UPMAIL Technical Report No. 92
December 29, 1994

Equality Elimination for the Inverse Method and Extension Procedures

Anatoli Degtyarev[*] [†]
Andrei Voronkov[‡]

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

email {anatoli,voronkov}@csd.uu.se

[*]Supported by a grant from the Swedish Institute
[†]On leave from Kiev University
[‡]Supported by a TFR grant

Abstract

We demonstrate how to handle equality in the inverse method using *equality elimination.* In the equality elimination method, proofs consist of two parts. In the first part we try to solve equations obtaining so called *solution clauses.* In the second part, we perform the usual sequent proof search by the inverse method. Our method is called equality elimination because we eliminate all occurrences of equality in the first part of the proof. Solution clauses are obtained by using a very strong strategy — basic superposition. Unlike the previous approach proposed by Maslov, we prove completeness of our method with most general substitutions and with ordering restrictions.

We also note that these technique is directly applicable to *extension procedures,* like the *connection method.* Unlike other approaches, we do not require the use of rigid or mixed E-unification.

Section 1

The inverse method

The inverse method of theorem proving in sequent calculi has been proposed by Maslov in the 1960s. The method is based on the bottom-up search in sequent calculi. The inverse method is completely local [MM 83, DV 94b] and can be efficiently implemented both for classical and non-classical logics [Vor 92].

The inverse method can be characterized as a bottom-up search in sequent calculi. In terms of efficiency, it is competitive with resolution. The inverse method requires no normal forms which can be an advantage for interactive provers. It is also applicable to non-classical logics.

The introduction of equality in the inverse method has not yet received proper attention. In this paper we consider the inverse method with equality based on equality elimination.

The structure of this paper is the following. In this section we briefly discuss introduction of equality in the inverse method in general. In Section 2 we introduce main definitions and notation. In Section 3 we define the equality elimination method and give several examples. The equality elimination method for Horn clauses with equality and for semantic tableaux has been considered in [DV 94a, DV 94b]. In Section 4 we show how one can generalize our technique to the connection method. Appendix A contains proofs of main theorems.

The first natural generalization of the inverse method to include equality has been made by Maslov [Mas 71]. It was based on theorems proved by Kanger [Kan 63] and Lifschitz [Lif 68] about the specialization of proofs in the sequent calculus with equality (but without free variables!). According to these theorems, all equality reasoning steps can be moved on top of the proof so that they precede all other steps. This gives the characterization of the inverse method with equality which in fact coincides with hyperresolution (on classical logic), but with a different initial set of clauses (closed collections in Maslov's terminology). While in the inverse method without equality the initial sequents correspond to axioms of the sequent calculus, in the equality case initial sequents are those derivable from axioms exclusively by equality rules. Kanger [Kan 63] called such sequents "directly demonstratable". In later papers the role of these sequents is played by (instantiated) equational mated sets [Gal 92] or $\mathcal{E}$-complementary eq-connections [Bib 87, Pet 93]. We call their analog *solution clauses*.

Let us come back to results on the specialization of sequent proofs. Is it possible to generalize these results on proofs with free variables (metavariables in Maslov's terminology)? In this case equality rules become paramodulation rules. Direct lifting from the ground case is now impossible without functional reflexivity axioms. As we shall show, the use of free variables is possible both for the traditional formulation of the inverse method and for the formulation proposed in our paper.

In the original presentation of the inverse method by Maslov, unification has been made by using arbitrary (not only most general) substitutions which made lifting from the ground case unnecessary[1]. Even in the paper [MM 83] evaluated in [Lif 89] as a "very clean explanation of the inverse method" it was only noted that "likewise the definition of a most general unifier, at each step one could select minimal substitutions". However the proof of completeness in the [MM 83] reformulation of the inverse method is made using unrestricted substitutions given by an arbitrary saturation of the Herbrand universe. Arbitrary substitutions are also used in [Lif 89] where the most general unifiers are only used for factoring.

An alternative approach would be to take into attention that theorems of [Kan 63] and [Lif 68] on the specialization of sequent proofs also allow one to restrict to simultaneous paramodulations only. "Simultaneous" means that in the application of the equality rules from the conclusion to the premise *all* occurrences of the "into-term" are replaced by the "from-term". As it has been shown in [Ben 90], for simultaneous paramodulation lifting is possible without the functional reflexivity axioms.

We do not, however, use simultaneous paramodulation here. Instead, we use most refined strategies of dealing with equality — ordering strategies and based on them basic superposition. It happens that these severe restrictions on the equality part of the proof is enough to preserve completeness using most general unifiers.

Our method is also directly applicable to so called "extension procedures" [Pra 60], including tableaux methods [Fit 90, DV 94b], mating or connection methods [And 86, Bib 87] and their generalization — consolution [Ede 91]. We shall illustrate such an application of our method in Section 4.

The major difference of our treatment of equality reasoning for connection methods is that instead of checking paths through an extension of the input formula for E-complementarity (i.e. for the existence of a rigid E-unifier) we first generate a set of E-complementary paths using ordinary unification, and then look for an appropriate extension. In [Pet 93] it is noted that the problem of combining paramodulation-based equality handling and extension procedure is non-trivial. What we do is to split equality reasoning and extension procedures into two separate procedures and connect them via solution clauses generated by the equality reasoning part. This allows one to use strong sides of both equality reasoning methods known so far and extension procedures.

[1]In terms of resolution, such a treatment of unification gives *unrestricted resolution* [Llo 87] — a calculus with too high non-determinism.

Section 2

Preliminaries

We present here a brief overview of notions and preliminary definitions necessary for the paper. We assume the basic knowledge of substitutions and unification.

Let Σ be a signature, and X a set of variables. $T(\Sigma, X)$ will denote the set of all terms in the signature Σ with variables from X. The set of all *ground* terms in the signature Σ will be denoted by $T(\Sigma)$

A *literal* is either an atomic formula or a negation of an atomic formula. We shall always write $s \neq t$ for the literal $\neg(s = t)$.

A *clause* is a set $\{L_1, \ldots, L_n\}$ of literals, denoted $L_1, \ldots, L_n$. If L is a literal, C a clause, L, C will denote the clause $\{L\} \bigcup C$. An *equation* is an expression $s = t$, where s, t are terms in $T(\Sigma, X)$. By a *ground expression* (i.e. term, equation, clause etc.) we mean an expression containing no variables. We write $A[s]$ to indicate that an expression A contains s as a subexpression and *ambiguously* denote by $A[t]$ the result of replacing particular occurrences of s in A by t. By $A\sigma$ we denote the result of applying the substitution σ to A.

We shall sometimes denote tuples of equations $s_1 = t_1, \ldots, s_n = t_n$ by $\bar{s} = \bar{t}$. The overbar notation $\bar{t}$ can also be used to denote sequences of terms. Substitutions θ with the domain $x_1, \ldots, x_n$ will be denoted by $[x_1/x_1\theta, \ldots, x_n/x_n\theta]$.

Let $\succ$ be a partial ordering on $T(\Sigma, X)$. It is called a *reduction ordering* iff

1. $\succ$ is well-founded;
2. if $s \succ t$ then $u[s\sigma] \succ u[t\sigma]$, for all terms s, t, u and substitutions σ.

We shall always assume that reduction orderings are total on the set $T(\Sigma)$.

Following [BGLS 92] we distinguish terms occurring in the original formula from terms introduced by substitution by using *closures,* i.e. pairs $C \cdot \sigma$, where C is a clause, σ a substitution. The closure $C \cdot \sigma$ is a representation of the clause $C\sigma$. The application of a substitution ρ to such a closure is defined by $(C \cdot \sigma)\rho = C \cdot \sigma\rho$. The clause C will often be identified with the closure $C \cdot \varepsilon$, where ε is the empty substitution. The set $Var(C \cdot \sigma)$ is defined as the set of all variables of $C\sigma$. Two closures $C_1 \cdot \sigma_1$ and $C_2 \cdot \sigma_2$ have *disjoint variables* iff $Var(C_1 \bigcup C_1\sigma_1) \bigcap Var(C_2 \bigcup C_2\sigma_2) = \emptyset$. An *instance of the closure* $C \cdot \rho$ is any closure $C \cdot \rho\sigma$. The instance is ground iff $Var(C\rho\sigma) = \emptyset$. Two closures are *variants* iff both are instances of each other.

When we use notation $\models$ or $\vdash$, they state for truth and provability *in classical first order logic.*

A formula is in *skolemized negation normal form* iff it is constructed from literals using connectives $\wedge, \vee$ and the quantifier $\exists$. There is a structure-preserving translation of formulas without equivalences into formulas in skolemized negation normal form with the same number of occurrences of atoms.

Section 3

The equality elimination method

For the rest of this section we assume that γ is a closed formula in skolemized negation normal form to be proved (the "goal"). We assume that all different occurrences of quantifiers in γ bind different variables. For example, γ cannot have the form $\exists x A \vee \exists x B$. **All formulas in this section are assumed to be subformulas of γ.**

We shall identify subformulas of γ with their *occurrences* in γ. For example, in the formula γ of the form

$$A \wedge (\mathbf{A} \vee \mathbf{B})$$

the boldfaced $\mathbf{A}$ is considered as a subformula of $(\mathbf{A} \vee \mathbf{B})$, but the ordinary A is not a subformula of $(\mathbf{A} \vee \mathbf{B})$. A subformula φ of γ is called *conjunctive* iff it is an occurrence in a subformula $\varphi \wedge \psi$ or in $\psi \wedge \varphi$. A *conjunctive superformula* of φ is a superformula ψ of φ that is conjunctive. The *least conjunctive superformula* of φ is the conjunctive superformula ψ of φ such that any other conjunctive superformula of φ is a superformula of ψ. Let us note that conjunctive superformulas not necessarily exist. For example, γ has no conjunctive superformula. Any formula having a conjunctive superformula has the unique least conjunctive superformula.

We can enumerate all conjunctive subformulas, for example in the order of their occurrences in γ. Thus we can inambiguously use "the nth conjunctive (sub)formula".

Let $A_1, A_2, \ldots$ be a set of predicate symbols not occurring in γ. We say that $A_n(x_1, \ldots, x_m)$ is *the γ-name for a subformula φ* iff there is the subformula ψ such that

1. ψ is the n-th conjunctive subformula of γ.
2. ψ is the least conjunctive superformula of φ.
3. $x_1, \ldots, x_m$ are all variables of ψ in the order of their occurrences in ψ.

γ-names not always exist. We shall use *the set of γ-names* of a subformula. The set of γ-names is either the empty set $\emptyset$ or a singleton $\{A_i(x_1, \ldots, x_k)\}$. Thus, we can consider sets of γ-names as clauses. Also note that different formulas may have the same γ-names.

The following example illustrates least conjunctive superformulas and γ-names:

Example 3.1 *Let γ be the formula*

$$(\exists x(A(x) \wedge (B(x) \vee \exists y C(x, y))) \wedge \exists z D(z)) \vee E$$

All subformulas of γ and their γ-names are shown in the following table:

Subformula	*least conjunctive superformula*	*γ-name*
$(\exists x(A(x) \wedge (B(x) \vee \exists y C(x,y))) \wedge \exists z D(z)) \vee E$		
$\exists x(A(x) \wedge (B(x) \vee \exists y C(x,y))) \wedge \exists z D(z)$		
$\exists x(A(x) \wedge (B(x) \vee \exists y C(x,y)))$	$\exists x(A(x) \wedge (B(x) \vee \exists y C(x,y)))$	A_1
$A(x) \wedge (B(x) \vee \exists y C(x,y))$	$\exists x(A(x) \wedge (B(x) \vee \exists y C(x,y)))$	A_1
$A(x)$	$A(x)$	$A_2(x)$
$B(x) \vee \exists y C(x,y)$	$B(x) \vee \exists y C(x,y)$	$A_3(x)$
$B(x)$	$B(x) \vee \exists y C(x,y)$	$A_3(x)$
$\exists y C(x,y)$	$B(x) \vee \exists y C(x,y)$	$A_3(x)$
$C(x,y)$	$B(x) \vee \exists y C(x,y)$	$A_3(x)$
$\exists z D(z)$	$\exists z D(z)$	A_4
$D(z)$	$\exists z D(z)$	A_4
E		

Lemma 3.1 *Let φ be a subformula of γ. Then*

1. *If ψ is the least conjunctive superformula of φ then $\models \forall(\varphi \supset \psi)$.*
2. *If there is no conjunctive superformula of φ then $\models \forall(\varphi \supset \gamma)$.*

This lemma partially explains the need in introducing least conjunctive superformulas. There are deterministic chains of inferences in sequent systems, where formulas with $\vee$ or $\exists$ occur. For example,[1]

$$\cfrac{\cfrac{\Gamma, \varphi}{\Gamma, \varphi \vee \psi}}{\Gamma, \exists x(\varphi \vee \psi)}$$

By restricting to conjunctive superformulas only, we eliminate these deterministic chains, making them in one step.

Now we can formulate our deductive system for the inverse method. The proof-search consists of two parts: the equality solution part and the sequent proof part. We encode formulas by their γ-names. Then sequents (i.e. sets of formulas) become clauses (i.e. sets of literals). In the first part of the proof, we try to solve equalities in the formula, generating a set of closures not containing equality. A closure not containing equality will be called a *solution clause.* Solution clauses $C \cdot \sigma$ will always be written in the ordinary clause notation, i.e. as $C\sigma$. This set of solution clauses is obtained by basic superposition from *initial closures* defined below. In the second part, the usual sequent style proof search by the inverse method is performed. The solution clauses generated in the first part are used as the initial clauses in the sequent part. Hence the name "equality elimination" for our method: we eliminate equality in the first part of the proof.

Initial closures are generated according to one of the three rules:

1. Whenever a literal $s \neq t$ occurs in γ and C is the set of γ-names for this occurrence of $s \neq t$, the closure $s = t, C \cdot \varepsilon$ is an initial closure.
2. Whenever $s = t$ occurs in γ and C is the set of γ-names for this occurrence of $s = t$, the closure $s \neq t, C \cdot \varepsilon$ is an initial closure.

[1]Since we are dealing with formulas in negation normal form, we can represent sequents as sets of formulas, similar to [Sch 60].

3. Let literals $P(s_1, \ldots, s_n)$ and $\neg P(t_1, \ldots, t_n)$, where P is different from $=$, occur in γ and C_1, C_2 are their sets of γ-names. Let the substitution σ rename variables such that variables of $P(s_1, \ldots, s_n)\sigma, C_1\sigma$ and $P(t_1, \ldots, t_n), C_2$ are disjoint. Then the closure $s_1\sigma \neq t_1, \ldots, s_n\sigma \neq t_n, C_1\sigma, C_2 \cdot \varepsilon$ is an initial closure.

The equality elimination method consists of five inference rules. The *basic superposition* rules are defined as in [BGLS 92]. We always assume that premises of rules have disjoint variables. The first three rules derive closures from the initial closures defined above. Applications of the first three rules form the *equational solution part of the proof.* The aim of the equational solution part of the proof is to generate *solution clauses* that are used as axioms of the sequent part of the proof. The sequent part of the proof uses the last two rules. The aim of the sequent part of the proof is to derive the empty clause.

Basic right superposition

$$\frac{(C, s = t) \cdot \sigma_1 \qquad (u[s'] = v, D) \cdot \sigma_2}{(u[t] = v, C, D) \cdot \sigma_1\sigma_2\rho}$$

where

1. ρ is a most general unifier of $s\sigma_1$ and $s'\sigma_2$;
2. $t\sigma_1\rho \not\succeq s\sigma_1\rho$;
3. $v\sigma_2\rho \not\succeq u[s']\sigma_2\rho$;
4. s' is not a variable.

(We remind that variables of the premises must be disjoint.)

Basic left superposition

$$\frac{(C, s = t) \cdot \sigma_1 \qquad (u[s'] \neq v, D) \cdot \sigma_2}{(u[t] \neq v, C, D) \cdot \sigma_1\sigma_2\rho}$$

with the same conditions as for basic right superposition. When there are several inequalities in the first clause, we shall use one more restriction on the basic left superposition: we only allow for superposition in the leftmost inequality.[2]

Equality solution

$$\frac{(C, s \neq t) \cdot \sigma}{C \cdot \sigma\rho}$$

where ρ is a most general unifier of $s\sigma, t\sigma$.

[2]According to our definitions, a clause is a set of literals, so the leftmost literal may sound as nonsense, but this restriction can easily be formalized using a variant of *locking* [Boy 71].

Conjunction rule Let $\varphi \wedge \psi$ be a subformula of γ, and N_φ, N_ψ and $N_{\varphi\wedge\psi}$ be sets of γ-names of φ, ψ and $\varphi \wedge \psi$. Let $\bar{x}$ be all variables of $\varphi \wedge \psi$. Let clauses Γ, Δ have no occurrences of equality. Then the following is a conjunction rule:

$$\frac{\Gamma, N_\varphi[\bar{x}/\bar{s}] \qquad \Delta, N_\psi[\bar{x}/\bar{t}]}{(\Gamma, \Delta, N_{\varphi\wedge\psi}[\bar{x}/\bar{s}])\theta}$$

where θ is a most general unifier of $\bar{s}$ and $\bar{t}$.

Factoring rule Let the clause Γ, A, B have no occurrences of equality. Then the following is a factoring rule

$$\frac{\Gamma, A, B}{(\Gamma, A)\theta}$$

where θ is a most general unifier of A, B.

The following theorems are true about the method:

Theorem 3.1 (Soundness) *If there is a derivation of the empty clause from the initial closures then γ is provable.*

Theorem 3.2 (Completeness) *If γ is provable then there is a derivation of the empty clause from the initial closures.*

Proofs of these two theorems are given in Appendix A.

Consider several examples. The following example is taken from [BH 92].

Example 3.2 *The formula to be proved is*

$$\exists x(g(f(x)) \neq c \vee (f(x) \neq g(x) \wedge (B \vee \neg B))) \vee g(g(a)) = g(f(b))$$

There are two conjunctive subformulas with the following names:

$$\begin{aligned} A_1(x) &\equiv f(x) \neq g(x) \\ A_2 &\equiv B \vee \neg B \end{aligned}$$

The corresponding conjunction rule is[3]

$$\frac{\Gamma, A_1(s) \qquad \Delta, A_2}{\Gamma, \Delta}$$

The initial closures are

[3]When atoms in the premises of a conjunction rule have different number of arguments one can always make them equal by miniscoping the quantifier $\exists$. In this example we could as well deal with the miniscoped formula

$$\exists x(g(f(x)) \neq c) \vee (\exists x(f(x) \neq g(x)) \wedge (B \vee \neg B)) \vee g(g(a)) = g(f(b))$$

In this case we would have $A_1 \equiv \exists x(f(x) \neq g(x))$ and the conjunction rule would become even simpler:

$$\frac{\Gamma, A_1 \qquad \Delta, A_2}{\Gamma, \Delta}$$

1. $g(f(x)) = c \cdot \varepsilon$
2. $f(x) = g(x), A_1(x) \cdot \varepsilon$
3. $g(f(b)) \neq g(g(a)) \cdot \varepsilon$
4. A_2

The first three are generated by equalities in the goal, the last one by the pair $B, \neg B$. Note that the last one is a solution clause, and thus it is written in the usual clause notation.

As the order $\succ$ we consider the recursive path ordering with $f > g > c > a > b$. Basic superposition and equality solution generate the following clauses:

5. $g(g(a)) \neq c \cdot \varepsilon$ [left superposition from 1,3]
6. $g(g(x)) = c, A_1(y) \cdot [x/y]$ [right superposition from 2,1]
7. $A_1(a)$ [left superposition + equality solution from 6,5]

The sequent part of the proof is very simple:

8. $\Box$ [conjunction rule from 7,4]

The next example is taken from [Mas 71]:

Example 3.3 *The formula to be proved is*

$$\exists x((a \neq x \vee \neg G(b) \vee G(f(f(x)))) \wedge (\neg G(f(x)) \vee x = b))$$

There are two conjunctive subformulas with the following names:

$$\begin{array}{lcl} A_1(x) & \equiv & a \neq x \vee \neg G(b) \vee G(f(f(x))) \\ A_2(x) & \equiv & \neg G(f(x)) \vee x = b \end{array}$$

The corresponding conjunction rule is

$$\frac{\Gamma, A_1(s) \qquad \Delta, A_2(t)}{(\Gamma, \Delta)\theta}$$

where θ is a most general unifier of s, t.

The initial closures are

1. $a = x, A_1(x) \cdot \varepsilon$
2. $x \neq b, A_2(x) \cdot \varepsilon$
3. $b \neq f(f(x)), A_1(x), A_1(y) \cdot \varepsilon$
4. $f(x) \neq f(f(y)), A_2(x), A_1(y) \cdot \varepsilon$

The first two are generated by equalities in the goal, the last two by pairs $G, \neg G$.

As the order $\succ$ we consider the recursive path ordering with $f > b > a$. Basic superposition and equality solution generate the following clauses:

5. $A_2(b)$ [equality solution from 2]
6. $b \neq a, A_1(x), A_1(y), A_1(z) \cdot [x/f(f(y))]$ [left superposition from 1,3]
7. $A_1(b), A_1(f(f(y))), A_1(y), A_1(z)$ [left superposition + equality solution from 1,6]
8. $A_1(x), A_2(f(x)))$ [equality solution from 4]

The sequent part of the proof is a bit more complicated:

9. $A_1(b), A_1(f(f(y))), A_1(y)$ [factoring from 7]
10. $A_1(b), A_1(f(f(b)))$ [factoring from 9]
11. $A_1(f(f(b)))$ [conjunction rule from 5,10]
12. $A_1(f(b))$ [conjunction rule from 8,11]
13. $A_1(b)$ [conjunction rule from 8,12]
14. $\Box$ [conjunction rule from 5,13]

The last example is taken from [Bec 94]:

Example 3.4 *The formula to be proved is*

$$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x) \vee b \neq c \vee \neg P(g(g(a)), b) \vee P(a, c)$$

There are two conjunctive subformulas with the following names:

$$\begin{aligned} A_1(x) &\equiv g(x) \neq f(x) \\ A_2(x) &\equiv x = a \end{aligned}$$

The corresponding conjunction rule is

$$\frac{\Gamma, A_1(s) \qquad \Delta, A_2(t)}{(\Gamma, \Delta)\theta}$$

where θ is a most general unifier of s, t.

The initial closures are

1. $f(x) = g(x), A_1(x) \cdot \varepsilon$
2. $A_2(x) \vee x \neq a \cdot \varepsilon$
3. $g(f(x)) = x \cdot \varepsilon$
4. $b = c \cdot \varepsilon$
5. $g(g(a)) \neq a, b \neq c \cdot \varepsilon$

The first four are generated by equalities in the goal, the last one by the pair $\neg P(g(g(a)), b), P(a, c)$.

As the order $\succ$ we consider the recursive path ordering with $f > g > b > c > a$. Basic superposition and equality solution generate the following clauses:

6. $g(g(x)) = x, A_1(y) \cdot [x/y]$ [right superposition from 1,3]
7. $A_2(a)$ [equality solution from 2]
8. $A_1(y), b \neq c \cdot [y/a]$ [left superposition + equality solution from 6,5]
9. $A_1(a)$ [left superposition + equality solution from 8,4]

The sequent part of the proof is also very simple:

10. $\Box$ [conjunction rule from 9,7]

There are many optimizations not considered in this paper. The most powerful one is *subsumption* that we can apply in both parts of the proof in order to prevent generation of unnecessary clauses. Subsumption for the basic strategy has been explained in [Deg 82, BGLS 92]. In the sequent part of the proof we can use ordinary subsumption, or even a stronger subsumption for sequent proofs proposed in [Vor 92].

Section 4

The connection method

We plan to consider equality elimination for the connection method in detail in a forthcoming publication. Here we only consider Example 3.3 in the connection format. We shall refer to definitions and statements from [Gal 92].

The formula to be proved is

$$\exists x((a \neq x \vee \neg G(b) \vee G(f(f(x)))) \wedge (\neg G(f(x)) \vee x = b))$$

As in [Gal 92], to prove this formula we try to establish unsatisfiability of its negation

$$\forall x((a = x \wedge G(b) \wedge \neg G(f(f(x)))) \vee (G(f(x)) \wedge x \neq b))$$

Denote the matrix of this formula via $M(x)$. According to Theorems 4.3 and 7.7 of [Gal 92] this formula is inconsistent iff there is a path-acceptable equational mating for some amplification $M(x_1) \wedge \ldots M(x_n)$. Consider the set of vertical paths through this amplification. According to Definition 4.4 of [Gal 92], the set of vertical paths $vp(A)$ through amplification A is the set of sets of literals defined inductively as follows:

If A is a literal, then $vp(A) = \{\{A\}\}$;
If $A = (B \wedge C)$, then $vp(A) = \{\pi_1 \bigcup \pi_2 \mid \pi_1 \in vp(B), \pi_2 \in vp(C)\}$;
If $A = (B \vee C)$, then $vp(A) = vp(B) \bigcup vp(C)$

Denote

$$\begin{array}{rcl} A_1(x) & = & \{a = x, G(b), \neg G(f(f(x)))\} \\ A_2(x) & = & \{G(f(x)), x \neq b\} \end{array}$$

From the above definition of vp it follows that the set $vp(M(x_1) \wedge \ldots \wedge M(x_n))$ has the form

$$\{A_{i_1}(x_1) \bigcup \ldots \bigcup A_{i_n}(x_n) \mid 1 \leq j \leq n \text{ and } i_j \in \{1, 2\}\}$$

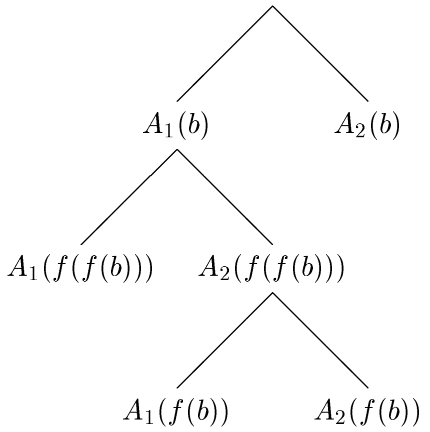
Note now that the solution clauses used in Example 3.3 define the following set of inconsistent instantiated partial paths:

$A_2(b)$
$A_1(b) \bigcup A_1(y) \bigcup A_1(f(f(y))) \bigcup A_1(z)$
$A_1(x) \bigcup A_2(f(x))$

Indeed, any of these partial paths contains an inconsistent instantiated mated set (see [Gal 92]) of $M(x_1) \wedge \ldots \wedge M(x_n)$. For example, for the second of these partial paths it will be the instantiated mated set

$$\{a = b, a = f(f(y)), G(b), \neg G(f(f(y)))\}$$

The composition of these instantiated mated sets forms an equational mating. It remains to find an instantiated amplification for which this mating is path acceptable. Such amplification can be extracted from the sequent part of the proof. The instantiated amplification for our example is shown on the following picture:



Thus, in this amplification $n = 3$, $x_1 = b$, $x_2 = f(f(b))$ and $x_3 = f(b)$. Note that this tree can be easily changed into a tableau proof of γ.

Appendix A

Proof of main theorems

In order to prove Theorems 3.1 and 3.2 we define several notions and state and prove several lemmas. They will be illustrated on Example 3.4, i.e. when γ is

$$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x) \vee b \neq c \vee \neg P(g(g(a)), b) \vee P(a, c)$$

Extend the naming relation to *all* positive subformulas in the following way. Let N be any one-to-one function from the set of all (occurrences) of positive subformulas of γ to $\{A_1, A_2, \ldots\}$ such that for the nth conjunctive subformula φ_n of γ we have $N(\varphi_n) = A_n$. Then γ-names for an arbitrary positive subformula φ of γ are defined as $A_k(x_1, \ldots, x_m)$, where $N(\varphi) = A_k$ and $x_1, \ldots, x_m$ are all variables of φ. For example, possible names of all positive subformulas of γ from our example are shown in the following table:

name	formula
A_3	$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x) \vee b \neq c \vee \neg P(g(g(a)), b) \vee P(a, c)$
A_4	$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x) \vee b \neq c \vee \neg P(g(g(a)), b)$
A_5	$P(a, c)$
A_6	$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x) \vee b \neq c$
A_7	$\neg P(g(g(a)), b)$
A_8	$\exists x(g(x) \neq f(x) \wedge x = a) \vee \exists x(g(f(x)) \neq x)$
A_9	$b \neq c$
A_{10}	$\exists x(g(x) \neq f(x) \wedge x = a)$
A_{11}	$\exists x(g(f(x)) \neq x)$
$A_{12}(x)$	$g(x) \neq f(x) \wedge x = a$
$A_1(x)$	$g(x) \neq f(x)$
$A_2(x)$	$x = a$
$A_{13}(x)$	$g(f(x)) \neq x$

Denote the name of a formula φ by $\overline{\varphi}$.

Consider now the set Φ_1 of formulas defined as

$$\Phi_1 = \{\forall(\varphi \equiv \overline{\varphi}) \mid \varphi \text{ is a positive subformula of } \gamma\} \bigcup \{\neg\overline{\gamma}\}$$

Lemma A.1 *The formula γ is provable iff Φ_1 is inconsistent.*

Proof. First note that γ is provable iff $\neg\gamma$ is inconsistent. Thus, the statement can be reformulated as: $\neg\gamma$ is consistent iff Φ_1 is consistent.

1. Let M be a model for Φ_1. Then $M \models \neg\overline{\gamma}$ and $M \models \gamma \equiv \overline{\gamma}$. Thus, $M \models \neg\gamma$.

2. Let $M \models \neg\gamma$. Consider the model M' defined as follows: predicates from the language of γ are interpreted as in M, and the interpretation of new predicates is given by $\varphi \equiv \overline{\varphi}$. Obviously, $M' \models \Phi_1$.

$\square$

Let Φ_2 be the following set of formulas:

$$\begin{aligned} \Phi_2 = \; & \{\forall(\overline{\varphi \wedge \psi} \equiv \overline{\varphi} \wedge \overline{\psi}) \mid \varphi \wedge \psi \text{ is a subformula of } \gamma\} \bigcup \\ & \{\forall(\overline{\varphi \vee \psi} \equiv \overline{\varphi} \vee \overline{\psi}) \mid \varphi \vee \psi \text{ is a subformula of } \gamma\} \bigcup \\ & \{\forall(\overline{\exists x \varphi} \equiv \exists x \overline{\varphi}) \mid \exists x \varphi \text{ is a subformula of } \gamma\} \bigcup \\ & \{\forall(\overline{L} \equiv L) \mid L \text{ is a literal in } \gamma\} \bigcup \\ & \{\neg\overline{\gamma}\} \end{aligned}$$

In our example Φ_2 is the set of the following formulas

$$\begin{aligned} & A_3 \equiv A_4 \vee A_5 \\ & A_4 \equiv A_6 \vee A_7 \\ & A_5 \equiv P(a, c) \\ & A_6 \equiv A_8 \vee A_9 \\ & A_7 \equiv \neg P(g(g(a)), b) \\ & A_8 \equiv A_{10} \vee A_{11} \\ & A_9 \equiv b \neq c \\ & A_{10} \equiv \exists x A_{12}(x) \\ & A_{11} \equiv \exists x A_{13}(x) \\ & \forall x(A_{12}(x) \equiv A_1(x) \wedge A_2(x)) \\ & \forall x(A_{13}(x) \equiv g(f(x)) \neq x) \\ & \forall x(A_1(x) \equiv g(x) \neq f(x)) \\ & \forall x(A_2(x) \equiv x = a) \\ & \neg A_3 \end{aligned}$$

Lemma A.2 *γ is provable iff Φ_2 is inconsistent.*

Proof. It is easy to see that Φ_1 and Φ_2 are equivalent: each of them can be obtained from the other by substitution of equivalent formulas for equivalent. Then apply Lemma A.1. $\square$

Let Φ_3 be the set of formulas defined as

$$\Phi_3 = \{\forall(A \supset \overline{B}) \mid \forall(\overline{B} \equiv A) \in \Phi_2\} \bigcup \{\neg\overline{\gamma}\}$$

Lemma A.3 *γ is provable iff Φ_3 is inconsistent.*

Proof. First prove that Φ_3 is consistent iff Φ_2 is consistent.

1. Obviously $\Phi_2 \models \Phi_3$.

2. Assume that $M \models \Phi_3$. Let $\varphi_1, \dots, \varphi_n$ be any enumeration of all proper subformulas of γ such that if φ_i is a subformula of φ_j, then $i \le j$. Let $\psi_i \in \Phi_3$ are formulas having the form $\forall(\delta_i \supset \overline{\varphi_i})$, for some δ_i. Let χ_i be formulas $\forall(\delta_i \equiv \overline{\varphi_i})$. Note that $\Phi_3 = \{\psi_1, \dots, \psi_n, \neg\overline{\gamma}\}$ and $\Phi_2 = \{\chi_1, \dots, \chi_n, \neg\overline{\gamma}\}$. Denote by Π_i sets $\{\chi_1, \dots, \chi_{i-1}, \psi_i, \dots, \psi_n, \neg\gamma\}$. Then $\Pi_1 = \Phi_3$. We prove, by induction on i, that Π_i are satisfiable. For $i = 1$ the model $M_1 = M$ satisfies Π_1. Let M_{i-1} be a model for Π_{i-1}. The predicate symbol of $\overline{\varphi_i}$ has only two occurrences in Π_i: one in $\forall(\delta_i \supset \overline{\varphi_i})$ and one negative occurrence in some ψ_j with $i < j$. Consider the model M_i in which the interpretation of the predicate of $\overline{\varphi_i}$ is defined by the formula $\delta_i \equiv \overline{\varphi_i}$ and the interpretation of other predicates is the same as in M_{i-1}. Obviously, $M_i \models \Pi_i$.

Note that $\Pi_{n+1} = \Phi_2$. These prove that Φ_2 is inconsistent iff Φ_3 is inconsistent. Then apply Lemma A.2. $\square$

In our example Φ_3 is the set of the following formulas

$$
\begin{array}{l}
A_4 \vee A_5 \supset A_3 \\
A_6 \vee A_7 \supset A_4 \\
P(a,c) \supset A_5 \\
A_8 \vee A_9 \supset A_6 \\
\neg P(g(g(a)), b) \supset A_7 \\
A_{10} \vee A_{11} \supset A_8 \\
b \neq c \supset A_9 \\
\exists x A_{12}(x) \supset A_{10} \\
\exists x A_{13}(x) \supset A_{11} \\
\forall x(A_1(x) \wedge A_2(x) \supset A_{12}(x)) \\
\forall x(g(f(x)) \neq x \supset A_{13}(x)) \\
\forall x(g(x) \neq f(x) \supset A_1(x)) \\
\forall x(x = a \supset A_2(x)) \\
\neg A_3
\end{array}
$$

One can note that this set can be considerably simplified, so that only the names of conjunctive subformulas will matter. The following lemma shows that names of non-conjunctive subformulas can be eliminated:

Lemma A.4 *The following are true:*

1. *Assume that* Γ *is a set of formulas such that*

$$\Gamma = \Delta \bigcup \{\forall(\overline{\varphi_1} \vee \overline{\varphi_2} \supset \overline{\varphi_3}), \forall(\psi_1 \supset \overline{\varphi_1}), \dots, \forall(\psi_m \supset \overline{\varphi_1}), \forall(\chi_1 \supset \overline{\varphi_2}) \dots, \forall(\chi_n \supset \overline{\varphi_2})\}$$

where Δ, ψ_i, χ_j *have no occurrences of the predicate symbols of* $\overline{\varphi_1}, \overline{\varphi_2}$. *Then* Γ *is satisfiable iff the set*

$$\Gamma = \Delta \bigcup \{\forall(\psi_1 \supset \overline{\varphi_3}), \dots, \forall(\psi_m \supset \overline{\varphi_3}), \forall(\chi_1 \supset \overline{\varphi_3}), \dots, \forall(\chi_n \supset \overline{\varphi_3})\}$$

is satisfiable.

2. *Assume that* Γ *is a set of formulas such that*

$$\Gamma = \Delta \bigcup \{\forall(\exists x\overline{\varphi} \supset \overline{\psi}), \forall(\chi_1 \supset \overline{\varphi}) \ldots, \forall(\chi_n \supset \overline{\varphi})\}$$

where $\Delta, \overline{\psi}, \chi_i$ *do not have occurrences of the predicate symbol of* $\overline{\varphi}$. *Then* Γ *is satisfiable iff the set*

$$\Gamma = \Delta \bigcup \{\forall(\chi_1 \supset \overline{\psi}) \ldots, \forall(\chi_n \supset \overline{\psi})\}$$

is satisfiable.

3. *Assume that* Γ *is a set of formulas such that*

$$\Gamma = \Delta \bigcup \{\forall(\varphi_1 \supset \overline{\gamma}), \ldots, \forall(\varphi_n \supset \overline{\gamma}), \neg\overline{\gamma}\}$$

where Δ, φ_i *do not have occurrences of the predicate symbol of* $\overline{\gamma}$. *Then* Γ *is satisfiable iff the set*

$$\Gamma = \Delta \bigcup \{\forall(\neg\varphi_1), \ldots, \forall(\neg\varphi_n)\}$$

is satisfiable.

Proof. Obvious. □

Consider how we can simplify Φ_3 using repeated applications of this lemma. On the first step we eliminate all disjunctions obtaining

$P(a,c) \supset A_3$
$\neg P(g(g(a)), b) \supset A_3$
$b \neq c \supset A_3$
$\exists x A_{12}(x) \supset A_3$
$\exists x A_{13}(x) \supset A_3$
$\forall x(A_1(x) \wedge A_2(x) \supset A_{12}(x))$
$\forall x(g(f(x)) \neq x \supset A_{13}(x))$
$\forall x(g(x) \neq f(x) \supset A_1(x))$
$\forall x(x = a \supset A_2(x))$
$\neg A_3$

Then we eliminate all occurrences of $\exists$ obtaining

$P(a,c) \supset A_3$
$\neg P(g(g(a)), b) \supset A_3$
$b \neq c \supset A_3$
$\forall x(A_1(x) \wedge A_2(x) \supset A_3)$
$\forall x(g(f(x)) \neq x \supset A_3)$
$\forall x(g(x) \neq f(x) \supset A_1(x))$
$\forall x(x = a \supset A_2(x))$
$\neg A_3$

Finally, we get rid of $\overline{\gamma}$, i.e. A_3:

$$\neg P(a,c)$$
$$P(g(g(a)),b)$$
$$b=c$$
$$\forall x \neg(A_1(x) \wedge A_2(x))$$
$$\forall x(g(f(x))=x)$$
$$\forall x(g(x) \neq f(x) \supset A_1(x))$$
$$\forall x(x=a \supset A_2(x))$$

Note that this set can be considered as a set of clauses.

$$\neg P(a,c)$$
$$P(g(g(a)),b)$$
$$b=c$$
$$\neg A_1(x), \neg A_2(x)$$
$$g(f(x))=x$$
$$g(x)=f(x), A_1(x))$$
$$x \neq a, A_2(x)$$

From now on we shall work with clauses instead of formulas. Denote the set of clauses obtained after this simplification by Φ_4. Lemma A.4 immediately gives

Lemma A.5 *γ is provable iff Φ_4 is inconsistent.*

The following trick allows one to get rid of the predicate symbols occurring in γ, thus only leaving new predicate symbols and equality:

Lemma A.6 *Let P be a predicate symbol different from equality, S a set of clauses such that*

$$S = S' \bigcup \{P(\bar{t_i}), C_i \mid i \in \{1,\dots,n\}\} \bigcup \{\neg P(\bar{s_j}), D_j \mid j \in \{1,\dots,m\}\}$$

where C_i, D_j are clauses and P does not occur in S' or any of C_i, D_j. Then S is consistent iff the set

$$R = S' \bigcup \{\bar{t_i} \neq \bar{s_j}, C_i, D_j \mid i \in \{1,\dots,n\}, j \in \{1,\dots,m\}\}$$

is consistent.

Proof.

$\Rightarrow$ is obvious, since $S \models R$.

$\Leftarrow$ Let R be consistent. Then there is a term model $M_R \models R$. Let $\approx$ be the congruence on $T(\Sigma)$ defining the carrier set of M_R. For a term t, denote by $[t]_\approx$ the equivalence class of t under $\approx$. Enrich M_R by the new predicate P obtaining the model M_S such that $M_S \models S$ as follows. For simplicity, but without loss of generality, we assume that P is unary. Obviously, it is enough to define P only on elements of the form $[t_i\sigma]_\approx$ and $[s_j\sigma]_\approx$. Let $[t_i\sigma]_\approx$ is such that $[s_j\theta]_\approx \neq [t_i\sigma]_\approx$, for all j and θ. Then we let $M_S \models P(t_i\sigma)$. Let $[s_j\theta]_\approx$ is such that $[t_i\theta]_\approx \neq [s_j\theta]_\approx$, for all i and σ. Then we let $M_S \models \neg P(s_j\theta)$.

Let an equivalence class w.r.t. $\approx$ in M_R contain terms $t_{i_1}\sigma_{i_1}, t_{i_2}\sigma_{i_2}, \ldots$ and terms $s_{j_1}\sigma_{j_1}, s_{j_2}\sigma_{j_2}, \ldots$. We prove that at least one of the truth values for $P([t_{i_1}\sigma_{i_1}])$ will not make S false in M_S. Assume the inverse. Let $M_S \not\models P([t_{i_1}\sigma_{i_1}])$ makes S false. Then there is p such that $M_S \not\models P(t_{i_p}\sigma_{i_p}) \vee C_{i_p}\sigma_{i_p}$.

Assume also that $M_S \models P([s_{j_1}\theta_{j_1}])$ makes S false. Then there is q such that $M_S \not\models \neg P(s_{j_q}\theta_{j_q}) \vee D_{j_q}\theta_{j_q}$. Then we obtain $M_R \not\models (t_{i_p} \neq s_{j_q} \vee C_{i_p} \vee D_{j_q})\delta$, where $\delta = \sigma_{i_p} \bigcup \theta_{j_q}$ (in the union operation $\bigcup$ we use that domains of σ_{i_p} and θ_{J_q} are disjoint). Contradiction.

Arbitrarily define P on all other equivalence classes. It follows that $M_S \models S$.

□

According to this lemma, we can replace all clauses containing predicate symbols from the original signature as stated in the lemma. Denote thus obtained set of clauses by S_γ. In our example the set S_γ looks as follows:

$a \neq g(g(a)), c \neq b$
$b = c$
$\neg A_1(x), \neg A_2(x)$
$g(f(x)) = x$
$g(x) = f(x), A_1(x))$
$x \neq a, A_2(x)$

Lemma A.7 *The set of clauses S_γ consists of*

1. *All clauses C, where $C \cdot \varepsilon$ is an initial closure.*
2. *All clauses $\neg A, \neg B, C$, where there exists a subformula $\varphi \wedge \psi$ of γ such that*
   (a) *A is the set of γ-names of φ.*
   (b) *B is the set of γ-names of ψ.*
   (c) *C is the set of γ-names of $\varphi \wedge \psi$.*

Proof. By routine inspection of the construction of S_γ. □

Lemmas A.5 and A.6 prove

Lemma A.8 *The formula γ is provable iff the set S_γ is inconsistent.*

The following lemma shows that all basic superpositions can be made in the beginning of the proof:

Lemma A.9 *S_γ is inconsistent iff there is a set of clauses S derived from S_γ by basic superpositions and equality solutions such that*

1. *S does not contain equality;*
2. *S is inconsistent.*

Proof. Due to completeness of the basic restriction, there is a resolution refutation of S_γ in which all paramodulations precede all other rules, and all equality solutions precede all resolutions (see [Deg 82, DV 86b]). Let us call such refutations *regular.* In S_γ all arguments of non-equality predicates are variables. Then the basic strategy requires that all paramodulations be made into equations, thus becoming left or right superpositions. For regular refutations, equality solutions can only be made into clauses containing no positive equations, and resolution can only be applied to clauses without equations. Take as S the set of all clauses in a regular refutation which are obtained by only superpositions and equality solutions and which do not contain equality. □

In our example a possible set S consists of the following three clauses:

$$\neg A_1(x), \neg A_2(x)$$
$$A_1(a)$$
$$A_2(a)$$

See Example 3.4 for derivations of $A_1(a)$ and $A_2(a)$ by basic superpositions and equality solutions. In the terminology of Section 3, $A_1(a)$ and $A_2(a)$ are *solution clauses.*

Lemma A.10 *Let S be an set of clauses without equality such that the clauses of S are the following:*

1. *All clauses of the form*

$$\neg A, \neg B, C \tag{A.1}$$

such that

$$\frac{\Gamma, A[\bar{x}/\bar{s}] \qquad \Delta, B[\bar{x}/\bar{t}]}{(\Gamma, \Delta, C)\theta}$$

is a conjunction rule.

2. *All clauses of the form*

$$\neg A, \neg B \tag{A.2}$$

such that

$$\frac{\Gamma, A[\bar{x}/\bar{s}] \qquad \Delta, B[\bar{x}/\bar{t}]}{(\Gamma, \Delta)\theta}$$

is a conjunction rule.

3. *Some positive clauses.*

Then S is inconsistent iff there is a derivation of the empty clause from the positive clauses in C using conjunction rules and factoring rules.

Proof. Obvious since every application of a conjunction rule is an application of hyperresolution rule with nuclei of the form (A.1) or (A.2). □

Combining Lemmas A.8, A.9 and A.10 we obtain

Theorem A.1 (Soundness) *If there is a derivation of the empty clause from the initial closures then γ is provable.*

Theorem A.2 (Completeness) *If γ is provable then there is a derivation of the empty clause from the initial closures.*

Bibliography

[And 86] P.B. Andrews. *An introduction to type theory: to truth through proof.* Academic Press, 1986.

[BDP 89] L. Bachmair, N. Dershowitz, and D. Plaisted. Completion without failure. In H. Aït Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2, pages 1–30. Academic Press, 1989.

[BGLS 92] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation and superposition. In D. Kapur, editor, *11th International Conference on Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 462–476, Saratoga Springs, NY, USA, June 1992. Springer Verlag.

[BG 94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BH 92] 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.

[Bec 94] B. Beckert. Adding equality to semantic tableaux. In K. Broda, M. D'Agostino, R. Goré, R. Jahnson, and S. Reeves, editors, *Proceedings, 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Abingdon*, pages 29–41, Imperial College, London, TR-94/5, 1994.

[Ben 90] D. Benanav. Simultaneous paramodulation. In M.E. Stickel, editor, *Proc. 10th Int. Conf. on Automated Deduction*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 442–455, 1990.

[Bib 87] W. Bibel. *Automated theorem proving.* Vieweg Verlag, 1987. 2nd edition.

[Boy 90] T. Boy de la Tour. Minimizing the number of clauses by renaming. In M.E. Stickel, editor, *Proc. 10th CADE*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 558–572, 1990.

[Boy 71] R.S. Boyer. *Locking: a Restriction to Resolution.* PhD thesis, Univ. of Texas, Austin, 1971.

[Bra 75] D. Brandt. Proving theorems with the modification method. *SIAM Journal of Computing*, 4:412–430, 1975.

[Deg 79] A.I. Degtyarev. The strategy of monotone paramodulation (in Russian). In *Fifth Soviet All-Union Conference on Mathematical Logic*, page 39, Novosibirsk, 1979.

[Deg 82] A.I. Degtyarev. On the forms of inference in calculi with equality and paramodulation. In Yu.V. Kapitonova, editor, *Automation of Research in Mathematics*, pages 14–26. Institute of Cybernetics, Kiev, Kiev, 1982.

[DV 86a] A.I. Degtyarev and A.A. Voronkov. Automated theorem proving I (in Russian). *Kibernetika*, (3), 1986. The journal is cover-to-cover translated to English.

[DV 86b] A.I. Degtyarev and A.A. Voronkov. Methods of handling equality in machine theorem proving (in Russian). *Kibernetika*, (3), 1986. The journal is cover-to-cover translated to English.

[DV 87] A.I. Degtyarev and A.A. Voronkov. Automated theorem proving II (in Russian). *Kibernetika*, (4), 1987. The journal is cover-to-cover translated to English.

[DV 94a] A. Degtyarev and A. Voronkov. A new procedural interpretation of Horn clauses with equality. UPMAIL Technical Report 89, Uppsala University, Computing Science Department, November 1994.

[DV 94b] A. Degtyarev and A. Voronkov. Equality elimination for semantic tableaux. UPMAIL Technical Report 90, Uppsala University, Computing Science Department, November 1994.

[DJ 90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 6, pages 243–309. North Holland, Amsterdam, 1990.

[Ede 91] E. Eder. Consolution and its relation with resolution. In *Proc. IJCAI'91*, pages 132–136, 1991.

[Fit 90] M. Fitting. *First Order Logic and Automated Theorem Proving*. Springer Verlag, New York, 1990.

[DH 86] V.J. Digricoli and M.C. Harrison. Equality-based binary resolution. *Journal of the Association for Computing Machinery*, 33:253–289, 1986.

[GNRS 92] 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):3–44, 1992.

[Gal 92] J. Gallier. Unification procedures in automated deduction methods based on matings: a survey. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, pages 439–485. Elsevier Science, 1992.

[Kan 63] S. Kanger. A simplified proof method for elementary logic. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning. Classical Papers on Computational Logic*, volume 1, pages 364–371. Springer Verlag, 1983. Originally appeared in 1963.

[Lif 68] V. Lifschitz. Specialized forms of derivation in predicate calculus with equality and functional symbols (in Russian). In *Trudy MIAN*, volume 98, pages 5–25. 1968. English translation in: Proc. Steklov Institute of Math., AMS, Providence, 1971.

[Lif 89] V. Lifschitz. What is the inverse method? *Journal of Automated Reasoning*, 5(1):1–23, 1989.

[Llo 87] J.W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer Verlag, 1987.

[MAT 94] A.M. Maeda, J.-I. Aoe, and H. Tomabechi. Signature-check based unification filter. *Software — Practice and Experience*, 24(7):603–622, July 1994.

[Mas 71] S.Yu. Maslov. The generalization of the inverse method to predicate calculus with equality (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 20:80–96, 1971. English translation in: Journal of Soviet Mathematics 1, no. 1.

[MM 83] S.Yu. Maslov and G. Mints. The proof-search theory and the inverse method (in Russian). In Mints G., editor, *Mathematical Logic and Automatic Theorem Proving*, pages 291–314. Nauka, Moscow, 1983.

[Mas 87] S.Yu. Maslov. *Theory of Deductive Systems and its Applications.* MIT Press, 1987.

[Mor 69] J.B. Morris. E-resolution: Extension of resolution to include the equality relation. In Donald E. Walker and Lewis M. Norton, editors, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-69)*, pages 287–294. Washington, D.C., May 1969.

[NR 92a] R. Nieuwenhuis and A. Rubio. Basic superposition is complete. In *ESOP'92*, volume 582 of *Lecture Notes in Computer Science*, pages 371–389. Springer Verlag, 1992.

[NR 92b] R. Nieuwenhuis and A. Rubio. Theorem proving with ordering constrained clauses. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 477–491, Saratoga Springs, NY, USA, June 1992. Springer Verlag.

[Ore 69] V.P. Orevkov. On non-increasing rule applications for equality (in Russian). *Zapiski Nauchnyh Seminarov LOMI*, 16:152–156, 1969. English Translation in: Seminars in Mathematics: Steklov Math. Inst. 16, Consultants Bureau, NY-London, 1971.

[Pra 60] D. Prawitz. An improved proof procedure. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning. Classical Papers on Computational Logic*, volume 1, pages 162–201. Springer Verlag, 1983. Originally appeared in 1960.

[Pet 93] U. Petermann. A framework for integrating equality reasoning into extension procedure. In *Proc. Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 195–207, 1993.

[Rus 91] M. Rusinowitch. Theorem proving with resolution and superposition: an extension of the Knuth and Bendix completion procedure as a complete set of inference rules. *Journal of Symbolic Computations*, 11:21–49, 1991.

[Sch 60] Schütte, K. *Beweistheorie (in German).* Springer Verlag, 1960.

[Vor 92] A.A. Voronkov. Theorem proving in non-standard logics based on the inverse method. In D. Kapur, editor, *11th International Conference on Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 648–662, Saratoga Springs, NY, USA, June 1992. Springer Verlag.