denoted $S_1 \dot{\subseteq} S_2$, if $\Gamma_1 \dot{\subseteq} \Delta_1$ and $\Gamma_2 \dot{\subseteq} \Delta_2$.

The following lemma asserts that proofs in $\mathbf{LJ}^{\approx}$ may be mirrored by proofs in $\mathbf{LJ}_1^{\approx}$.

Lemma 3.21 For every inference Π with the conclusion S in $\mathbf{LJ}^{\approx}$ and every extension S' of S there exists an inference with the conclusion S' in $\mathbf{LJ}_1^{\approx}$ obtained from Π by replacing all occurrences of sequents by their extensions.

Proof. We consider only two cases, other cases are similar. Suppose that the inference in $\mathbf{LJ}^{\approx}$ has one of the following forms.

$$\frac{\Gamma, \psi \rightarrow \Delta \quad \Gamma, \varphi \supset \psi \rightarrow \varphi}{\Gamma, \varphi \supset \psi \rightarrow \Delta}\ (\supset\rightarrow) \qquad\qquad \frac{\Gamma, \varphi \rightarrow \psi}{\Gamma \rightarrow \varphi \supset \psi}\ (\rightarrow\supset)$$

Consider the following inferences in $\mathbf{LJ}_1^{\approx}$:

$$\frac{\Gamma', \varphi \supset \psi, \psi \rightarrow \Delta' \quad \Gamma', \varphi \supset \psi \rightarrow \Delta', \varphi}{\Gamma', \varphi \supset \psi \rightarrow \Delta'}\ (\supset\rightarrow) \qquad\qquad \frac{\Gamma', \varphi \rightarrow \psi}{\Gamma' \rightarrow \Delta', \varphi \supset \psi}\ (\rightarrow\supset)$$

where Γ' and Δ' are selected to satisfy the conditions of the lemma. It is clear that these inferences in $\mathbf{LJ}_1^{\approx}$ satisfy the statement. $\square$

Applying Lemma 3.21 inductively, we obtain

Lemma 3.22 For every derivation Π of a sequent S in $\mathbf{LJ}^{\approx}$ and any extension S' of S there exists a derivation of S' in $\mathbf{LJ}_1^{\approx}$ obtained from Π by replacing all occurrences of sequents by their extensions. $\square$

Now we note that $\mathbf{LJ}_1^{\approx}$ is sound.

Lemma 3.23 If a formula φ is provable in $\mathbf{LJ}_1^{\approx}$, then it is also provable in $\mathbf{LJ}^{\approx}$.

Proof. It is not hard to check that for every sequent $\Gamma \rightarrow \psi_1, \ldots, \psi_n$ provable in $\mathbf{LJ}_1^{\approx}$, the sequent $\Gamma \rightarrow \psi_1 \vee \ldots \vee \psi_n$ is provable in $\mathbf{LJ}^{\approx}$. This implies the statement of the lemma. $\square$

Lemma 3.24 A sequent S is provable in $\mathbf{LJ}^{\approx}$ if and only if it has a regular proof in $\mathbf{LJ}_1^{\approx}$.

Proof. Suppose S is provable in $\mathbf{LJ}^{\approx}$. By Lemma 3.20, there exists a regular proof Π of S in $\mathbf{LJ}^{\approx}$. Using Lemma 3.22, we can construct a proof Π' of S in $\mathbf{LJ}_1^{\approx}$ obtained by replacing in Π some sequents by their extensions. Evidently, Π' is regular.

Conversely, suppose S has a regular proof in $\mathbf{LJ}_1^{\approx}$. By Lemma 3.23, S is provable in $\mathbf{LJ}^{\approx}$. $\square$

The *calculus* $\mathbf{LJ}_2^{\approx}$ is given in Figure 5. Similar calculi are studied in [15, 23].

Using the same proof as in Lemma 3.23, we can establish the soundness of $\mathbf{LJ}_2^{\approx}$:

Lemma 3.25 If a formula φ is provable in $\mathbf{LJ}_2^{\approx}$, then it is also provable in $\mathbf{LJ}^{\approx}$. $\square$

$$\frac{}{\Gamma, A \to \Delta, A}\ (Ax) \qquad \frac{}{\Gamma, \bot \to \Delta}\ (\bot) \qquad \frac{}{\Gamma \to \Delta, t \approx t}\ (\approx)$$

$$\frac{\Gamma(s), s \approx t \to \Delta(s)}{\Gamma(t), s \approx t \to \Delta(t)}\ (\approx_2) \qquad \frac{\Gamma(t), s \approx t \to \Delta(t)}{\Gamma(s), s \approx t \to \Delta(s)}\ (\approx_r)$$

$$\frac{\Gamma, \varphi \wedge \psi, \varphi, \psi \to \Delta}{\Gamma, \varphi \wedge \psi \to \Delta}\ (\wedge \to) \qquad \frac{\Gamma \to \Delta, \varphi \wedge \psi, \varphi \quad \Gamma \to \Delta, \varphi \wedge \psi, \psi}{\Gamma \to \Delta, \varphi \wedge \psi}\ (\to \wedge)$$

$$\frac{\Gamma, \varphi \vee \psi, \varphi \to \Delta \quad \Gamma, \varphi \vee \psi, \psi \to \Delta}{\Gamma, \varphi \vee \psi \to \Delta}\ (\vee \to) \qquad \frac{\Gamma \to \Delta, \varphi \vee \psi, \varphi, \psi}{\Gamma \to \Delta, \varphi \vee \psi}\ (\to \vee)$$

$$\frac{\Gamma, \varphi \supset \psi, \psi \to \Delta \quad \Gamma, \varphi \supset \psi \to \Delta, \varphi}{\Gamma, \varphi \supset \psi \to \Delta}\ (\supset\to)$$

$$\frac{\Gamma, \varphi \to \psi}{\Gamma \to \Delta, \varphi \supset \psi}\ (\to\supset_1) \qquad \frac{\Gamma, \varphi \to \Delta, \varphi \supset \psi, \psi}{\Gamma, \varphi \to \Delta, \varphi \supset \psi}\ (\to\supset_2)$$

$$\frac{\Gamma, \forall x \varphi(x), \varphi(t) \to \Delta}{\Gamma, \forall x \varphi(x) \to \Delta}\ (\forall \to) \qquad \frac{\Gamma \to \varphi(c)}{\Gamma \to \Delta, \forall x \varphi(x)}\ (\to \forall)$$

$$\frac{\Gamma, \varphi(c) \to \Delta}{\Gamma, \exists x \varphi(x) \to \Delta}\ (\exists \to) \qquad \frac{\Gamma \to \Delta, \varphi(t)}{\Gamma \to \Delta, \exists x \varphi(x)}\ (\to \exists)$$

In this figure, all sequents satisfy the same conditions as for $\mathbf{LJ}_1^{\approx}$. In addition, the rules satisfy the following restrictions (we write $\varphi \notin \Gamma$ to denote that φ is not an element of the multiset Γ).

$(\wedge \to)$: either $\varphi \notin \Gamma$ or $\psi \notin \Gamma$.

$(\to \wedge)$: $\varphi \notin \Delta$ and $\psi \notin \Delta$.

$(\vee \to)$: $\varphi \notin \Gamma$ and $\psi \notin \Gamma$.

$(\to \vee)$: $\varphi \notin \Delta$ or $\psi \notin \Delta$.

$(\supset\to)$: $\psi \notin \Gamma$ and $\varphi \notin \Delta$.

$(\to\supset_1)$: $\varphi \notin \Gamma$.

$(\to\supset_2)$: $\psi \notin \Delta$.

Figure 5: Calculus $\mathbf{LJ}_2^{\approx}$

We are going to establish the completeness of $\mathbf{LJ}_2^{\approx}$ with respect to regular derivations. To this end, we prove a technical lemma. For finite multisets Γ_1, Γ_2, we write $\Gamma_1 \subseteq \Gamma_2$ to denote that every element of Γ_1 is an element of Γ_2. For sequents $S_1 = (\Gamma_1 \to \Delta_1)$ and $S_2 = (\Gamma_2 \to \Delta_2)$ we write $S_1 \subseteq S_2$ to denote that $\Gamma_1 \subseteq \Gamma_2$ and $\Delta_1 \subseteq \Delta_2$. Note that the relation $\subseteq$ is different from the previously introduced $\dot{\subseteq}$. For example, we have $A, A \subseteq A$ but not $A, A \dot{\subseteq} A$.

Lemma 3.26 Let Π be an essential inference in $\mathbf{LJ}_1^{\approx}$:

$$\frac{S_1 \quad \ldots \quad S_n}{S}$$

and $S \subseteq S'$. Then there exist sequents $S'_1, \ldots, S'_m$ with the following properties:

1. there exists an essential derivation Π' of S' in $\mathbf{LJ}_2^{\approx}$ from $S'_1, \ldots, S'_m$;
2. For every S'_i there exists j such that $S_j \subseteq S'_i$.

Proof. We only consider three cases, other cases are analogous.

1. The inference Π is by the rule $(\wedge \to)$:

$$\frac{\Gamma, \varphi, \psi \to \Delta}{\Gamma, \varphi \wedge \psi \to \Delta} \ (\wedge \to).$$

   Then S' has the form $\Gamma', \varphi \wedge \psi \to \Delta'$. Consider two cases.

   (a) $\varphi \in \Gamma'$ *and* $\psi \in \Gamma'$.
   In this case we have $(\Gamma, \varphi, \psi \to \Delta) \subseteq S'$. We can take Π' to be the 0-step derivation of S' from S'.

   (b) $\varphi \notin \Gamma'$ *or* $\psi \notin \Gamma'$.
   In this case we can use the inference $(\wedge \to)$ of $\mathbf{LJ}_2^{\approx}$:

$$\frac{\Gamma, \varphi \wedge \psi, \varphi, \psi \to \Delta}{\Gamma', \varphi \wedge \psi \to \Delta'} \ (\wedge \to).$$

2. The inference Π is by the rule $(\to \wedge)$:

$$\frac{\Gamma \to \Delta, \varphi \quad \Gamma \to \Delta, \psi}{\Gamma \to \Delta, \varphi \wedge \psi} \ (\to \wedge).$$

   Then S' has the form $\Gamma' \to \Delta', \varphi \wedge \psi$. Consider three cases.

   (a) $\varphi \in \Delta'$.
   In this case we have $(\Gamma \to \Delta, \varphi) \subseteq S'$. We can take Π' to be the 0-step derivation of S' from S'.

   (b) $\psi \in \Delta'$.
   This case is analogous to the previous one.

(c) $\varphi \notin \Delta'$ *and* $\psi \notin \Delta'$.
In this case we can use the inference $(\to \wedge)$ of $\mathbf{LJ}_2^{\approx}$:

$$\frac{\Gamma' \to \Delta', \varphi \wedge \psi, \varphi \quad \Gamma' \to \Delta', \varphi \wedge \psi, \psi}{\Gamma' \to \Delta', \varphi \wedge \psi}\ (\to \wedge).$$

3. The inference Π is by the rule $(\to\supset)$:

$$\frac{\Gamma, \varphi \to \psi}{\Gamma \to \Delta, \varphi \supset \psi}\ (\to\supset).$$

Then S' has the form $\Gamma' \to \Delta', \varphi \supset \psi$. Consider three cases.

(a) $\varphi \in \Gamma'$ *and* $\psi \in \Delta'$.
In this case we have $(\Gamma, \varphi \to \psi) \subseteq S'$. We can take Π' to be the 0-step derivation of S' from S'.

(b) $\varphi \notin \Gamma'$.
In this case we can use the inference $(\to\supset_1)$ of $\mathbf{LJ}_2^{\approx}$:

$$\frac{\Gamma', \varphi \to \psi}{\Gamma' \to \Delta', \varphi \supset \psi}\ (\to\supset_1).$$

(c) $\varphi \in \Gamma'$ *and* $\psi \notin \Delta'$.
In this case we can use the inference $(\to\supset_2)$ of $\mathbf{LJ}_2^{\approx}$:

$$\frac{\Gamma' \to \Delta', \varphi \supset \psi, \psi}{\Gamma' \to \Delta', \varphi \supset \psi}\ (\to\supset_2). \quad \Box$$

Lemma 3.27 If a sequent S is provable in $\mathbf{LJ}^{\approx}$, then it has a regular proof in $\mathbf{LJ}_2^{\approx}$.

Proof. Suppose S is provable in $\mathbf{LJ}^{\approx}$. By Lemma 3.24, S has a regular proof Π in $\mathbf{LJ}_1^{\approx}$. By repeated applications of Lemma 3.26, we can replace all inferences in Π by inferences of $\mathbf{LJ}_2^{\approx}$. It is easy to see that the obtained proof in $\mathbf{LJ}_2^{\approx}$ is regular. $\Box$

Let Π be a regular derivation. The *essential part* of Π is the figure obtained from Π by removing all applications of inessential rules. Note that since Π is regular, the essential part of Π is an essential derivation (in any of the systems introduced so far). We call the *depth* of a derivation Π the length of the longest branch in Π. We call the *essential depth* of a regular derivation Π the depth of the essential part of Π.

Lemma 3.28 Let φ be a quantifier-free formula with n subformulas and Π be an essential derivation of φ in $\mathbf{LJ}_2^{\approx}$. Then the depth of Π is at most n^2.

Proof. Observe the following. If

$$\frac{\Gamma_1 \to \Delta_1 \quad \ldots \quad \Gamma_n \to \Delta_n}{\Gamma \to \Delta}\ ,$$

$n = 1, 2$, is an inference in the essential part of Π, then for every $i \in \{1, \ldots, n\}$ one of the two conditions is satisfied:

1. the number of different formulas in each Γ_i is strictly greater than the number of different formulas in Γ; or

2. the number of different formulas each in Γ_i is equal to the number of different formulas in Γ, and the number of different formulas in Δ_i is strictly greater than the number of different formulas in Δ.

Since all formulas used in the derivation Π are subformulas of φ, this implies the statement of the lemma. □

Let $S = (\Gamma \to \Delta)$ be a sequent, $B_1, \ldots, B_n$ be all atomic formulas in Γ and $A_1, \ldots, A_m$ be all atomic formulas in Δ. Denote by *lit*(S) the formula

$$\neg B_1 \vee \ldots \vee \neg B_n \vee A_1 \vee \ldots \vee A_m,$$

if $m + n > 0$, and the formula $\perp$ if $m, n = 0$.

The following lemma characterizes provability by equality proofs.

Lemma 3.29 Let S be a sequent. Then S has an equality proof in any of the calculi $\mathbf{LJ}^{\approx}$–$\mathbf{LJ}_3^{\approx}$ if and only if *lit*(S) is provable in classical logic. □

The proof of this lemma is straightforward.

The calculus $\mathbf{LJ}_2^{\approx}$ has a polynomial depth derivation property for quantifier-free formulas and essential derivations. However, it is not directly applicable for the proof of the equivalence of Existencial Intuitionistic and Formula Instantiation. The problem is that derivations of propositional formulas are not closed under substitutions: there exists derivations Π and substitutions θ such that $\Pi\theta$ is not a derivation. We introduce one more calculus $\mathbf{LJ}_3^{\approx}$ that is similar to $\mathbf{LJ}_2^{\approx}$, but closed under substitutions.

The *calculus* $\mathbf{LJ}_3^{\approx}$ is obtained from $\mathbf{LJ}_2^{\approx}$ by dropping all restrictions to the inference rules and replacement of $(\to\supset_2)$ by a rule $(\to\supset_2')$ defined below. To define this new inference rule, we first introduce the notion of a disagreement between any two formulas.

Let φ, ψ be quantifier-free formulas. A *disagreement* between φ and ψ, denoted $D(\varphi, \psi)$ is the set of equations defined as follows.

1. If $\varphi = \psi = \perp$, then $D(\varphi, \psi) = \emptyset$.

2. If $\varphi = P(s_1, \ldots, s_n)$ and $\psi = P(t_1, \ldots, t_n)$, then $D(\varphi, \psi) = \{s_1 \approx t_1, \ldots, s_n \approx t_n\}$.

3. If $\varphi = \varphi_1 \wedge \varphi_2$, $\psi = \psi_1 \wedge \psi_2$, and $D(\varphi_1, \psi_1)$, $D(\varphi_2, \psi_2)$ are defined, then $D(\varphi, \psi) = D(\varphi_1, \psi_1) \cup D(\varphi_2, \psi_2)$, and similar for $\vee$ and $\supset$ instead of $\wedge$.

4. In all other cases $D(\varphi, \psi)$ is undefined.

It is not hard to show that for every substitution σ, $\varphi\sigma = \psi\sigma$ if and only if $D(\varphi, \psi)$ is defined and σ solves all equations in $D(\varphi, \psi)$.

The inference rule $(\to\supset_2')$ is defined as follows:

$$\frac{\Gamma, \varphi_1 \to \Delta, \varphi_2 \supset \psi, \psi \quad \to s_1 \approx t_1 \quad \ldots \quad \to s_n \approx t_n}{\Gamma, \varphi_1 \to \Delta, \varphi_2 \supset \psi} \; (\to\supset_2')$$

Where $D(\varphi_1, \varphi_2) = \{s_1 \approx t_1, \ldots, s_n \approx t_n\}$.

Using Lemmas 3.27 and 3.28, it is not hard to show the following.

Lemma 3.30 Every formula provable in $\mathbf{LJ}_3^{\approx}$ is provable in $\mathbf{LJ}^{\approx}$. Every formula φ provable in $\mathbf{LJ}^{\approx}$ has a regular derivation in $\mathbf{LJ}_3^{\approx}$ of essential depth at most n^2, where n is the number of subformulas in φ. $\square$

The next lemma reduces Formula Instantiation for intuitionistic logic to Formula Instantiation in classical logic.

Lemma 3.31 Let φ be a quantifier-free formula of a signature Σ. Then there exists a finite set of formulas $\{\psi_1, \ldots, \psi_m\}$ of the same signature with the following property. For every substitution θ, the formula $\varphi\theta$ is provable in intuitionistic logic if and only if for some $i \in \{1, \ldots, m\}$, the formula $\psi_i\theta$ is provable in classical logic.

Proof. Let n be the number of subformulas of φ (note that n is also the number of subformulas of every formula of the form $\varphi\theta$). By Lemma 3.30, if φ is provable in intuitionistic logic, then it has a regular derivation with the essential depth at most n^2. Consider all essential derivations of φ in $\mathbf{LJ}_2^{\approx}$ of depth $\leq n^2$. Denote these derivations by $\Pi_1, \ldots, \Pi_m$. Take any of these essential derivations Π_i. Let the top sequents of this derivation be $S_1, \ldots, S_k$. Then we define the formula ψ_i to be $lit(S_1) \wedge \ldots \wedge lit(S_k)$.

Let us now prove that the formulas ψ_i satisfy the statement.

($\Rightarrow$) Suppose that $\varphi\theta$ is provable in intuitionistic logic. By Lemma 3.30, $\varphi\theta$ has a regular proof Ξ in $\mathbf{LJ}_3^{\approx}$. Any such proof has the following form, where all Ξ_i are equality proofs and the essential part of Ξ is the derivation of $\varphi\theta$ from $S'_1, \ldots, S'_m$:

$$\begin{array}{ccc} \Xi_1 & & \Xi_m \\ S'_1 & \cdots & S'_m \\ & \to \varphi\theta & \end{array}$$

Since φ is quantifier-free, it is not hard to argue that there exists an essential derivation Π_i of φ in $\mathbf{LJ}_3^{\approx}$ from some sequents $S_1, \ldots, S_m$ such that $S'_j = S_j\theta$, for all $j \in \{1, \ldots, m\}$ (note that this property does not hold for $\mathbf{LJ}_2^{\approx}$).

Since the sequents $S'_1, \ldots, S'_m$ have equality proofs, by Lemma 3.29 the formula

$$lit(S'_1) \wedge \ldots \wedge lit(S'_m)$$

is provable in classical logic. It remains to note that this formula is precisely $\psi_i\theta$.

($\Leftarrow$) Suppose some $\psi_i\theta$ is provable in classical logic. By the definition of ψ_i, there exist sequents $S_1, \ldots, S_m$ and an essential derivation Π_i of φ in $\mathbf{LJ}_3^{\approx}$ from $S_1, \ldots, S_m$ such that ψ_i is $lit(S_1) \wedge \ldots \wedge lit(S_m)$. Since $\psi_i\theta$ is provable in classical logic, every formula $lit(S_j\theta)$ is also provable. By Lemma 3.29, every sequent $S_j\theta$ has a proof in $\mathbf{LJ}_3^{\approx}$. Since Π_i is an essential derivation in $\mathbf{LJ}_3^{\approx}$ of a quantifier-free formula φ from $S_1, \ldots, S_m$, we have that $\Pi_i\theta$ is an essential derivation in $\mathbf{LJ}_3^{\approx}$ of the formula $\varphi\theta$ from $S_1\theta, \ldots, S_m\theta$. Therefore, $\varphi\theta$ is provable in $\mathbf{LJ}_3^{\approx}$. By Lemma 3.30, $\varphi\theta$ is provable in $\mathbf{LJ}^{\approx}$. $\square$

The following property of intuitionistic provability can be easily proved by examining derivations in $\mathbf{LJ}^{\approx}$:

Lemma 3.32 Let $\exists \mathbf{x}\varphi$ be a sentence of a signature Σ. It is provable in $\mathbf{LJ}^{\approx}$ if and only if so is the formula $\varphi(\mathbf{t})$, for some terms $\mathbf{t}$ of the signature Σ. $\square$

Theorem 3.33 Let Σ be a signature. $\mathsf{EI}(\Sigma)^{\approx}$ is decidable if and only if so is $\mathsf{FI}(\Sigma)^{\approx}$.

Proof. Suppose $\mathsf{EI}(\Sigma)^{\approx}$ is decidable. We show an effective reduction of $\mathsf{SREU}(\Sigma)$ to $\mathsf{EI}(\Sigma)^{\approx}$. This would imply that $\mathsf{SREU}(\Sigma)$ is decidable. By Theorem 3.11, this will also imply the decidability of $\mathsf{FI}(\Sigma)$.

Let

$$\{E_i \vdash_{\forall} s_i \approx t_i \mid 1 \leq i \leq n\} \tag{8}$$

be an instance of $\mathsf{SREU}(\Sigma)$. Consider the formula

$$\varphi = \bigwedge_{i=1}^{n} (\bigwedge_{e \in E_i} e \supset s_i \approx t_i).$$

Using Lemma 3.20 on the existence of regular proofs in $\mathbf{LJ}^{\approx}$ and by routine inspection of regular derivations in $\mathbf{LJ}^{\approx}$, it is not hard to argue that for every substitution θ, θ is a solution to (8) if and only if $\varphi\theta$ is provable in $\mathbf{LJ}^{\approx}$. Let $\mathbf{x}$ be all variables of φ. By Lemma 3.32, $\exists \mathbf{x}\varphi$ is provable in $\mathbf{LJ}^{\approx}$ if and only if for some substitution θ, so is the formula $\varphi\theta$. Therefore, (8) is solvable if and only if $\exists \mathbf{x}\varphi$ is a yes-instance of $\mathsf{EI}(\Sigma)^{\approx}$.

Conversely, suppose $\mathsf{FI}(\Sigma)$ is decidable. Let $\exists \mathbf{x}\varphi(\mathbf{x})$ be an instance of $\mathsf{EI}(\Sigma)^{\approx}$, such that $\varphi(\mathbf{x})$ is quantifier-free. By Lemma 3.32, the formula $\exists \mathbf{x}\varphi(\mathbf{x})$ is provable in $\mathbf{LJ}^{\approx}$ if and only if there exists a substitution θ such that the formula $\varphi(\mathbf{x})\theta$ is provable in $\mathbf{LJ}^{\approx}$.

Construct by $\varphi(\mathbf{x})$ formulas $\psi_1, \ldots, \psi_n$ as in Lemma 3.31. By Lemma 3.31, $\varphi(\mathbf{x})\theta$ is provable in intuitionistic logic for some θ if and only if one of ψ_i is a yes-instance of $\mathsf{FI}(\Sigma)^{\approx}$. Thus, $\mathsf{EI}(\Sigma)^{\approx}$ is decidable. $\square$

Lemma 3.34 Let Σ be a signature with an infinite number of constants. Then $\mathsf{PI}(\Sigma)^{\approx}$ is polynomial-time equivalent to $\mathsf{EI}(\Sigma)^{\approx}$.

Proof. Since $\mathsf{EI}(\Sigma)^{\approx}$ is a special case of $\mathsf{PI}(\Sigma)^{\approx}$, it is enough to only show a reduction of $\mathsf{PI}(\Sigma)^{\approx}$ to $\mathsf{EI}(\Sigma)^{\approx}$. The proof is based on a "skolemization" property of intuitionistic logic with equality [11].

Let φ be a prenex sentence

$$Q_1 x_1 \ldots Q_n x_n \psi$$

in the signature Σ. Let Δ be the signature of φ and $c_1, c_2, \ldots, c_n$ be constants of Σ not occurring in Δ. Define two sets of numbers $E = \{i \mid Q_i \text{ is } \exists\}$ and $A = \{j \mid Q_j \text{ is } \forall\}$. For every $i \in E$ define the signature Δ_i as follows. The signature Δ_0 is Δ. For $i > 0$ let

$$\Delta_i = \Delta \cup \{c_j \mid j < i \text{ and } j \in A\}.$$

For any sequence of terms $T = (t_i)_{i \in E}$, define the substitution θ_T as follows:

$$x\theta_T = \begin{cases} t_i, & \text{if } x = x_i \text{ and } i \in E; \\ c_j, & \text{if } x = x_j \text{ and } i \in A; \\ x, & \text{otherwise.} \end{cases}$$

By routine inspection of derivations of φ in $\mathbf{LJ}^{\approx}$ one can prove the following:

($*$) *The formula φ is provable in $\mathbf{LJ}^{\approx}$ if and only if there exist a sequence of terms $T = (t_i)_{i \in E}$ in the signatures Δ_i, respectively, such that the formula $\psi\theta_T$ is provable in $\mathbf{LJ}^{\approx}$.*

Now introduce a substitution σ:

$$x\sigma = \begin{cases} c_j, & \text{if } x = x_j \text{ and } i \in A; \\ x, & \text{otherwise.} \end{cases}$$

Define the formula ψ' as follows:

$$\psi' = \psi\sigma \wedge \bigwedge_{i \in E} sig_{\Delta_i}(x_i).$$

By Lemmas 3.12 and 3.32, one can see that

($**$) *The formula $\exists\psi'$ is provable in $\mathbf{LJ}^{\approx}$ if and only if there exist a sequence of terms $T = (t_i)_{i \in E}$ in the signatures Δ_i, respectively, such that the formula $\psi\theta_T$ is provable in $\mathbf{LJ}^{\approx}$.*

Comparing ($*$) and ($**$), we can establish that the transformation $\varphi \Rightarrow \exists\psi'$ is a polynomial-time reduction of $\mathsf{PI}(\Sigma)^{\approx}$ to $\mathsf{EI}(\Sigma)^{\approx}$. □

In most cases one can get rid of the infinite number of constants in Lemma 3.34, by using the encodings $B(t)$ and $U(t)$:

Theorem 3.35 For every signature of the form $(\_, \_, \geq 1)$, or $(\_, \geq 2, 0)$, or $(\geq 2, 0, 0)$, the problem $\mathsf{PI}(\Sigma)^{\approx}$ is polynomial-time equivalent to $\mathsf{EI}(\Sigma)^{\approx}$.

Proof. (Sketch.) For $(\_, \_, \geq 1)$ use the binary encoding and the technique of the proof of Lemma 3.16. For $(\_, \geq 2, 0)$ use the unary encoding and the technique of the proof of Theorem 3.18. For $(\geq 2, 0, 0)$, the problem is PSPACE-complete. The sketch of the proof is given in [9], and is based in "guessing" the values of all variables among a finite number of constants. □

4 Undecidability results

Summarizing the results of Section 3, we obtain the following.

Theorem 4.1 Let Σ be a signature $(\_, \_, \geq 1)$. The following conditions are equivalent.

1. $\mathsf{HS}(\Sigma)^{\approx}$ is decidable;
2. $\mathsf{FI}(\Sigma)^{\approx}$ is decidable;
3. $\mathsf{MI}(\Sigma)^{\approx}$ is decidable;
4. $\mathsf{SI}(\Sigma)^{\approx}$ is decidable;
5. $\mathsf{PI}(\Sigma)^{\approx}$ is decidable;
6. $\mathsf{EI}(\Sigma)^{\approx}$ is decidable;

7. $\mathsf{SREU}(\Sigma)$ is decidable;
8. $\mathsf{SREU}(1,0,1)$ is decidable.

The same holds with the replacement of

- $(\_,\_,\geq 1)$ by $(\_,\geq 2,0)$ and $(1,0,1)$ by $(1,2,0)$; or
- $(\_,\_,\geq 1)$ by $(\omega,1,0)$ and $(1,0,1)$ by $(\omega,1,0)$. □

In this section we consider signatures with a binary function symbol. Signatures with unary function symbols are considered in Section 5.

4.1 Undecidability

Simultaneous Rigid E-unification witnessed several faulty proofs of the decidability in 1988–1994, until it was proved undecidable in [10] by reduction of so-called monadic semi-unification [2]. The technique used in the proof is, in fact, an encoding of Second-Order Unification [18]. Other undecidability proofs have been published as follows:

- using reduction of second-order unification [12];
- using reduction of Hilbert's Tenth problem [37, 36];
- using reduction of the Post Correspondence Problem [31];
- using an encoding of Turing machines [34].

The undecidability results can be formulated as follows.

Theorem 4.2 $\mathsf{SREU}(\_,\_,\geq 1)$ is undecidable. Hence, problems 1–7 for signatures $(\_,\_,\geq 1)$ are undecidable. □

Second-order unification Since the proofs are published elsewhere, we do not reproduce them here. We shall only demonstrate ideas of the simplest proof [12] since this proof shows close connections between Simultaneous Rigid E-unification and Second-Order Unification[4], the latter proved undecidable in [18].

We illustrate the reduction of [12] by an example. Consider an instance of Second-Order Unification:

$$x(y,g(y)) \approx f(g(z),a,y) \tag{9}$$

It has at least the following solutions:

$$\begin{array}{l}[x \mapsto \lambda w_1 w_2.f(w_2,a,w_1), y \mapsto b, z \mapsto b],\\ [x \mapsto \lambda w_1 w_2.f(g(a),a,b), y \mapsto b, z \mapsto a].\end{array} \tag{10}$$

[4] Margus Veanes (private communication) claims that there also exists a natural reduction of Simultaneous Rigid E-unification to Second Order Unification.

In general, the solutions to (9) are the substitutions

$$[x \mapsto \lambda w_1 \lambda w_2 . t_x, y \mapsto t_y, z \mapsto t_z]$$

that satisfy the following conditions:

$(a1)$ t_x is a term built from $\{f, g, a, w_1, w_2\}$;
$(a2)$ t_y is a term built from $\{f, g, a\}$;
$(a3)$ t_z is a term built from $\{f, g, a\}$;
$(a4)$ If we replace w_1 by t_y
and w_2 by $g(t_y)$,
then t_x becomes $f(g(t_z), a, t_y)$.

We shall treat w_1, w_2 as constants. Lemma 3.13 shows how to represent all terms of a given signature by rigid equations, allowing us to encode properties $(a1)$–$(a3)$.

To encode $(a4)$, we should be able to represent replacement of constants by terms. Let s be a term. Denote by $s^{t_1, \ldots, t_n}_{c_1, \ldots, c_n}$ the term obtained from s by simultaneous replacement of all occurrences of constants c_i by terms t_i, $i = 1, \ldots, n$.

Lemma 4.3 Let $c_1, \ldots, c_n$ be different constants and $t_1, \ldots, t_n$ be terms such that no c_i occurs in any t_j. Then

$$\vdash c_1 \approx t_1, \ldots, c_n \approx t_n \supset s \approx t$$

if and only if $s^{t_1 \ldots t_n}_{c_1 \ldots c_n} = t^{t_1 \ldots t_n}_{c_1 \ldots c_n}$.

Using Lemma 3.13 and this lemma, we can encode $(a1)$–$(a4)$ as follows:

$(b1)$ $f(a, a) \approx a, g(a) \approx a, c_1 \approx a, c_2 \approx a \vdash_\forall x \approx a$;
$(b2)$ $f(a, a) \approx a, g(a) \approx a \vdash_\forall y \approx a$;
$(b3)$ $f(a, a) \approx a, g(a) \approx a \vdash_\forall z \approx a$;
$(b4)$ $c_1 \approx y, c_2 \approx g(y) \vdash_\forall x \approx f(g(z), a, y)$.

Compare the following two solutions of this system with (10):

$$[x \mapsto f(c_2, a, c_1), y \mapsto b, z \mapsto b],$$
$$[x \mapsto f(g(a), a, b), y \mapsto b, z \mapsto a].$$

This technique can be generalized to encode arbitrary instances of Second-Order Unification.

4.2 Undecidable fragment

A stronger undecidability results is proved in [31]:

Theorem 4.4 Simultaneous rigid E-unification with ground left sides is undecidable. □

The technique of [31] had later been simplified and used in [34, 35] to prove an even stronger result:

Theorem 4.5 $\mathsf{SREU}(1,0,1)$ is undecidable for systems with ground left-hand sides, two variables and three rigid equations. □

The decidability of Simultaneous Rigid E-unification with two rigid equations is an open problem.

The restriction on left-hand sides to be ground corresponds to the so-called ground-negative formulas, introduced in [16] and independently in [41]. A formula φ is called *ground-negative* if all negative occurrences of atomic subformulas in φ are ground. Article [41] proves that the class of ground-negative formulas is decidable and Π_2^p-complete (with or without equality, and even in the signature with two constants). Surprisingly, the results of [31, 34] imply the following.

Theorem 4.6 (i) Formula Instantiation and Matrix Instantiation are undecidable for ground-negative formulas with two variables. (ii) The $\exists\exists$-fragment of intuitionistic logic with equality is undecidable for ground-negative formulas. (iii) Herbrand Skeleton is undecidable for ground-negative formulas. □

The significance of these results for automated reasoning with tableau-like methods are discussed in [40, 42].

5 The monadic case

We call *Monadic SREU* Simultaneous Rigid E-unification in any signature $(\_, \geq 1, 0)$. This section discusses results on Monadic SREU.

Proofs of results about monadic SREU use the technique of finite automata and word equations. The connection between Monadic SREU and word equations was noted already in [8]. The results explained in this section have been published in [9, 20, 21].

Terms as words. We consider signatures $\mathcal{F} \cup \mathcal{C}$ where $\mathcal{F}$ is finite set of unary function symbols and $\mathcal{C}$ is a finite set of constants. In this section we write terms in such signatures using the *reversed Polish notation*: $f_1(f_2(\dots f_n(c)))$ is written as $cf_n \dots f_2 f_1$.

Hence, every ground term in $\mathcal{F} \cup \mathcal{C}$ has the form cW where c is a constant and W is a word on $\mathcal{F}$. Using Lemma 3.13, we can express the property *"x starts with a constant c"*. Therefore, we can use *word variables* in rigid equations, e.g.

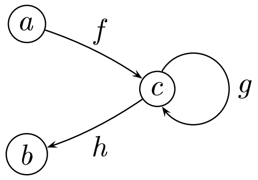
$$cvf \approx cu \vdash_\forall cg \approx cv$$

where u, v are word variables.

5.1 Representation of some word problems

In this section we show how to represent some word problems using monadic SREU.

Regular sets. For every regular set there exists a rigid equation whose solutions represent this regular set. For example, consider the deterministic finite automaton (DFA for short)



with the initial state a and final state b. Consider the rigid equation of one variable v:

$$af \approx c, cg \approx c, ch \approx b \vdash_{\forall} av \approx b.$$

(Compare this equation with the automaton.) Evidently, a substitution $[v \mapsto W]$ is a solution to this rigid equation if and only if W is recognized by the automaton. This observation can be used to represent an arbitrary regular set of words as the set of solutions to a rigid equation with ground left-hand sides and one variable:

Lemma 5.1 For every regular set $S \in \mathcal{F}^*$ there exists a rigid equation R with ground left-hand sides and one word variable v such that for every word V, we have $V \in S$ if and only if $[v \mapsto V]$ is a solution to R.

Word equations. The following simple observation shows that word equations can reduced to monadic SREU:

Lemma 5.2 $b \approx aV \vdash bU \approx aW$ holds if and only if $VU = W$. □

Power. The following simple observation shows how to represent that one word U is a power of another word V:

Lemma 5.3 $a \approx aV \vdash a \approx aU$ holds if and only if $U = V^n$ for some natural number n. □

Lemmas 5.1, 5.2 and 5.3 are important tools for studying monadic SREU.

5.2 Results

Here we cite main results of [9, 20, 21]. Some of these results have nontrivial proofs, so we describe only the main ideas of the proofs.

By Lemma 5.1, we can describe any regular set as a set of solutions to a rigid equation with a ground left-hand side and one variable. It is also not hard to represent solutions to such equations as regular sets. This implies that monadic SREU with ground left-hand sides and one variable is polynomial-time equivalent to the intersection nonemptiness problem for DFA. This proves the following [20]:

Theorem 5.4 Monadic SREU with one variable and ground left sides is PSPACE-complete. □

By more complex arguments, [20] also proves the following.

Theorem 5.5 Monadic SREU with one variable is PSPACE-complete. □

The proof is based on a technical lemma showing that solutions to a rigid equation with one variable can be searched among elements of a finite number of regular sets satisfying a rigid equation with ground left-hand sides. Then one can use Theorem 5.4.

As we have mentioned, [31] proves the undecidability of SREU with ground left sides. The corresponding monadic case is shown to be decidable [20]:

Theorem 5.6 Monadic SREU with ground left sides is decidable. □

The idea of the proof is the following. It is shown that solutions to rigid equations with ground left-hand sides can be described by word equations plus restrictions that word variables belong to regular sets. Then the result of [32] on the decidability of word equations with regular constraints is used.

As for the general case, the decidability remains an open problem. [20] proves polynomial-time equivalence of monadic SREU to a word problem described below.

Denote by $\mathbf{W}$ a set of pairs of words on $\mathcal{F}$. Introduce on $\mathbf{W}$ a binary function $*$, a unary function ^{r} and a binary relation $\leq$ in the following way:

$$\begin{array}{rcl} (U,V)*(V,W) & = & (U,W), \\ (U_1,U_2)^r & = & (U_2,U_1), \\ (U,V) & \leq & (UW,VW). \end{array}$$

An *ideal* on $\mathbf{W}$ is any set of pairs containing the pair of empty words $(\varepsilon,\varepsilon)$, closed under functions $*$ and ^{r} and upward closed under $\leq$. The *ideal generated by a set of pairs S*, denoted $ideal(S)$ is defined as the least ideal containing S.

An ideal membership question is an expression

$$(U,V) \in ideal(\{(U_1,V_1),\ldots,(U_n,V_n)\}),$$

where $n \geq 0$ and $U,V,U_1,\ldots,U_n,V_1,\ldots,V_n$ are words, possibly with variables. *The ideal membership problem* is the problem of solvability of ideal membership questions. [20] proves

Theorem 5.7 Monadic SREU id decidable if and only if so is the ideal membership problem. □

5.3 One unary function symbol

By Theorem 3.18, monadic SREU is polynomial-time equivalent to $\mathsf{SREU}(1,2,0)$. It is easy to see that $\mathsf{SREU}(\geq 2,0,0)$ is NP-complete. What happens in the case of signatures $(\_,1,0)$, i.e., signatures with one unary function symbol? This question is investigated in [9].

Assume that s is the only function symbol of such a signature. Then a word s^n represents the natural number n. Concatenation of words becomes the addition operation $+$ on numbers, the power relation "$s^k = (s^m)^n$ for some natural number n" represents the divisibility relation $m \mid k$. It follows from Lemmas 5.2 and 5.3 that Monadic SREU with one unary symbol can represent the so-called *Diophantine problem for addition and divisibility*, i.e., the problem of solvability of expressions using $0,1,+$ and $\mid$ over natural numbers.

In [9] a reduction of $\mathsf{SREU}(\_,1,0)$ to the Diophantine problem for addition and divisibility is given. The latter problem is known to be decidable [3, 26, 24]. This implies

Theorem 5.8 $\mathsf{SREU}(\_,1,0)$ is decidable. □

The precise complexity of $\mathsf{SREU}(\_,1,0)$ is not known, as well as the precise complexity of the Diophantine problem for addition and divisibility [25].

6 Another decidable fragment

Simultaneous Rigid E-unification with two variables is undecidable. For the one-variable case, the following result is known [6, 7]:

Theorem 6.1 $\mathsf{SREU}(\_,\_,\geq 1)$ with one variable is DEXPTIME-complete. □

The proof of [6] is based on polynomial-time reductions between Simultaneous Rigid E-unification and the intersection nonemptiness problem for bottom-up tree automata [33]. This theorem implies the following consequences for other decision problems discussed in our article.

Theorem 6.2 Formula Instantiation and Matrix Instantiation with one variable are decidable. The $\forall^*\exists\forall^*$-fragment of intuitionistic logic with equality is decidable. □

7 Conclusion

The known results on the decision problems explained in this article are summarized in Table 1.

The following two problems are still open:

- the decidability of Monadic SREU;
- The decidability of SREU with two rigid equations.

As is follows from the table, the precise complexity of some problems is also unknown.

Other open problems related to the Herbrand theorem and its use in automated reasoning are discussed in [42].

Acknowledgments

I thank all people who took part in the development of results explained in this paper: Anatoli Degtyarev, Yuri Gurevich, Yuri Matiyasevich, Paliath Narendran and Margus Veanes. I am also grateful to the editor of this volume Sergei Adian for his patience.

References

[1] P.B. Andrews. Refutations by matings. *IEEE Trans. Comput.*, 25:801–807, 1976.

[2] 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.

[3] A.P. Beltyukov. Decidability of the universal theory of natural numbers with addition and divisibility (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 60:15–28, 1976. English translation in: Journal of Soviet Mathematics.

[4] W. Bibel. On matrices with connections. *Journal of the Association for Computing Machinery*, 28(4):633–645, 1981.

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

[6] 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. To appear in Theoretical Computer Science.

restrictions / problem	no equality	no function symbols	one unary function symbol	monadic case	binary symbols	other special cases
Herbrand Skeleton	decidable	decidable	decidable	?	undecidable	
Formula Instantiation	Σ_2^p	Σ_2^p	decidable	?	undecidable	one variable: $[\mathrm{DEXP}, 2\text{-}\mathrm{NEXP}]$
Matrix Instantiation	the same as Formula Instantiation					
Skeleton Instantiation	NP	NP	decidable	?	undecidable	
Prenex Intuitionistic	PSPACE	PSPACE	decidable	?	undecidable	$\forall^*\exists\forall^*$-fragment: decidable
Existencial Intuitionistic	the same as Prenex Intuitionistic					$\exists\exists$-fragment: undecidable
SREU		NP	decidable	?	undecidable	one variable: DEXP two variables, ground left-hand sides and three rigid equations: undecidable one rigid equation: NP two rigid equations: ? monadic and ground left-hand sides: decidable

Table 1: Summary of results

[7] A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid E-unification with one variable. In *Rewriting Techniques and Applications, RTA'98*, page 15, 1998.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] C. Dwork, P.C. Kanellakis, and J.C. Mitchell. On the sequential nature of unification. *Journal of Logic Programming*, 1:35–50, 1984.

[14] C. Dwork, P.C. Kanellakis, and L.J. Stockmeyer. Parallel algorithms for term matching. *SIAM Journal of Computing*, 17(4):711–731, 1988.

[15] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *Journal of Symbolic Logic*, 57:795–807, 1992.

[16] C.F. Fermüller and A. Leitsch. Decision procedures and model building in equational clause logic. *Logic Journal of the IGPL*, 1998. To appear.

[17] 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.

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

[19] Yu. Gurevich and M. Veanes. Some undecidable problems related to the Herbrand theorem. UPMAIL Technical Report 138, Uppsala University, Computing Science Department, March 1997.

[20] Yu. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. UPMAIL Technical Report 137, Uppsala University, Computing Science Department, February 1997. Revised June 20, 1997. To appear in Theoretical Computer Science.

[21] Yu. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. In P. Degano, R. Corrieri, and A. Marchetti-Spaccamella, editors, *Automata, Languages and Programming. 24th International Colloquium, ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 154–165, Bologna, Italy, July 1997.

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

[23] J. Hudelmaier. An $O(n \log n)$-space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.

[24] L. Lipshitz. The Diophantine problem for addition and divisibility. *Transactions of the American Mathematical Society*, 235:271–283, January 1978.

[25] L. Lipshitz. Some remarks on the Diophantine problem for addition and divisibility. *Bull. Soc. Math. Belg. Sér. B*, 33(1):41–52, 1981.

[26] V.I. Mart'janov. Universal extended theories of integers. *Algebra i Logika*, 16(5):588–602, 1977.

[27] G.E. Mints. Herbrand theorem (in Russian). In A.V. Idelson and G.E. Mints, editors, *Mathematical Theory of Logical Inference*, pages 273–275. Nauka, Moscow, 1967.

[28] G.E. Mints. The Herbrand theorem for the predicate calculus with equality and function symbols. *Soviet Mathematical Doklady*, (2):273–275, 1969.

[29] V.P. Orevkov. A specialized form of derivation in Gentzen calculi and its applications (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 32:110–118, 1974.

[30] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[31] 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.

[32] K.U. Schulz. Makanin's algorithm: Two improvements and a generalization. In K.U. Schulz, editor, *Proceedings of the First International Workshop on Word Equations and Related Topics, Tübingen*, number 572 in Lecture Notes in Computer Science, 1990.

[33] M. Veanes. On computational complexity of basic decision problems of finite tree automata. UPMAIL Technical Report 126, Uppsala University, Computing Science Department, January 1996.

[34] M. Veanes. *On Simultaneous Rigid E-Unification.* PhD thesis, Uppsala University, 1997.

[35] M. Veanes. The undecidability of simultaneous rigid E-unification with two variables. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Computational Logic and Proof Theory. 5th Kurt Gödel Colloquium, KGC'97*, volume 1289 of *Lecture Notes in Computer Science*, pages 305–318, Vienna, Austria, 1997.

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

[37] A. Voronkov. Proof search in intuitionistic logic based on constraint satisfaction. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Theorem Proving with Analytic Tableaux and Related Methods. 5th International Workshop, TABLEAUX '96*, volume 1071 of *Lecture Notes in Artificial Intelligence*, pages 312–329, Terrasini, Palermo Italy, May 1996.

[38] A. Voronkov. Proof-search in intuitionistic logic based on the constraint satisfaction. UPMAIL Technical Report 120, Uppsala University, Computing Science Department, January 1996. Updated March 11, 1996.

[39] 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.

[40] A. Voronkov. Strategies in rigid-variable methods. In M.E. Pollack, editor, *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, volume 1, pages 114–119, Nagoya, Japan, August 23–29 1997.

[41] A. Voronkov. The ground-negative fragment of first-order logic is Π_2^p-complete. *Journal of Symbolic Logic*, 1998. To appear.

[42] A. Voronkov. Herbrand's theorem, automated reasoning and semantic tableaux. UPMAIL Technical Report 151, Uppsala University, Computing Science Department, February 1998.

[43] A. Voronkov. Proof-search in intuitionistic logic with equality, or back to simultaneous rigid E-unification. *Journal of Automated Reasoning*, 1998. To appear.

[44] H. Yasuura. On parallel computational complexity of unification. In *Proc. of the Conference on Fifth Generation Computer Systems*, pages 235–243. ICOT, 1984.