Strategies in Rigid-Variable Methods

Andrei Voronkov*

Abstract

We study complexity of methods using rigid variables, like the method of matings or the tableau method, on a decidable class of predicate calculus with equality. We show some intrinsic complications introduced by rigid variables. We also consider strategies for increasing multiplicity in rigid-variable methods, and formally show that the use of intelligent strategies can result in an essential gain in efficiency.

1 Introduction

Automated reasoning methods for first-order classical logic can generally be divided in two classes.

Methods of the first class use *universal variables* (resolution [Robinson, 1965], the inverse method [Maslov, 1983]). Variables in these methods are local to a clause (formula, sequent) and can be considered as universally quantified in this clause (respectively formula or sequent). [Maslov *et al.*, 1983; Maslov, 1987] characterized these methods as *local methods* (see also [Mints, 1990]).

Methods of the second class use *rigid variables* (the tableau method [Beth, 1959], the mating or the connection method [Andrews, 1981; Bibel, 1981], model elimination [Loveland, 1968], SLD-resolution [Kowalski, 1974; Apt and van Emden, 1982], SLO-resolution [Demolombe, 1989; Rajasekar, 1989]). Variables in these methods are local to a set of clauses (formulas, sequents) and can be considered as universally quantified in this set of clauses (respectively formulas or sequents). [Maslov *et al.*, 1983; Maslov, 1987] characterized these methods as *global methods*. In this paper, we shall call such methods *rigid-variable methods*.

Both kinds of methods have their advantages and disadvantages which are well-known. There are papers comparing resolution and tableau-like calculi, for example [Eder, 1988; 1991] (see also [Bibel and Eder, 1993]). Recently, there have been proposals combining both kinds of methods in one calculus, for example the equality elimination method [Degtyarev and Voronkov, 1995b; 1995a; 1996a] or a modification of model elimination [Moser *et al.*, 1995].

Although there are many implementations of rigid-variable methods, there are almost no papers investigating the intrinsic complexity of problems arising in rigid-variable methods. In this paper we study several complexity problems related to rigid-variable methods on a decidable fragment of predicate calculus with equality. We show that the use of such methods can introduce essential complications even for this relatively simple fragment. We also demonstrate that methods with rigid variables can gain from the intelligent use of *strategies for multiplicity*. In addition, we show that a recent result of [Voda and Komara, 1995] on the Herbrand skeleton problem is related to an inadequate formulation of the problem, and pose a new open problem.

2 Preliminaries

A term or atomic formula is *ground* iff it has no variables. The symbol $\vdash$ denotes provability in first-order logic. When we write $\varphi_1, \ldots, \varphi_n \vdash \varphi$, where $\varphi_1, \ldots, \varphi_n, \varphi$ are formulas, it means provability of the formula $\varphi_1 \wedge \ldots \wedge \varphi_n \supset \varphi$. *Substitutions* of terms $t_1, \ldots, t_n$ for variables $x_1, \ldots, x_n$ are denoted $\{t_1/x_1, \ldots, t_n/x_n\}$. The *application of such a substitution* θ *to a term* t, is the operation of simultaneous replacement of all occurrences of x_i by t_i. The result of the application is the term denoted $t\theta$. We shall also apply substitutions to formulas, equations and sets of equations and use the same notation for the result of the application.

We shall consider first-order predicate logic with equality. The equality predicate is denoted by $\simeq$, in order to distinguish it from the metasymbol $=$ used to denote the identity of two expressions. The symbol $\rightleftharpoons$ means "equal by definition". Atomic formulas $s \simeq t$ are

*Computing Science Department, Uppsala University, Box 311, S-751 05 Uppsala, Sweden. Supported by a TFR grant. URL page: http://www.csd.uu.se/~voronkov. Email: voronkov@csd.uu.se.

called *equations,* and their negations $\neg s \simeq t$, denoted $s \not\simeq t$, *disequations.* We do not distinguish an equation $s \simeq t$ from the equation $t \simeq s$. For a technical convenience, we can also use equations and disequations between atomic formulas, for example $P(s_1, \ldots, s_m) \simeq Q(t_1, \ldots, t_n)$.

Let $s_1, t_1, \ldots, s_n, t_n, s, t$ be terms. We write $s_1 \simeq t_1, \ldots, s_n \simeq t_n \vdash s \simeq t$ to denote that the formula $\forall(s_1 \simeq t_1 \wedge \ldots \wedge s_n \simeq t_n \supset s \simeq t)$ is true, i.e. it is provable in first-order logic. Equivalently, we can say that s and t lie in the same class of the congruence generated by $\{s_1 \simeq t_1, \ldots, s_n \simeq t_n\}$.

A *rigid equation* is an expression $\mathcal{E} \vdash_\forall s \simeq t$, where $\mathcal{E}$ is a finite set of equations. $\mathcal{E}$ is called the *left hand side* of this rigid equation, and the equation $s \simeq t$ — its *right hand side.* A *solution to a rigid equation* $\{s_1 \simeq t_1, \ldots, s_n \simeq t_n\} \vdash_\forall s \simeq t$ is any substitution σ such that $s_1\sigma \simeq t_1\sigma, \ldots, s_n\sigma \simeq t_n\sigma \vdash s\sigma \simeq t\sigma$. A *system of rigid equations* is a finite set of rigid equations. A *solution to a system of rigid equations* $\mathcal{R}$ is any substitution that is a solution to every rigid equation in $\mathcal{R}$. The problem of solvability of rigid equations is known as *rigid E-unification.* The problem of solvability of systems of rigid equations is known as *simultaneous rigid E-unification,* or SREU for short.

We shall denote sets of equations by $\mathcal{E}$, systems of rigid equations by $\mathcal{R}$ and rigid equations by R. We shall sometimes write the left hand side of a rigid equation as a *sequence* of equations, for example $x \simeq a \vdash_\forall g(x) \simeq x$ instead of $\{x \simeq a\} \vdash_\forall g(x) \simeq x$.

For simplicity, in this paper we consider the provability problem for closed prenex existential formulas, i.e. formulas of the form $\exists \bar{x} \varphi(\bar{x})$, where $\varphi(\bar{x})$ is a quantifier-free formula. There is a provability-preserving polynomial time translation of arbitrary formulas to closed prenex existential formulas by means of Skolemization and prenexing.

According to the Herbrand theorem, such a formula is provable if and only if there exists a positive integer μ and a substitution θ such that the formula $(\varphi(\bar{x}_1) \vee \ldots \vee \varphi(\bar{x}_\mu))\theta$ is provable. This fact is used in several automated reasoning methods, for example in the method of matings, in the way shown in Figure 1. The procedure shown in that figure will simply be called *the Procedure.*

1. Choose a positive integer μ.
2. Construct the formula $\varphi_\mu = (\varphi(\bar{x}_1) \vee \ldots \vee \varphi(\bar{x}_\mu))$ and check whether there is a substitution θ such that the formula $\varphi_\mu\theta$ is provable.
3. If such a substitution exists, then the goal formula $\exists \bar{x} \varphi(\bar{x})$ is provable. Otherwise, increase μ and come back to step (2).

Figure 1: The Procedure: a typical procedure for rigid-variable methods

In the method of matings, before step (2) the formula φ_μ is represented in the form of a *matrix* M and the provability of $\varphi_\mu\theta$ means that any vertical path in $M\theta$ is inconsistent. The tableau method represents the formula in the form of a tree and uses branches of the tree instead of vertical paths.

The number μ used in the Procedure (the number of copies of $\varphi(\bar{x})$ which can be used) is usually called *multiplicity.* Of course, there are various modifications of the Procedure, for example, the goal formula may be non-prenex. In this case the notion of multiplicity is more complicated. Our results can also be generalized to more complex notions of multiplicity. However, we shall only consider prenex existential formulas, for which the notion of multiplicity is defined as a positive integer number.

We informally call a *strategy for multiplicity* any procedure which selects the initial multiplicity and increases multiplicity in the Procedure. The *standard strategy for multiplicity* is the strategy which sets μ initially to 1 and increments it by 1 on any further step. A strategy for multiplicity is called *formula-independent* iff it does not depend on the input formula.

There are various algorithms for checking, for a given formula φ_μ, whether there is a substitution making this formula provable. Instead of studying concrete procedures, we shall study the intrinsic complexity of the problem which can be formulated as follows.

Problem 1 (Herbrand Skeleton) *Given a quantifier-free formula $\varphi(\bar{x})$ and a positive integer μ, are there term sequences $\bar{t}_1, \ldots, \bar{t}_n$ such that the formula $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_\mu)$ is provable?*

[Degtyarev *et al.*, 1996a] give an informal survey of several decision problems arising from the Herbrand theorem, including the Herbrand Skeleton problem. It is clear that Problem 1 is decidable if and only if the following problem is decidable.

Problem 2 (Formula Instantiation) *Given a quantifier-free formula $\varphi(\bar{x})$, is there a term sequence $\bar{t}$ such that the formula $\varphi(\bar{t})$ is provable?*

Note that the formula instantiation problem is repeatedly used in the procedures used by the method of matings or the tableau method.

The decidability of these problems is equivalent to the decidability of SREU. Unfortunately, it turned out that SREU has almost no decidable fragments which are general enough. Some known results on SREU are the following.

- SREU is undecidable [Degtyarev and Voronkov, 1996b].
- SREU with ground left hand sides is undecidable [Plaisted, 1995].
- SREU with ground left hand sides and two variables is undecidable [Veanes, 1996].
- SREU with one variable is DEXPTIME-complete [Degtyarev *et al.*, 1997].

The case of one variable is hardly useful in automated reasoning. When all function symbols have arity ≤ 1, Formula Instantiation is equivalent to *monadic SREU,* i.e. SREU in the signature where all function symbols have arity ≤ 1. The decidability of monadic SREU is an open problem. The following facts are known about monadic SREU.

- Monadic simultaneous rigid E-unification with one function symbol is decidable (this fact has a rather non-trivial proof in [Degtyarev *et al.*, 1996b]).
- Monadic simultaneous rigid E-unification with more than one function symbol is equivalent to a non-trivial extension of word equations [Gurevich and Voronkov, 1997a].

Some other decidable fragments of monadic SREU are considered in [Gurevich and Voronkov, 1997a], but they are hardly of much use for automated reasoning.

Since predicate calculus is undecidable, the undecidability of SREU does not add much to the complexity of predicate calculus. In this paper we consider the behavior of methods based on rigid variables on a decidable fragment of predicate calculus.

3 Ground-negative fragment of predicate calculus

A formula φ is called *positive* if all atomic subformulas of φ are positive. A closed formula φ is *ground-negative* iff any occurrence of a variable in φ is either an occurrence in a positive atomic subformula of φ or is bound by an essentially universal quantifier (i.e. an universal quantifier occurring in φ positively or an existential quantifier occurring in φ negatively)[1]. [Kozen, 1977] proves the following result.

Theorem 1 *The class of provable formulas of the form $A_1 \wedge \ldots \wedge A_n \supset \varphi$, where $A_1, \ldots, A_n$ are ground atomic formulas and φ is a positive formula, is NP-complete.*

Using this result, one can prove the following:

[1]Variables occurring in essentially universal quantifiers can also be characterized as eigenvariables. Thus, a formula φ is ground-negative if and only if every variable occurring in a negative atom in φ is an eigenvariable.

Theorem 2 *The provability problem for ground-negative formulas is in $\mathbf{\Pi}_2^p$.*

Since we only consider prenex existential formulas, such a formula $\exists\bar{x}\varphi(\bar{x})$ is ground-negative if and only if all negative atomic subformulas of $\varphi(\bar{x})$ are ground. So we consider the Herbrand skeleton problem and the formula instantiation problem for ground-negative formulas. We assume that $\exists\bar{x}\varphi(\bar{x})$ is a fixed ground-negative existential prenex formula.

Theorem 3 *The formula instantiation problem for ground-negative formulas is undecidable.*

Proof. We shall use the result proved in [Plaisted, 1995] that SREU with ground left hand sides is undecidable. Consider any system $\mathcal{R}$ of rigid equations with ground left hand sides:

$$\begin{array}{lll} s_{11} \simeq t_{11}, \ldots, s_{1n_1} \simeq t_{1n_1} & \vdash_\forall & s_1 \simeq t_1 \\ & \cdots & \\ s_{m1} \simeq t_{m1}, \ldots, s_{mn_m} \simeq t_{mn_m} & \vdash_\forall & s_m \simeq t_m \end{array}$$

Consider also the following formula φ:

$$\begin{array}{llll} (s_{11} \simeq t_{11} \wedge \ldots \wedge s_{1n_1} \simeq t_{1n_1} & \supset & s_1 \simeq t_1) & \wedge \\ & \cdots & & \wedge \\ (s_{m1} \simeq t_{m1} \wedge \ldots \wedge s_{mn_m} \simeq t_{mn_m} & \supset & s_m \simeq t_m) & \end{array}$$

It is straightforward that a substitution θ is a solution to $\mathcal{R}$ if and only if the formula $\varphi\theta$ is provable. Note that all negative atoms in $\varphi\theta$ are ground. Hence, SREU with ground left hand sides is effectively reducible to the formula instantiation problem for ground-negative formulas. Thus, the formula instantiation problem for ground-negative formulas is undecidable. □

If we consider which systems of rigid equations arise from ground-negative formulas (e.g. according to the procedures of [Gallier *et al.*, 1990; 1992]), we shall find out that these are precisely all systems of rigid equations with ground left hand sides.

Theorem 3 shows that a straightforward use of rigid-variable methods can create unnecessary complications, for example, the necessity to solve an intermediate undecidable subproblem in order to solve a problem in $\mathbf{\Pi}_2^p$. This theorem can be reformulated as a statement about the standard strategy for multiplicity:

Theorem 4 *For the standard strategy for multiplicity, a subproblem arising at step (2) of the Procedure is undecidable for the class of ground-negative formulas.*

Proof. Indeed, the subproblem with $\mu = 1$ arising at the first iteration of the algorithm is equivalent to the formula instantiation problem which is undecidable by Theorem 3. □

Hence, the use of the standard strategy may introduce unnecessary complications into rigid-variable methods. We can prove that the same holds for arbitrary formula-independent strategies. To this end, we shall use a result proven in [Veanes, 1997]. First, we cite a result by [Voda and Komara, 1995] which generalizes the undecidability of SREU.

We call a *specialization* of the Herbrand skeleton problem for any fixed μ the following problem[2]:

> Given a quantifier-free formula $\varphi(\bar{x})$, are there term sequences $\bar{t}_1, \ldots, \bar{t}_\mu$ such that the formula $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_\mu)$ is provable?

The following result is proved in [Voda and Komara, 1995]:

Theorem 5 *The specialization of the Herbrand skeleton problem for any fixed μ is undecidable.*

This result has recently been improved in [Veanes, 1997], where they have shown that it also holds for ground-negative formulas:

Theorem 6 *The specialization of the Herbrand skeleton problem for ground-negative formulas and any fixed multiplicity μ is undecidable.*

Theorem 4 about the standard strategy for multiplicity can be generalized as follows.

Theorem 7 *For any formula-independent strategy, a subproblem arising at step (2) of the Procedure for the class of ground-negative formulas is undecidable.*

Proof. Since the strategy is formula-independent, some μ independent of the input formula will be selected at the first iteration of step (2). Then continue similar to the proof of Theorem 4 but using Theorem 6 instead of Plaisted's result. $\square$

We shall consider an intelligent strategy for multiplicity in Section 4.

4 The Herbrand Skeleton Problem and intelligent strategies for multiplicity

Intelligent strategies for multiplicity have always been considered of paramount importance for rigid-variable methods. However, essentially no formal results are known about such strategies. Existing systems based on rigid-variable methods use some heuristic methods for incrementing multiplicity and universal variables whenever possible (see e.g. [Hähnle *et al.*, 1994]). In this section we show that there is an efficient formula-dependent strategy for multiplicity for the class of ground-negative formulas.

[2]The Herbrand skeleton problem described in [Voda and Komara, 1995] is precisely this specialization.

In this section $\psi = \exists\bar{x}\varphi(\bar{x})$ will denote a fixed ground-negative existential prenex formula. Let $A_1, \ldots, A_n$ be all its negative atomic subformulas. Denote by N the set of all subsets of $\{1, \ldots, n\}$. Introduce 2^n sets of ground atoms $\mathcal{E}_I \rightleftharpoons \{A_i \mid i \in I\}$, for every $I \in N$. Using transformations similar to those used in the conjunctive normal form translation, we can assume that $\varphi(\bar{x})$ has the form

$$\bigwedge_{I \in N} ((\bigwedge_{i \in I} A_i) \supset \varphi_I),$$

where φ_I are formulas constructed from atomic formulas using $\wedge, \vee$. Without loss of generality we can assume that for every I, J, if $J \subseteq I$, then $\vdash \varphi_I \supset \varphi_J$. Indeed, if this is not true, we can replace φ_I by $\varphi_I \wedge \varphi_J$, then ψ will be replaced by an equivalent formula.

The notion of the least Herbrand model of a set of formulas is standard and can be found in e.g. [Lloyd, 1987] or [Apt, 1990].

Lemma 4.1 *Let $\mathfrak{A}$ be a set of ground atomic formulas and φ be a closed formula constructed from atomic formulas using only $\wedge$, $\vee$ and $\exists$. Let $\mathfrak{M}$ be the least Herbrand model of $\mathfrak{A}$. If $\mathfrak{M} \models \varphi$, then $\mathfrak{A} \vdash \varphi$.*

Proof. Straightforward, by induction on φ. $\square$

A more general variant of Lemma 4.1 also holds, where $\mathfrak{A}$ is a set of universal quasy-identities (see [Makowski, 1986] or [Sheperdson, 1988]).

Lemma 4.2 *Let $\vdash \psi$. Then for every $I \in N$ there is a substitution θ such that $\mathcal{E}_I \vdash \varphi_I\theta$.*

Proof. Denote by $\mathfrak{M}_\mathfrak{I}$ the least Herbrand model of $\mathcal{E}_I$. We have $\mathfrak{M}_\mathfrak{I} \models \psi$. Since $\mathfrak{M}_\mathfrak{I}$ is a Herbrand model, we have $\mathfrak{M}_\mathfrak{I} \models \varphi(\mathfrak{x})\theta$ for some substitution θ making all variables in $\bar{x}$ ground. We prove that θ satisfies the claim.

Indeed, we have $\mathfrak{M}_\mathfrak{I} \models (\bigwedge_{\mathfrak{i} \in \mathfrak{I}} \mathfrak{A}_\mathfrak{i}) \supset \varphi_\mathfrak{I}\theta$. Since $\mathfrak{M}_\mathfrak{I} \models (\bigwedge_{\mathfrak{i} \in \mathfrak{I}} \mathfrak{A}_\mathfrak{i})$, we have $\mathfrak{M}_\mathfrak{I} \models \varphi_\mathfrak{I}\theta$. By Lemma 4.1 we have $\mathcal{E}_I \vdash \varphi_I\theta$. $\square$

For every $I \in N$, we denote by θ_I some substitution satisfying Lemma 4.2.

Theorem 8 *Let $\vdash \psi$. Then $\vdash \bigvee_{I \in N} \varphi(\bar{x})\theta_I$.*

Proof. Let $\mathfrak{M}$ be an arbitrary model. Consider $I \rightleftharpoons \{i \mid \mathfrak{M} \models \mathfrak{A}_\mathfrak{i}\}$. Since $\mathfrak{M} \models \bigwedge_{\mathfrak{i} \in \mathfrak{I}} \mathfrak{A}_\mathfrak{i}$, by Lemma 4.2 we have $\mathfrak{M} \models \varphi_\mathfrak{I}\theta_\mathfrak{I}$. Consider an arbitrary $J \in N$. Let us prove that $\mathfrak{M} \models (\bigwedge_{\mathfrak{i} \in \mathfrak{J}} \mathfrak{A}_\mathfrak{i}) \supset \varphi_\mathfrak{J}\theta_\mathfrak{I}$. Consider two cases:

1. $J \subseteq I$. Then we have $\vdash \varphi_I \supset \varphi_J$. It follows that $\vdash \varphi_I\theta_I \supset \varphi_J\theta_I$. Hence, $\mathfrak{M} \models \varphi_\mathfrak{J}\theta_\mathfrak{I}$.
2. $J \not\subseteq I$. By the choice of $\mathfrak{M}$ we have $\mathfrak{M} \not\models \bigwedge_{\mathfrak{i} \in \mathfrak{J}} \mathfrak{A}_\mathfrak{i}$.

In both cases we have $\mathfrak{M} \models (\bigwedge_{\mathfrak{i}\in\mathfrak{J}} \mathfrak{A}_\mathfrak{i}) \supset \varphi_\mathfrak{J}\theta_\mathfrak{J}$. Since J was arbitrary, we have

$$\bigwedge_{J\in N} ((\bigwedge_{i\in J} A_i) \supset \varphi_J)\theta_I,$$

i.e. $\mathfrak{M} \models \varphi(\mathfrak{x})\theta_\mathfrak{J}$.

Then we have $\mathfrak{M} \models \bigvee_{\mathfrak{J}\in\mathfrak{N}} \varphi(\mathfrak{x})\theta_\mathfrak{J}$. Since $\mathfrak{M}$ is arbitrary, we have $\vdash \bigvee_{I\in N} \varphi(\bar{x})\theta_I$. □

Theorem 9 *The following problem is in $\mathbf{\Pi}_2^p$ (compare it with the Herbrand Skeleton problem).*

Given any ground-negative formula $\exists\bar{x}\varphi(\bar{x})$ and any positive integer $\mu \geq 2^n$, where n is the number of negative atomic subformulas of $\varphi(\bar{x})$, are there term sequences $\bar{t}_1, \ldots, \bar{t}_\mu$ such that the formula $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_\mu)$ is provable?

Proof. By Theorem 8, if the formula $\exists\bar{x}\varphi(\bar{x})$ is provable, then such term sequences $\bar{t}_1, \ldots, \bar{t}_\mu$ exists. Obviously, the converse is also true. Hence, the problem defined in the theorem is polynomial-time equivalent to the provability problem for ground-negative formulas. By Theorem 2, the provability problem for such formulas is in $\mathbf{\Pi}_2^p$. □

Thus, for ground-negative formulas we have an interesting phenomenon. For small values of multiplicity μ the Procedure should solve an undecidable subproblem at step (2); for large enough values of μ, this subproblem is in $\mathbf{\Pi}_2^p$. Thus, we have formally shown that for the class of ground-negative formulas, formula-dependent strategies for multiplicity can result in a huge gain in efficiency.

Theorem 9 also shows that the result of Voda and Komara is not related to formula-dependent strategies for multiplicity. In order to formally define a formula-dependent strategy, we can represent the value of μ on the nth iteration of the Procedure as a function f of two arguments: the input formula $\exists\bar{x}\varphi(x)$ and the number n. Since μ should be increased with each next iteration, we have $f(\exists\bar{x}\varphi(x), n+1) > f(\exists\bar{x}\varphi(x), n)$.

Then we can formally define a *strategy for multiplicity* as any function f such that

1. The first argument of f ranges over prenex existential formulas $\exists\bar{x}\varphi(x)$;
2. The second argument of f ranges over positive integers;
3. for every positive integers $k > m$ and every prenex existential formula ψ we have $f(\psi, k) > f(\psi, m)$.

The following problem arises:

Problem 3 *Is there a strategy for multiplicity f such that*

1. *f is computable;*
2. *The following problem is decidable. Given a number k and a prenex existential formula $\exists\bar{x}\varphi(x)$, are there term sequences $\bar{t}_1, \ldots, \bar{t}_{f(\exists\bar{x}\varphi(x),k)}$ such that the formula $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_{f(\exists\bar{x}\varphi(x),k)})$ is provable?*

This problem is still open. We conjecture that such function does not exist, but the proof of this fact would require some non-trivial diagonalization.

Even if this problem has a negative solution, there are still at least two known ways for the use of rigid-variable methods for logic with equality. One way is to augment rigid-variable methods by universal-variable parts, as for example in [Degtyarev and Voronkov, 1995b; 1996a] or [Moser *et al.*, 1995]. Another way is to use incomplete but terminating algorithms on step (2) of the Procedure, as demonstrated in [Plaisted, 1995] or [Degtyarev and Voronkov, 1996c].

Acknowledgments

I thank Anatoli Degtyarev, Yuri Gurevich and Margus Veanes. Some ideas of this paper appeared as a result of our discussions. I thank an anonymous referee who pointed out serious inaccuracies in the preliminary version of this paper.

References

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

[Apt, 1990] K.R. Apt. Logic programming. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 10, pages 493–574. Elsevier Science, Amsterdam, 1990.

[Apt and van Emden, 1982] K. Apt and M. van Emden. Contributions to the theory of logic programming. *Journal of the Association for Computing Machinery*, 29(3), 1982.

[Beth, 1959] E.W. Beth. *The Foundations of Mathematics.* North Holland, 1959.

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

[Bibel and Eder, 1993] 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.

[Degtyarev and Voronkov, 1995a] 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.

[Degtyarev and Voronkov, 1995b] A. Degtyarev and A. Voronkov. General connections via equality elimination. In M. De Glas and Z. Pawlak, editors, *Second World Conference on the Fundamentals of Artificial Intelligence (WOCFAI-95)*, pages 109–120, Paris, July 1995. Angkor.

[Degtyarev and Voronkov, 1996a] 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.

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

[Degtyarev and Voronkov, 1996c] A. Degtyarev and A. Voronkov. What you always wanted to know about rigid E-unification. In J.J. Alferes, L.M. Pereira, and E. Orlowska, editors, *Logics in Artificial Intelligence. European Workshop, JELIA'96*, volume 1126 of *Lecture Notes in Artificial Intelligence*, pages 50–69, Évora, Portugal, September/October 1996.

[Degtyarev *et al.*, 1996a] 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, page ??? October 1996. The "Logic in Computer Science" column.

[Degtyarev *et al.*, 1996b] 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.

[Degtyarev *et al.*, 1997] 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.

[Demolombe, 1989] R. Demolombe. An efficient strategy for non-Horn deductive databases. In G.X. Ritter, editor, *Information Processing 89*, pages 325–330. Elsevier Science, 1989.

[Eder, 1988] 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.

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

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

[Gallier *et al.*, 1992] J. Gallier, P. Narendran, S. Raatz, and W. Snyder. Theorem proving using equational matings and rigid E-unification. *Journal of the Association for Computing Machinery*, 39(2):377–429, 1992.

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

[Gurevich and Voronkov, 1997a] Yu. Gurevich and A. Voronkov. Decision problems for logic with unary function symbols. In *ICALP'97*, page 12, 1997. To appear.

[Gurevich and Voronkov, 1997b] Yu. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. UPMAIL Technical Report 134, Uppsala University, Computing Science Department, January 1997.

[Hähnle *et al.*, 1994] R. Hähnle, B. Beckert, and S. Gerberding. The many-valued tableau-based theorem prover ${}_3T^AP$. Technical Report 30/94, Universität Karlsruhe, Fakultät für Informatik, November 1994.

[Kowalski, 1974] R.A. Kowalski. Predicate logic as a programming language. In *Proc. IFIP'74*, pages 569–574. North Holland, 1974.

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

[Lloyd, 1987] J.W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer Verlag, 1987.

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

[Makowski, 1986] J.A. Makowski. Why Horn formulas matter in computer science: Initial structures and generic examples. In H. Ehrig et al., editor, *TAPSOFT'86*, volume 185 of *Lecture Notes in Computer Science*, pages 374–387. Springer Verlag, 1986.

[Maslov, 1983] S.Yu. Maslov. An inverse method for establishing deducibility of nonprenex formulas of the predicate calculus. In J.Siekmann and G.Wrightson, editors, *Automation of Reasoning (Classical papers on Computational Logic)*, volume 2, pages 48–54. Springer Verlag, 1983.

[Maslov, 1987] S.Yu. Maslov. *Theory of Deductive Systems and its Applications.* MIT Press, 1987.

[Maslov *et al.*, 1983] S.Y. 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.

[Mints, 1990] G. Mints. Gentzen-type systems and resolution rules. Part I. Propositional logic. In P. Martin-Löf and G. Mints, editors, *COLOG-88*, volume 417 of *Lecture Notes in Computer Science*, pages 198–231. Springer Verlag, 1990.

[Moser *et al.*, 1995] 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.

[Plaisted, 1995] 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.

[Rajasekar, 1989] A. Rajasekar. *Semantics for Disjunctive Logic Programs.* PhD thesis, University of Maryland, 1989.

[Robinson, 1965] J.A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, 12(1):23–41, 1965.

[Sheperdson, 1988] J.C. Sheperdson. Negation in logic programming. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 19–88. Morgan Kaufmann, 1988.

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

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