Herbrand's theorem, automated reasoning and semantic tableaux

Andrei Voronkov*
Computing Science Department
Uppsala University
Box 311, S 751 05, Uppsala, Sweden

Abstract

We overview recent results related to Herbrand's theorem and tableau-like methods of automated deduction and prove some new results. Based on an analysis and discussion of these results, new research directions are suggested.

1 Introduction

Numerous results related to the Herbrand theorem and the foundations of semantics tableaux were obtained recently. These results shed a new light on the nature of such methods and computational complexity of several problems related to the Herbrand theorem. The aim of this paper is to analyse these results, sketch new research directions and pose some open problems. We also prove several new results.

The main questions we ask in the paper are the following.

- Are tableaux and related methods inherently inefficient?
- In what cases are these methods efficient?
- In what cases and how can we increase efficiency of these methods?

This paper is structured as follows. In Section 3 we give a short description of the tableau and related method and consider a typical proof-search procedure for such methods. We analyze this procedure and point out two problems, instantiation of a quantifier-free formula to a valid formula, and strategies for selecting multiplicity. In Section 4 we prove several results on the complexity of the formula instantiation problem. Section 5 discusses some results about strategies for multiplicity. Both in Section 4 and Section 5 we mention some related open problems. Finally, Section 6 indicates some interesting research directions and defines more open problems in connections with these directions.

*Partially supported by TFR grants.

This paper does not pretend to be a historical survey. When we cite particular results, we try to choose most accessible publications which are often not the earliest ones. However, we always put the authors of a result first in the citation list for this result. When a theorem is new, it is numbered and provided with a proof (or a reference to the full version [54]). Theorems proved elsewhere are not numbered.

By semantic tableaux and related methods we understand:

1. the tableau method [27];
2. the method of matings [1, 28], or the connection method [7, 10];
3. model elimination [39, 40].

There is no unifying name for these kind of methods. They may be called matrix methods [15] due to Prawitz [47], extension methods [9, 45], non-local methods [42] or rigid-variable methods [53]. We shall call them *tableau-like methods* and use matings for illustration.

An extended version of this article with more complete presentation and proofs is available as [54].

2 Preliminaries

In this section we introduce some notation and terminology. We use boldface notation to denote sequences of symbols. For example, t is used to denote terms and $\mathbf{t}$ is used to denote sequences of terms. The equality predicate is denoted $\approx$. The symbol $\vdash$ denotes provability in classical logic. An expression (e.g. a term or a set of terms) is called *ground* if it contains no variables.

Substitutions. By $[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$ we denote the substitution that replaces variables x_i by terms t_i. If θ is a substitution and t is a term, $t\theta$ denotes the term obtained from t by applying θ. We shall also apply substitution to formulas. When applying a substitution θ to a formula φ, we rename, if necessary, bound variables of φ. The domain of a substitution θ is denoted $dom(\theta)$. The *variable range of* θ, denoted $vran(\theta)$, is the set of variables occurring in $(dom(\theta))\theta$.The restriction of a substitution θ to a set of variables X is denoted $\theta\restriction_X$.

Formulas. By formulas we mean first-order formulas. For every formula φ, we denote by $\exists\varphi$ the *existential closure* of φ. A *literal* is either an atomic formula or the negation of an atomic formula. A formula is in *negation normal form* if it is constructed from literals using $\wedge, \vee, \forall, \exists$.

Signatures. 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. Our results will depend on the signature and logic in which formulas are written. We consider two logics: classical 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). In the case of logic with equality, the corresponding signature will be denoted $(k, l, m)^{\approx}$. 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, $(< \omega, 0, 0)^{\approx}$ denotes a signature with any finite number of constants and no function symbols of arity ≥ 1, considered in logic with equality.

Theories. A *theory* is any set of first-order sentences. We denote by $\vdash \varphi$ (respectively, $\mathcal{T} \vdash \varphi$) provability of φ (respectively from $\mathcal{T}$) in first-order classical logic.

3 Tableaux and related methods

We assume that formulas to be checked for provability are prenex existential sentences, i.e. formulas of the form

$$\exists x_1 \ldots \exists x_n \varphi(x_1, \ldots, x_n), \tag{1}$$

where $\varphi(x_1, \ldots, x_n)$ is a quantifier-free formula all whose free variables are among $x_1, \ldots, x_n$.

Given a formula (1), the method of matings constructs a *matrix*, using an appropriate *multiplicity*. Multiplicity is, intuitively, the number of copies of $\varphi(x_1, \ldots, x_n)$ that can be used in the search for a proof.

For example, for the input formula

$$\exists x(P(x) \vee (\neg P(a) \wedge \neg P(b))), \tag{2}$$

using two copies of $\exists x$ we get the following matrix

$$\begin{bmatrix} P(x_1) \\ \begin{bmatrix} \neg P(a) & \neg P(b) \end{bmatrix} \\ P(x_2) \\ \begin{bmatrix} \neg P(a) & \neg P(b) \end{bmatrix} \end{bmatrix}$$

This matrix represents two variable-disjoint copies of $P(x) \vee (\neg P(a) \wedge \neg P(b))$, namely

$$P(x_1) \vee (\neg P(a) \wedge \neg P(b)) \text{ and}$$
$$P(x_2) \vee (\neg P(a) \wedge \neg P(b)),$$

with disjunctions displayed vertically and conjunctions displayed horizontally.

The method of matings is based on the following characterization of provability:

Theorem 1 A formula φ is provable if there exist a multiplicity μ and a substitution θ so that every vertical path in $M\theta$ is a tautology, where M is the matrix constructed by φ and μ.

The tableau method uses trees instead of matrices. The notion of a multiplicity is representation-dependent and may vary. For example, model elimination provers often use bounded-depth search. The depth bound is an example of another notion of multiplicity. For nonprenex formulas, the multiplicity can be a tree of integers, because we can use different subformulas a different number of times.

The results of this paper can be reformulated for different notions of multiplicity and different characterizations of provability. We choose one particular characterization and the corresponding representation-independent notion. This characterization is nothing but Herbrand's theorem.

Theorem 2 (Herbrand [33]) A formula $\exists\mathbf{x}\varphi(\mathbf{x})$, where $\varphi(\mathbf{x})$ is quantifier-free, is provable iff there exists a positive integer μ and a substitution θ such that $(\varphi(\mathbf{x}_1) \vee \ldots \vee \varphi(\mathbf{x}_\mu))\theta$ is provable.

The matrix characterization of provability is just a reformulation of Herbrand's theorem. Now we call *multiplicity* any positive integer and *matrix* any quantifier-free formula.

Given as input a prenex existential formula $\exists \mathbf{x} \varphi(\mathbf{x})$, implementations of matrix methods usually follow the following loop or its variations:

1. Choose an initial multiplicity μ.

2. Consider the matrix $\varphi_\mu = (\varphi(\mathbf{x}_1) \vee \ldots \vee \varphi(\mathbf{x}_\mu))$ and check whether there exists a substitution θ such that the formula $\varphi_\mu \theta$ is provable.

3. If such a substitution exists, then the goal formula $\exists \mathbf{x} \varphi(\mathbf{x})$ is provable. Otherwise, increase μ and come back to step 2.

We refer to this loop in the sequel as the loop. Our aim is to understand the efficiency of methods based on this loop and ways to improve performance of such methods. To this end we analyze the following two problems.

1. How difficult is the check for the existence of the substitution θ (step 2)?

2. What kind of strategies for increasing multiplicity should be used? Are some strategies more efficient than the other?

Some analysis of these two problems is made in the next two sections.

4 Formula instantiation

The problem that should be solved at step 2 of the loop gives rise to the following decision problem.

Definition 4.1 (Formula Instantiation) *Formula Instantiation* is the following decision problem. Given a quantifier-free formula $\varphi(\mathbf{x})$, is there a term sequence $\mathbf{t}$ such that $\varphi(\mathbf{t})$ is provable?

In fact, we have a family of Formula Instantiation problems, depending on the signature and presence of equality. Formula Instantiation for the formulas of a signature Σ will be denoted $FI(\Sigma)$ for logic without equality, and $FI(\Sigma)^{\approx}$ for logic with equality.

To consider some particular theories as built-in, we introduce a more general Formula Instantiation.

Definition 4.2 *Formula Instantiation* with respect to a theory T is the following decision problem. Given a quantifier-free formula $\varphi(\mathbf{x})$, is there a term sequence $\mathbf{t}$ such that $T \vdash \varphi(\mathbf{t})$.

When we consider a provability in a particular theory instead of provability in first-order logic, we do not necessarily have an analogue of Herbrand's theorem. Herbrand's theorem takes place for theories axiomatized by universally quantified formulas. Automated deduction with respect to a particular theory is considered in a number of papers, for example in [49].

As a particular example we can consider the theory of transitive relations that plays an important role in automated reasoning [2] in connection with axiomatization of orderings.

First, we note a trivial but useful property of Formula Instantiation.

Lemma 4.3 Let $\varphi(\mathbf{x})$ be a quantifier-free formula. If there exists a term sequence $\mathbf{t}$ such that $\varphi(\mathbf{t})$ is provable, then there exists such a term sequence in the signature of φ. $\square$

Of course this lemma does not hold when provability is replaced by provability in a particular theory. To make it hold for a theory T, it is enough to require that T is axiomatized by universally quantified formulas, and consider the signature of T and φ instead of the signature of φ.

4.1 Logic without equality

There exist several procedures for solving Formula Instantiation in logic without equality. Usually, they are presented in the form of searching for a substitution making all paths in a matrix tautologies [28, 10] or closing all branches in a tableau [27]. The following theorem shows the complexity of Formula Instantiation for logic without equality.

Theorem 4.4 For every signature Σ with at least two symbols, $FI(\Sigma)$ is decidable and Σ_2^p-complete.

Proof. First, we show that Formula Instantiation 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 symbols.

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 formula (2). Suppose that $\varphi(x_1, x_2)$ is the formula

$$\begin{array}{l}\big(P(x_1) \vee (\neg P(a) \wedge \neg P(b))\big) \vee \\ \big(P(x_2) \vee (\neg P(a) \wedge \neg P(b))\big).\end{array}$$

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, if we replace the atoms $P(a), P(x_1)$ by a propositional letter A and the atoms $P(b), P(x_2)$ by a propositional letter B, we obtain the tautology $\varphi_S = \big(A \vee (\neg A \wedge \neg B)\big) \vee \big(B \vee (\neg A \wedge \neg B)\big)$. 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 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 coNP-complete (by obvious reductions to and of the propositional provability problem).

3. The problem of the existence of an appropriate substitution θ can be solved in polynomial time (it is the so-called *simultaneous unification problem* proved P-complete in [21, 55, 22]).

It remains to prove that Formula Instantiation is Σ_2^p-hard. We prove this for the signature $(2, 0, 0)$, but the proof is valid for every other signature with at least two symbols, by replacing two constants of this signature with any two distinct ground terms. We use reduction of the QBF_2 problem known to be Σ_2^p-complete, see e.g. [44]. 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

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \Phi(x_1, \ldots, x_n, y_1, \ldots, y_m), \quad (3)$$

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

Given an instance (3) of QBF_2, we construct the corresponding instance of Formula Instantiation as follows. The instance uses unary predicate symbols $P_1, \ldots, P_n$ and propositional symbols $Q_1, \ldots, Q_m$. Denote by Φ' the formula obtained from Φ by replacing

- each occurrence of $x_i \approx 0$ by $\neg P_i(x_i)$;
- each occurrence of $x_i \approx 1$ by $P_i(x_i)$;
- each occurrence of $y_j \approx 0$ by $\neg Q_j$;
- each occurrence of $y_j \approx 1$ by Q_j.

Consider the instance of Formula Instantiation

$$\bigwedge_{1 \leq i \leq n} (\neg P_i(0) \wedge P_i(1)) \supset \Phi'. \quad (4)$$

We claim that (3) holds if and only if (4) does.

By Lemma 4.3, if an instance of Formula Instantiation 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.

It is not hard to argue that θ satisfies (4) if and only if

$$\mathbf{2} \models \forall y_1 \ldots \forall y_m \Phi(x_1\theta, \ldots, x_n\theta, y_1, \ldots, y_m).$$

This proves the equivalence of (3) and (4). $\square$

When the signature contains only one constant and no function symbols, the complexity is characterized by the following statement proved in [54].

Theorem 4.5 *FI*$(1, 0, 0)$ and *FI*$(1, 0, 0)^{\approx}$ are coNP-complete. $\square$

4.2 Equality and other relations

For logic with equality, Formula Instantiation is decidable for formulas of a signature Σ if and only if simultaneous rigid E-unification [29] is decidable for formulas of this signature. Most (un)decidability results cited below were proved about simultaneous rigid E-unification, we reformulate them for Formula Instantiation.

Theorem 3 (Degtyarev and Voronkov [18, 19]) *FI*$(\_, \_, \geq 1)^{\approx}$ is undecidable. It is undecidable already for formulas with two variables [50]. $\square$

As for the number of variables, this undecidability result is sharp, as the following theorem shows.

Theorem 4 (Degtyarev et.al. [12]) *FI*$(\_,\_,\_)^{\approx}$ is decidable for formulas with one variable. $\Box$

[12] prove that simultaneous rigid E-unification with one variable in the signature $(\_,\_,\geq 1)$ is EXPTIME-complete. This gives the 2-NEXPTIME upper bound and EXPTIME lower bound on the complexity of *FI*$(\_,\_,\leq 1)^{\approx}$ for formulas with one variable. The precise complexity is however not known:

Open problem 1 What is the complexity of *FI*$(\_,\_,\geq 1)^{\approx}$? for formulas with one variable?

All proofs of the undecidability of Formula Instantiation for logic with equality essentially use a function symbol of arity ≥ 2. For Monadic Formula Instantiation, i.e. Formula Instantiation with unary function symbols only, the decidability is an open problem. Some known results are the following. In the case of one unary function symbol, we have

Theorem 5 (Degtyarev, Matiyasevich, and Voronkov [14]) *FI*$(\_,1,0)^{\approx}$ is decidable. $\Box$

The decidability proof of this theorem essentially uses the decidability of the so-called *Diophantine problem for addition and divisibility* proved in [5, 41, 37], i.e. solving equations over natural numbers in the signature consisting of the function $+$ and the divisibility predicate $|$. Establishing complexity of *FI*$(\_,1,0)^{\approx}$ will most likely require a complexity result for the Diophantine problem for addition and divisibility. The complexity of the latter problem was investigated in [38], but it is only known that the problem is NP-hard. Thus, we have

Open problem 2 What is the complexity of *FI*$(\_,1,0)^{\approx}$?

For Monadic Formula Instantiation with more than one function symbol, the following results are known. Using a standard encoding of words on an arbitrary alphabet by words on a two-letter alphabet, one can easily prove

Theorem 6 (Degtyarev, Matiyasevich, and Voronkov [14]) *FI*$(\_,2,0)^{\approx}$ is polynomial-time equivalent to *FI*$(\_,\geq 2,0)^{\approx}$. $\Box$

Moreover, [14] noted that there is a polynomial-time reduction of word equations to *FI*$(\_,2,0)^{\approx}$. The complexity of word equations is between NP and 3-NEXPTIME [35]. A more precise correspondence between word equations is described in [31, 32], where an extension of word equations called *the ideal membership problem* is defined. We have

Theorem 7 (Gurevich and Voronkov [31, 32]) The ideal membership problem is decidable if and only if *FI*$(\_,\geq 2,0)^{\approx}$ is decidable. $\Box$

It is known that the if ideal membership problem is decidable, then it is also decidable with regular constraints (every word variable ranges over a regular set) [31]. However, this does not give the undecidability result for *FI*$(\_,\geq 2,0)^{\approx}$, since word equations with regular constraints are decidable [48]. Thus, we have

Open problem 3 Is *FI*$(\_,\geq 2,0)^{\approx}$ decidable?

This problem is also related to the classification of decidable prenex fragments of intuitionistic logic depending on the signature and the quantifier prefix [16, 13].

As for Formula Instantiation with equality but no function symbols, we shall establish its complexity after proving some quite general results.

Denote by R, S, T axioms of reflexivity, symmetry and transitivity, respectively, for some predicate.

Theorem 4.6 Let $\mathcal{T}$ be a theory in logic without equality axiomatized by any combination of R, S, T for any number of predicates. Then Formula Instantiation with respect to $\mathcal{T}$ is Σ_2^p-complete.

Proof. We only sketch the proof. The complete proof can be found in [54].

When we have a predicate symbol for which no axioms are specified, Σ_2^p-hardness follows from Theorem 4.4. When every predicate symbol of the language satisfies one or more axioms R, S, T, the proof of Σ_2^p-hardness can still be modified and we do not consider details here. We only prove containment in Σ_2^p.

For simplicity, but without loss of generality, we assume that the language contains only one predicate P and the theory $\mathcal{T}$ consists of the transitivity axiom for P, i.e., $\forall x \forall y \forall z (P(x,y) \wedge P(y,z) \supset P(x,z))$. Let φ be any quantifier-free formula. First, we characterize Formula Instantiation for φ in terms of so-called *argument splits.*

Denote by T the set of terms occurring as arguments to a predicate occurring in φ. For example, if φ contains the subformula $P(a, f(x))$, then the terms a and $f(x)$ are members of T. We call an *argument split* any equivalence relation on T. Given an argument split S, we obtain from φ a formula φ_S as follows. Replace arguments of predicates in φ by constants so that terms belonging to the same equivalence class of S are replaced by the same constant and terms belonging to different equivalence classes are replaced by different constants.

Let θ be any substitution. The *split by* θ, denoted $S(\theta)$, is defined as follows. Two terms $s, t \in T$ are equivalent w.r.t. $S(\theta)$ if and only if $s\theta = t\theta$.

Lemma 4.7 $\mathcal{T} \vdash \varphi\theta$ if and only if $\mathcal{T} \vdash \varphi_{S(\theta)}$. $\square$

The proof can be found in [54].

For splits S and S', we write $S \leq S'$ if every equivalence class by S is a subset of an equivalence class by S'. For example, for three splits

$$S_1 = \{\{a\}, \{b, c\}\},\ S_2 = \{\{a, b\}, \{c\}\},\ S_3 = \{\{a, b, c\}\}$$

we have $S_1 \leq S_3$ and $S_2 \leq S_3$, but S_1 and S_2 are incomparable w.r.t. $\leq$.

Lemma 4.8 If $S \leq S'$ and $\mathcal{T} \vdash \varphi_S$, then $\mathcal{T} \vdash \varphi_{S'}$.

Proof. The formula $\varphi_{S'}$ is obtained from φ_S by replacement of some constants c_i by other constants c_j. It is not hard to argue that every model of $R + \neg\varphi_{S'}$ can be made into a model of $R + \neg\varphi_S$. $\square$

An argument split S is called a *justification of* φ if (i) $\mathcal{T} \vdash \varphi_S$; and (ii) there exists a substitution θ such that for every equivalence class E by S and every terms $s, t \in E$ we have $s\theta = t\theta$.

Lemma 4.9 There exists a substitution θ such that $\mathcal{T} \vdash \varphi\theta$ if and only if there exists a justification of φ. $\square$

The proof uses Lemma 4.7 and can be found in [54]. Using the existence of a polynomial-time algorithm for finding the transitive closure, we can prove

Lemma 4.10 Let U be a finite set of ground literals. The problem of checking whether U has a model where P is a transitive relation, can be checked in polynomial time. $\square$

Lemma 4.11 Let ψ be a quantifier-free ground formula. The problem of checking whether $\mathcal{T} \vdash \psi$ is in coNP.

Proof. The formula ψ is equivalent to its conjunctive normal form, i.e. a formula $\psi' = \psi_1 \wedge \ldots \wedge \psi_n$ where each ψ_i is a disjunction of literals. It is not hard to argue that all disjunctions ψ_i can be found by a nondeterministic polynomial-time algorithm and that $\mathcal{T} \vdash \psi$ if and only if for every ψ' we have $\mathcal{T} \vdash \psi'$. Evidently, $\mathcal{T} \nvdash \psi'$ if there exists a model of $\neg\psi'$ in which P is a transitive relation. Using Lemma 4.10, the existence of such a model can be checked in polynomial time. $\square$

Lemma 4.12 The problem of checking, given φ and a split S, that S is a justification of φ, is in coNP.

Proof. The check $\mathcal{T} \vdash \varphi_S$, is in co-NP, by Lemma 4.11. The existence of a substitution θ such that for every equivalence class E by S and every terms $s, t \in E$ we have $s\theta = t\theta$ can be made in polynomial time by a simultaneous unification algorithm. $\square$

Now we can complete the proof of Theorem 4.6. By Lemma 4.9, there exists a substitution θ such that $\mathcal{T} \vdash \varphi\theta$ if and only if there exists a justification of φ. All splits can be generated using a nondeterministic polynomial-time algorithm. By Lemma 4.12, the problem whether a split is a justification is in coNP. Hence, Formula Instantiation with respect to $\mathcal{T}$ is in Σ_2^p. $\square$

Theorem 4.13 $FI(\geq 2, 0, 0)^{\approx}$ is Σ_2^p-complete.

Proof. In the absence of function symbols, the theory of equality $\mathcal{T}$ equality is axiomatized as an equivalence relation, i.e. by reflexivity, transitivity and symmetry axioms. In addition, for every predicate symbol P different from equality, $\mathcal{T}$ contains the formula

$$x_1 \approx y_1 \wedge \ldots \wedge x_n \approx y_n \wedge P(x_1, \ldots, x_n) \supset P(y_1, \ldots, y_n).$$

The proof is the same as that of Theorem 4.6. $\square$

Theorem on the undecidability of Formula Instantiation for logic with equality buried hopes for handling equality in tableau-like methods in the ordinary way. After this theorem was proved, several papers with nonstandard approaches to equality appeared, for example

- The equality elimination method: a combination of tableaux and resolution [15, 17].
- A goal-oriented strategy for model elimination [43].
- Incomplete but terminating procedures for Formula Instantiation [46, 20].
- Optimized translations of logic with equality into logic without equality [3].

5 Strategies for multiplicity

In this section we present in a unified framework some previously published results about strategies for multiplicity and prove some new results.

Informally, a strategy for multiplicity is any procedure that selects the initial multiplicity and increases multiplicity in the loop. Therefore, such a strategy can be formalized as a function producing, by a given formula, an increasing sequence of multiplicities. Since a sequence it itself a function on positive integers, we define strategies for multiplicity as follows.

Definition 5.1 (Strategy for multiplicity) A *strategy for multiplicity* is a computable binary function $\mathfrak{S}$, whose first argument ranges over quantifier-free formulas and second argument ranges over positive integers. The range of $\mathfrak{S}$ is the set of positive integers and $\mathfrak{S}$ is strictly monotone on the second argument, i.e. $\mathfrak{S}(\varphi, n) > \mathfrak{S}(\varphi, m)$ whenever $n > m$.

The monotonicity requirement is due to the fact that the multiplicity in the loop increases with every iteration. When we fix a strategy $\mathfrak{S}$, we can reformulate the loop as follows. Given as the input a prenex existential formula $\exists \mathbf{x} \varphi(\mathbf{x})$, do the following.

1. Let $i = 1$ (to denote the iteration step).
2. Define $\mu = \mathfrak{S}(\varphi(\mathbf{x}), i)$. Consider the matrix $\varphi_\mu = (\varphi(\mathbf{x}_1) \vee \ldots \vee \varphi(\mathbf{x}_\mu))$ and check whether there exists a substitution θ such that the formula $\varphi_\mu \theta$ is provable.
3. If such a substitution exists, then the goal formula $\exists \mathbf{x} \varphi(\mathbf{x})$ is provable. Otherwise, increase i by 1 and come back to step 2.

Now we can investigate decidability and complexity problems for this loop, depending on the strategy $\mathfrak{S}$.

We introduce one special kinds of strategies.

Definition 5.2 A strategy $\mathfrak{S}$ is called *formula-independent* if for every formulas φ, ψ and a natural number n we have $\mathfrak{S}(\varphi, n) = \mathfrak{S}(\psi, n)$. The *standard strategy* $\mathfrak{S}$ is defined by $\mathfrak{S}(\varphi, n) = n$, for every φ.

Thus, a strategy is formula-independent if it does not depend on the input formula. The standard strategy sets multiplicity initially to 1 and increments it by 1 at any further step. Evidently, the standard strategy is formula-independent.

If we choose multiplicity using a strategy, we can consider the following family of decision problems.

Definition 5.3 (Herbrand Instantiation) Let $\mathfrak{S}$ be a strategy. *Herbrand Instantiation using* $\mathfrak{S}$, denoted $HI(\mathfrak{S})$, is the following decision problem. Given a quantifier-free formula $\varphi(\mathbf{x})$ and a positive integer i (written in unary notation), are there term sequences $\mathbf{t}_1, \ldots, \mathbf{t}_{\mathfrak{S}(\varphi(\mathbf{x}), i)}$ such that the formula $\varphi(\mathbf{t}_1) \vee \ldots \vee \varphi(\mathbf{t}_{\mathfrak{S}(\varphi(\mathbf{x}), i)})$ is provable? Similarly, we denote by $HI(\mathfrak{S})^{\approx}$ Herbrand Instantiation using $\mathfrak{S}$ for logic with equality.

We shall always denote the arguments of a strategy $\mathfrak{S}$ by φ and n and denote the size of φ of $|\varphi|$.[1] For example, $HI(2^{n|\varphi|})$ denotes Herbrand Instantiation using the strategy $\mathfrak{S}(\varphi, n) = 2^{n|\varphi|}$.

Evidently, for every signature in which $FI(\Sigma)^{\approx}$ is decidable, the problem $HI(\mathfrak{S})^{\approx}$ is also decidable.

[51] introduced a decision problem called Herbrand Skeleton.

Definition 5.4 (Herbrand Skeleton) Let k be a positive integer. Herbrand Skeleton for multiplicity k, denoted HS_k (and $HS_k^{\approx}$ for logic with equality), is the following decision problem. Given a quantifier-free formula $\varphi(\mathbf{x})$, is there a substitution θ such that the formula $\big(\varphi(\mathbf{x}_1) \vee \ldots \vee \varphi(\mathbf{x}_k)\big)\theta$ is provable?

Note that Formula Instantiation can be described as HS_1 (or $HS_1^{\approx}$ for formulas with equality). [30] also give a reduction of $HS_1^{\approx}$ to $HS_k^{\approx}$. This reduction also works for many fragments of predicate calculus.

[51] prove that $HS_k^{\approx}$ is undecidable, for every k, for formulas with equality. They note that the Herbrand Skeleton problem plays an important role in automated deduction in tableau-like methods. As we shall see from the following theorem, this problem can be reformulated as a statement about formula-independent strategies.

Theorem 5.5 If $HS_k^{\approx}$ for a class Φ of quantifier-free formulas is undecidable for every k, then $HI(\mathfrak{S})^{\approx}$ for the class Φ is undecidable for every formula-independent strategy $\mathfrak{S}$.

Proof. Suppose, by contradiction, that $\mathfrak{S}$ is a formula-independent strategy such that $HI(\mathfrak{S})^{\approx}$ is decidable. Define the function f by $f(n) = \mathfrak{S}(\varphi, n)$, where φ is an arbitrary formula. The problem $HI(\mathfrak{S})^{\approx}$ can be reformulated as the following decision problem. Given a quantifier-free formula $\varphi(\mathbf{x})$ and a positive integer i, are there term sequences $\mathbf{t}_1, \ldots, \mathbf{t}_{f(i)}$ such that the formula $\varphi(\mathbf{t}_1) \vee \ldots \vee \varphi(\mathbf{t}_{f(i)})$ is provable? Since this

[1] At the moment any reasonable definition of the size of a formula is enough for our purpose.

problem is decidable, its specialization for each particular i is also decidable. By specializing this problem for $i = 0$ we obtain $HS^{\approx}_{f(0)}$. Therefore, $HS^{\approx}_{f(0)}$ is undecidable. Contradiction. □

As a consequence, we have

Theorem 5.6 $HI(\mathfrak{S})^{\approx}$ is undecidable for every formula-independent strategy $\mathfrak{S}$. □

When we proved the undecidability of Formula Instantiation for logic with equality, we received comments that predicate logic in undecidable in any case, so the undecidability of Formula Instantiation does not add much complexity to tableau-like methods.

Let us now consider how Formula Instantiation and Herbrand Instantiation behave on a *decidable* class of so-called ground-negative formulas. A formula φ is ground-negative [53] if all negative atomic subformulas of φ are ground (i.e. variable-free). The class of ground-negative formulas is decidable.

Theorem 8 (Voronkov [52]) The provability problem for ground-negative formulas is decidable and Π^p_2-complete, with or without function symbols and with or without equality. □

Theorem 9 (Voronkov [53]) For all formula-independent strategies $\mathfrak{S}$, the problem $HI(\mathfrak{S})^{\approx}$ is undecidable for the class of ground-negative formulas. □

This theorem is a consequence of Theorem 5.5 and the undecidability of all $HS^{\approx}_k$ for ground-negative formulas [30]. The theorem shows that for a "simply decidable" fragment of predicate calculus we have to solve undecidable subproblems, when the traditional loop is used. Note that [26] proved that the class of ground-negative formulas can be decided by traditional resolution-based methods, namely by positive hyperresolution with ordered paramodulation.

The next question we address is the following. Can formula-dependent strategies be better than formula-independent ones? We shall prove that this is the case for the ground-negative formulas.

Theorem 10 (Voronkov [53]) $HI(2^{|\varphi|} + n)$ is Π^p_2-complete for ground-negative formulas.

Instead of $2^{|\varphi|} + n$ we could take any strategy that chooses the initial multiplicity $\geq 2^p$, where p is the number of negative atomic subformulas in φ. This theorem shows that more intelligent strategies can lead to much simpler problems (at least for some classes of formulas).

6 Open problems

It will be of a great importance for automated reasoning by tableau-like methods to find intelligent strategies for multiplicity. The problem of finding such strategies was noted as early as in [6]. Our results show that such intelligent strategies exist at least for some decidable classes of predicate logic.

Theorem 5.6 shows that $HI(\mathfrak{S})^{\approx}$ is undecidable for every formula-independent strategy $\mathfrak{S}$. The following problem is open.

Open problem 4 Is there a strategy for multiplicity $\mathfrak{S}$ such that $HI(\mathfrak{S})^{\approx}$ is decidable?

The answer is most likely "no", but we could not find a proof. One can easily show that such a strategy exists for every decidable class of formulas.

Theorem 6.1 For every decidable class Φ of formulas there exists a strategy for multiplicity $\mathfrak{S}$ such that $HI(\mathfrak{S})^{\approx}$ is decidable for the class Φ.

Proof. We only define such a strategy $\mathfrak{S}(\varphi, n)$ leaving the proof of the decidability to the reader. Using a decision algorithm for the class Φ, check whether $\exists\varphi$ is provable. If not, define $\mathfrak{S}(\varphi, n) = n$, for all n. If $\exists\varphi$ is provable, find any proof of this formula (for example by the enumeration of all possible proofs in predicate calculus or all resolution refutations of $\neg\varphi$). Extract, using some algorithm, from this proof a number μ such that $\varphi(\mathbf{x}_1) \vee \ldots \vee \varphi(\mathbf{x}_\mu)$ is provable and define $\mathfrak{S}(\varphi, n) = \mu + n$. □

This proof does not expose any *efficient* (in whatever sense) strategy $\mathfrak{S}$, it just shows that $HI(\mathfrak{S})^{\approx}$ can be made decidable for any decidable class. An interesting relevant question is the following.

Open problem 5 What is the complexity of Formula Instantiation for known decidable classes of predicate logic with/out equality? What is the complexity of $HI(\mathfrak{S})$ (or $HI(\mathfrak{S})^{\approx}$), where $\mathfrak{S}$ is the standard strategy, for these classes?

We can ask similar questions for other natural strategies. If we get some complexity results, it would be quite interesting to compare them with the complexity of known resolution-based decision procedures [34, 25, 36]. Some comparison of resolution and tableau-like methods was done in [8, 23, 24].

The aim of Problem 5 is to compare resolution and tableau-like methods in the cases when they can be compared. However, the ultimate goal of automated reasoning is to find "more efficient" semi-decision procedures for predicate logic as a whole. In order to understand the inherent efficiency (or inefficiency) of tableau-like methods in general, we should understand what happens when we work with large multiplicities.

For classes of formulas for which Formula Instantiation is undecidable, a relevant question is asked in Problem 4. Consider now theories for which Formula Instantiation is decidable (e.g. logic without equality or the theory of a transitive relation). An example problem is

Open problem 6 What is the complexity of $HI(2^{2^{2^{2^{2^{2^{n|\varphi|}}}}}})$ for logic without equality?

Posing of this problem does not mean that we propose to work with such large multiplicities. The problem is interesting for the following reason. If the complexity is still 6 or 7 exponents in the size of φ and n, then we probably cannot gain of using large multiplicities, which means that the standard loop and strategy (or some modifications thereof) can be used. If the complexity collapses (for example becomes 2 exponents), we should try to identify the reason for such a collapse. After identification, we may reconsider the standard approaches to tableau-like methods.

Several authors proposed to use "universal variables" in tableaux in addition to the rigid ones [4]. Logically speaking, this means that instead of solving Formula Instantiation for quantifier-free formulas, we solve it for more general classes of formulas. Consider an example "matrix" that contains quantified formulas.

$$\begin{bmatrix} \vdots \\ \neg P(x,y) \\ \vdots \\ \exists z P(y,z) \\ \vdots \end{bmatrix}$$

The substitution $[x \mapsto y]$ makes this path contain $\neg P(y,y) \vee \exists z P(y,z)$, which is a tautology. Some tableau-based procedures would identify z as a "universal variable"[2], and instead of solving two equations $x \approx y, y \approx z$ would just solve $x \approx y$. This observation can be formalized as follows.

[2] The term "universal variable" is due to the fact that tableau-like methods are usually represented as refutation methods, in which case z would be universally quantified inside the matrix.

Definition 6.2 (General Formula Instantiation) *General Formula Instantiation* is the following decision problem. Given a formula $\varphi(\mathbf{x})$ (not necessarily quantifier-free), is there a term sequence $\mathbf{t}$ such that $\varphi(\mathbf{t})$ is provable?

Of course General Formula Instantiation is, in general, undecidable even for logic without equality, since the provability problem for first order logic is a special case of General Formula Instantiation.

However, we can show that the complexity of Formula Instantiation does not grow when we allow quantifiers applied to "simple" formulas.

Theorem 6.3 Let Φ be the class of formulas without equality in negation normal form such that in every formula in Φ all occurrences of quantifiers have the form $\exists x_1 \ldots \exists x_n L$, where L is a literal. General Formula Instantiation for Φ is decidable and Σ_2^p-complete.

Proof. Σ_2^p-hardness follows from Theorem 4.4.

To prove inclusion in Σ_2^p, we introduce several notions in the style of [28] and prove several simple lemmas. A formula φ is called *rectified* if all free variables of φ are different from all bound variables of φ and each distinct occurrences of quantifiers in φ bind distinct variables. For example, the formulas $\exists x P(x) \vee \exists x \neg P(x)$ or $P(x) \vee \exists x \neg P(x)$ are not rectified. Without loss of generality, we only consider rectified formulas of Φ.

We call an $\exists$-*literal* any formula of the form $\exists x_1 \ldots \exists x_n L$, where L is a literal. Two $\exists$-literals E_1 and E_2 are called *complementary under a substitution* σ if $\vdash E_1\sigma \vee E_2\sigma$.

By an immediate application of Herbrand's theorem we can prove the following lemma.

Lemma 6.4 Let A, B be atoms, $E_1 = \exists \mathbf{x} A$ and $E_2 = \exists \mathbf{y} \neg B$ be two $\exists$-literals, and Z be the set of free variables of $E_1 \vee E_2$. Let σ be a substitution such that (i) $dom(\sigma) \subseteq Z$ and (ii) $vran(\sigma)$ contains no variables in $\mathbf{x}, \mathbf{y}$. Then E_1, E_2 are complementary under σ if and only if there exists a substitution θ such that $A\theta = B\theta$ and $\sigma = \theta\restriction_Z$. $\Box$

Using this lemma and Herbrand's theorem we can prove the following lemma.

Lemma 6.5 Let $E_1, \ldots, E_n$ be $\exists$-literals and σ be a substitution. Then $\vdash (E_1 \vee \ldots \vee E_n)\sigma$ if and only if for some i, j, the $\exists$-literals E_i and E_j are complementary under σ. $\Box$

Let $\varphi \in \Phi$. The set of *$\exists$-literals of* φ is the smallest set S of $\exists$-literals so that φ is constructed from $\exists$-literals in S using $\vee$ and $\wedge$. For example, if φ is the formula $\neg P(x,y) \vee \exists z \exists u P(y, f(z,u))$, then $\neg P(x,y)$ and $\exists z \exists u P(y, f(z,u))$ are the only $\exists$-literals of φ. Note that the formulas $P(x,y)$ and $\exists u P(y, f(z,u))$ are $\exists$-literals and subformulas of φ, but none of them is an $\exists$-literal of φ.

A *mated pair in* φ is a pair of $\exists$-literals of φ of the form $\exists \mathbf{x} P(\mathbf{t}), \exists \mathbf{y} \neg P(\mathbf{s})$. A *premating in* φ is a set of mated pairs in φ. A *mating in* φ is a pair (M, σ), where M is a premating in φ and σ is a substitution so that every pair in M is complementary under σ.

Lemma 6.6 The following problem can be solved in polynomial time: *given a premating M in φ, does there exist a mating (M, σ) in φ?*

Proof. Suppose that M consists of pairs $(\exists \mathbf{x}_i A_i, \exists \mathbf{y}_i \neg B_i)_{i \in \{1, \dots, n\}}$. By renaming bound variables, we can assume that all variables in $\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}_1, \dots, \mathbf{y}_n$ are pairwise different. Using Lemma 6.5, we can prove that a mating (M, σ) exists if and only if there exists a simultaneous unifier of pairs (A_i, B_i). The existence of such a simultaneous unifier can be checked in polynomial time. $\square$

Lemma 6.7 Let $\psi \in \Phi$ be a formula in conjunctive normal form

$$\psi = \bigwedge_{i \in I} \bigvee_{j \in J_i} E_{i,j}$$

Then ψ is a yes-instance of General Formula Instantiation if and only if there exists a mating (M, σ) in ψ such that for every $i \in I$ there exists $j, k \in J_i$ such that $(E_{i,j}, E_{i,k}) \in M$. $\square$

The proof uses Lemma 6.5 and can be found in [54].

Now we can complete the proof of Theorem 6.3. Let $\varphi \in \Phi$ and $\psi \in \Phi$ be obtained from φ by the standard translation into conjunctive normal form. Then $\vdash \varphi \equiv \psi$, every mating in φ is also a mating in ψ and vice versa. We have $\psi = \bigwedge_{i \in I} \bigvee_{j \in J_i} E_{i,j}$.

By Lemma 6.7 ψ is a yes-instance of General Formula Instantiation if and only if there exists a mating (M, σ) in ψ such that for every $i \in I$ there exists $j, k \in J_i$ such that $(E_{i,j}, E_{i,k}) \in M$.

Using $\vdash \varphi \equiv \psi$ and using that matings in ψ are precisely matings in φ, we have that the following conditions are equivalent: (i) φ is a yes-instance of General Formula Instantiation; (ii) there exists a premating M in φ and a substitution σ such that (M, σ) is a mating and for every $i \in I$ there exists $j, k \in J_i$ such that $(E_{i,j}, E_{i,k}) \in M$.

It remains to note that (ii), considered as a decision problem of the given φ, is in Σ_2^p. Indeed, all prematings can be generated by a nondeterministic polynomial-time algorithm. By Lemma 6.6, for every premating M the existence of σ can be checked in polynomial time. Finally, note that all sets

$$\{E_{i,j} \mid j \in J_i\}_{i \in I}$$

can be generated by a nondeterministic polynomial-time algorithm. $\square$

General Formula Instantiation gives rise to a number of other interesting problems.

Open problem 7 Find classes of formulas and/or theories for which Formula Instantiation is of a "reasonable" complexity. Design procedures for checking Formula Instantiation and returning the resulting substitutions.

One particular classification of General Formula Instantiation may be in terms similar to those of the classical decision problem [11].

Open problem 8 Classify formulas, e.g. depending on a signature, quantifier prefix and the number of free variables, w.r.t. General Formula Instantiation (decidability, complexity, the number of solutions).

In view of Theorem 6.3 and connections between General Formula Instantiation and tableau-based theorem proving, instead of prefix formulas one can consider formulas constructed from prefix ones using $\wedge, \vee$. It is also reasonable to restrict oneself to only existencial quantifiers, since for universally quantified formulas we have no analogue of Herbrand's theorem.

Summary

In this article we undertook a mathematical investigation of problems related to tableau-based theorem proving. We proved/cited relevant results and formulated a number of open problems.

In our opinion, we shall know more about the potentials of tableau-based methods when we obtain more results about the above mentioned problems and try to implement new procedures where the results show possible efficiency.

Acknowledgments

I thank Anatoli Degtyarev for his support and valuable remarks. I also thank Vaughan Pratt for helpful comments to a preliminary version of this paper.

References

[1] P.B. Andrews. Theorem proving via general matings. *Journal of the Association for Computing Machinery*, 28(2):193–214, 1981.

[2] L. Bachmair and H. Ganzinger. Rewrite techniques for transitive relations. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*. IEEE Computer Society Press, 1994.

[3] L. Bachmair, H. Ganzinger, and A. Voronkov. Elimination of equality with ordering constraints. In *CADE'98*, 1998. Submitted.

[4] B. Beckert and R. Hähnle. An improved method for adding equality to free variable semantic tableaux. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 678–692, Saratoga Springs, NY, USA, June 1992. Springer Verlag.

[5] 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.

[6] W. Bibel. A theoretical basis for the systematic proof procedure. In P. Dembinski, editor, *Proc. of 9th Symposium on Mathematical Foundations of Computer Science*, volume 88 of *Lecture Notes in Computer Science*, pages 154–167, Rydzzyna, Poland, 1980.

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

[8] W. Bibel. A comparative study of several proof procedures. *Artificial Intelligence*, 18(3):269–293, May 1982.

[9] W. Bibel. *Automated theorem proving.* Vieweg Verlag, 1987. 2nd edition.

[10] W. Bibel and E. Eder. Methods and calculi for deduction. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, chapter 3, pages 67–182. Oxford University Press, 1993.

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

[12] 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 RTA'98.

[13] A. Degtyarev, Yu. Gurevich, and A. Voronkov. Herbrand's theorem and equational reasoning: Problems and solutions. In *Bulletin of the European Association for Theoretical Computer Science*, volume 60, pages 78–95. October 1996. The "Logic in Computer Science" column.

[14] 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.

[15] A. Degtyarev and A. Voronkov. Equality elimination for the inverse method and extension procedures. In C.S. Mellish, editor, *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, pages 342–347, Montréal, August 1995.

[16] 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.

[17] A. Degtyarev and A. Voronkov. Equality elimination for the tableau method. In J. Calmet and C. Limongelli, editors, *Design and Implementation of Symbolic Computation Systems. International Symposium, DISCO'96*, volume 1128 of *Lecture Notes in Computer Science*, pages 46–60, Karlsruhe, Germany, September 1996.

[18] A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. In H. Kleine Büning, editor, *Computer Science Logic. 9th International Workshop, CSL'95*, volume 1092 of *Lecture Notes in Computer Science*, pages 178–190, Paderborn, Germany, September 1995, 1996.

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

[20] A. Degtyarev and A. Voronkov. What you always wanted to know about rigid E-unification. *Journal of Automated Reasoning*, 20(1):47–80, 1998.

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

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

[23] E. Eder. A comparison of the resolution calculus and the connection method, and a new calculus generalizing both methods. In E. Börger, G. Jäger, H. Kleine Büning, and M.M. Richter, editors, *CSL'88 (Proc. 2nd Workshop on Computer Science Logic)*, volume 385 of *Lecture Notes in Computer Science*, pages 80–98. Springer Verlag, 1988.

[24] E. Eder. Consolution and its relation with resolution. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 132–136, 1991.

[25] C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Methods for the Decision Problem*, volume 679 of *Lecture Notes in Computer Science*. Springer Verlag, 1993.

[26] C.F. Fermüller and A. Leitsch. Decision procedures and model building in equational clause logic. *Logic Journal of the IGPL*, 6(1):17–41, January 1998.

[27] M. Fitting. *First Order Logic and Automated Theorem Proving*. Springer Verlag, New York, 1996. Second edition.

[28] J. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Harper and Row, New York, 1986.

[29] 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.

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

[31] 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.

[32] 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.

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

[34] W. Joyner jr. Resolution strategies as decision procedures. *Journal of the Association for Computing Machinery*, 23:398–417, 1976.

[35] A. Kościelski and L. Pacholski. Complexity of Makanin's algorithm. *Journal of the Association for Computing Machinery*, 43(4):670–684, 1996.

[36] A. Leitsch. *The Resolution Calculus*. Springer Verlag, 1997.

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

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

[39] D.W. Loveland. Mechanical theorem proving by model elimination. *Journal of the Association for Computing Machinery*, 15:236–251, 1968.

[40] D.W. Loveland. *Automated Theorem Proving: a Logical Basis*. North Holland, 1978.

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

[42] S.Yu. Maslov, G.E. Mints, and V.P. Orevkov. Mechanical proof-search and the theory of logical deduction in the USSR. In J.Siekmann and G.Wrightson, editors, *Automation of Reasoning (Classical papers on Computational Logic)*, volume 1, pages 29–38. Springer Verlag, 1983.

[43] M. Moser, C. Lynch, and J. Steinbach. Model elimination with basic ordered paramodulation. Technical Report AR-95-11, Fakultät für Informatik, Technische Universität München, München, 1995.

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

[45] U. Petermann. A framework for integrating equality reasoning into the extension procedure. In D. Basin, R. Hähnle, B. Fronhöfer, J. Posegga, and C. Schwind, editors, *Proc. Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 195–207, 1993.

[46] 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.

[47] D. Prawitz. A proof procedure with matrix reduction. In M. Laudet, L. Lacombe, L. Nolin, and M Schutzenberger, editors, *Symp. on Automatic Demonstration*, volume 125 of *Lecture Notes in Mathematics*, pages 207–213. Springer Verlag, 1970.

[48] 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.

[49] M. Stickel. Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1:333–355, 1985.

[50] 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.

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

[52] A. Voronkov. The ground-negative fragment of first-order logic is Π_2^p-complete. UPMAIL Technical Report 144, Uppsala University, Computing Science Department, May 1997.

[53] 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.

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

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