UPMAIL Technical Report No. 128
September 14, 1996
ISSN 1100-0686

Herbrand's Theorem and Equational Reasoning: Problems and Solutions

Anatoli Degtyarev*

Computing Science Department
Uppsala University
Box 311, S 751 05 Uppsala,
Sweden
email anatoli@csd.uu.se

Yuri Gurevich†

EECS Department
University of Michigan
Ann Arbor, MI, 48109-2122
USA
email gurevich@umich.edu

Andrei Voronkov‡

Computing Science Department
Uppsala University
Box 311, S-751 05 Uppsala,
Sweden
email voronkov@csd.uu.se

*Supported by a grant from the Swedish Royal Academy of Sciences and by TFR grant 92-818.
†Partially supported by grants from NSF, ONR and the Faculty of Science and Technology of Uppsala University.
‡Partially supported by TFR grant 96-859.

Abstract

The famous Herbrand's theorem of mathematical logic plays an important role in automated theorem proving. In the first part of this article, we recall the theorem and formulate a number of natural decision problems related to it. Somewhat surprisingly, these problems happen to be equivalent. One of these problems is the so-called simultaneous rigid E-unification problem. In the second part, we survey recent result on the simultaneous rigid E-unification problem.[2]

Section 1

Problems

> This simultaneous rigid E-unification ... It has too procedural a definition.
> *Vaughan Pratt, private communication.*

We start with some decision problems having non-procedural definitions. Most of these problems naturally stem from the famous Herbrand theorem.

1.1 Herbrand's theorem and six decision problems

The Herbrand theorem [28] asserts that an existential first-order formula $\exists\bar{x}\varphi(\bar{x})$ is provable (in classical first-order logic) if and only if a particular disjunction $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_n)$ is provable. Here each $\bar{t}_i$ is a sequence (of the length of $\bar{x}$) of terms in the signature of φ. Note that $n = 1$ may not suffice: for example, for the formula

$$\exists x(P(a) \vee P(b) \supset P(x)) \tag{1.1}$$

the disjunction $\big(P(a) \vee P(b) \supset P(a)\big) \vee \big(P(a) \vee P(b) \supset P(b)\big)$ is provable, but no formula of the form $P(a) \vee P(b) \supset P(t)$ is provable. Recall that, by the completeness theorem for classical logic, a formula is provable if and only if it holds in all models of its signature. In the above example, in some models the formula $P(a) \vee P(b) \supset P(x)$ holds with $x = a$, and in the others with $x = b$.

One can see at once that the Herbrand theorem is relevant to automated reasoning. It suggests proving $\exists\bar{x}\varphi(\bar{x})$ in two steps: guess an appropriate number n (often called the *multiplicity* to denote the number of copies of φ that can be used) and then find particular terms $\bar{t}_1, \ldots, \bar{t}_n$. The following decision problem arises naturally:

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

The Herbrand Skeleton problem is equivalent to the following one.

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

(To reduce Herbrand Skeleton to Formula Instantiation, transform a given instance $(\varphi(\bar{x}), n)$ of Herbrand Skeleton to an instance $\varphi(\bar{x}_1) \vee \ldots \vee \varphi(\bar{x}_n)$ of Formula Instantiation where different $\bar{x}_i$ have different variables.)

Formula Instantiation is obviously solvable in the case of pure logic, that is logic without equality. To illustrate this, consider the formula

$$\big(P(f(y)) \supset P(x)\big) \wedge \big(Q(y) \supset Q(g(x)) \vee Q(f(z))\big). \tag{1.2}$$

The question corresponding to this instance of Formula Instantiation is whether there are terms r, s, t such that

$$\big(P(f(s)) \supset P(r)\big) \wedge \big(Q(s) \supset Q(g(r)) \vee Q(f(t))\big) \tag{1.3}$$

is provable. This formula is provable if and only if the following is a valid derivation in a sequent calculus (for example, Kanger's calculus from [31, 32]):

$$\cfrac{\cfrac{P(f(s)) \to P(r)}{\to P(f(s)) \supset P(r)}\;(\to\supset) \qquad \cfrac{\cfrac{Q(s) \to Q(g(r)), Q(f(t))}{Q(s) \to Q(g(r)) \vee Q(f(t))}\;(\to\vee)}{\to Q(s) \supset Q(g(r)) \vee Q(f(t))}\;(\to\supset)}{\to \big(P(f(s)) \supset P(r)\big) \wedge \big(Q(s) \supset Q(g(r)) \vee Q(f(t))\big)}\;(\to\wedge)$$

This is a valid derivation if and only if $f(s) = r$ and either $s = g(r)$ or $s = f(t)$. Here "$f(s) = r$" means of course that $f(s)$ and r coincide. For writing equations, it is convenient to have a different (from $=$) notation for the (formal) equality predicate. We will use $\simeq$ for that purpose. Instead of $\neg s \simeq t$ we will write $s \not\simeq t$. One can seek for appropriate terms r, s, t among terms of the *Herbrand universe* of the signature Υ of formula (1.2). The Herbrand universe of Υ is the set of ground terms (that is terms without variables) of Υ, augmented with a constant if necessary. The necessity to augment Υ with a constant (that is nullary function symbol) arises when Υ has no constant of its own and therefore has no ground terms.

Early methods of automated theorem proving would check provability of instances of (1.3) generating, in some order, all possible values for r, s, t in the Herbrand universe. This proved to be inefficient. Later, starting with Prawitz [46], people used *free variable methods* where instantiation of variables to terms is delayed until some extra information about instantiation is obtained.

In our example, such extra information can be obtained by analysis of the top sequents of the derivation. Such terms r, s, t exist if and only if one of the systems of term equations $\{f(y) \simeq x, y \simeq g(x)\}$ or $\{f(y) \simeq x, y \simeq f(z)\}$ has a solution. Solvability of such systems of equations can be checked in almost linear time by the unification algorithm of Martelli and Montanari [41] or in linear time by the unification algorithm of Paterson and Wegman [44] using more complicated data structures.

For those unfamiliar with sequent calculi, we sketch an alternative reduction of formula instantiation to systems of term equations on the same example, i.e. formula (1.2). Formula (1.2) is provable if and only if so is its conjunctive normal form

$$\big(\neg P(f(s)) \vee P(r)\big) \wedge \big(\neg Q(s) \vee Q(g(r)) \vee Q(f(t))\big).$$

This conjunctive normal form is provable if and only if both formulas $\neg P(f(s)) \vee P(r)$ and $\neg Q(s) \vee Q(g(r)) \vee Q(f(t))$ are provable. As above, such appropriate terms r, s, t exist if and only if one of the systems of term equations $\{f(y) \simeq x, y \simeq g(x)\}$ or $\{f(y) \simeq x, y \simeq f(z)\}$ has a solution.

What happens in the case of logic with equality? We shall give an answer later.

Until now we spoke about classical logic where provability of an existential formula $\exists\bar{x}\varphi(\bar{x})$ is characterized by the Herbrand theorem. In intuitionistic logic, a closed formula $\exists\bar{x}\varphi(\bar{x})$ is provable if and only if there is a single sequence of (ground) terms $\bar{t}$ such that $\varphi(\bar{t})$ is provable. For example, formula (1.1) is not provable in intuitionistic logic: this well-known fact shows that Formula Instantiation is equivalent to the following decision problem:

Problem 3 (Existential Intuitionistic)
Is a given existential formula $\exists\bar{x}\varphi(\bar{x})$ provable in intuitionistic logic?

We shall denote classical provability of a formula φ by $\vdash \varphi$ and intuitionistic provability of φ by $\vdash_{\text{int}} \varphi$.

One can show the equivalence of Existential Intuitionistic to a more general problem: the decidability of the prenex fragment of intuitionistic logic. Recall that a formula is prenex if it has all its quantifiers up front.

Problem 4 (Prenex Intuitionistic)
Is a given prenex formula provable in intuitionistic logic?

There are several proofs of the equivalence of Existential Intuitionistic and Prenex Intuitionistic. One way is to use the same skolemization that is used in classical logic. However, standard skolemization introduces new function symbols which one may want to avoid. (Some results explained in Section 2 depend on the signature). There is an alternative way of skolemizing given by Degtyarev and Voronkov [13], where only new constants are introduced. We shall informally illustrate it here. Existential Intuitionistic obviously reduces to Prenex Intuitionistic. Let us take some instance of Prenex Intuitionistic and reduce it to an instance of Existential Intuitionistic.

Consider, for example, the case when the prenex formula has the form $\forall x\exists y\forall z\exists u\varphi(x,y,z,u)$ and the signature Υ. This formula is provable if and only if so is the formula $\exists y\forall z\exists u\varphi(c,y,z,u)$, where c is a new constant. The second formula is in turn provable if and only if so is some formula of the form $\forall z\exists u\varphi(c,s,z,u)$, where s is a ground term in the signature $\Upsilon \cup \{c\}$. In the same way reduce the provability of the third formula to finding a ground term t of the signature $\Upsilon \cup \{c,d\}$ such that d is a new constant and $\varphi(c,s,d,t)$ is provable.

But how can we speak about finding the desired terms s and t? There is a way. Suppose for simplicity that $\Upsilon = \{f,a\}$ where f is a binary function symbol and a is a constant and consider the following two formulas:

$$\psi_1(x) = f(a,a) \simeq a \wedge c \simeq a \supset x \simeq a$$
$$\psi_2(x) = f(a,a) \simeq a \wedge c \simeq a \wedge d \simeq a \supset x \simeq a$$

The formula $\psi_1(s)$ is provable in intuitionistic (or classical) logic if and only if s is a ground term in the signature $\Upsilon \cup \{c\}$. The if implication is provable by an easy induction on s. To prove the only if implication, use e.g. [32] according to which $\psi_1(s)$ is provable if and only if $s \simeq a$ can be derived from $a \simeq a$ in a calculus with axioms $f(a,a) \simeq a, c \simeq a$ and derivation rules that replace equals with equals.

Similarly, $\psi_2(t)$ is provable if and only if t is a ground term in $\Upsilon \cup \{c,d\}$. The target instance of Existential Intuitionistic is

$$\exists y\exists u(\psi_1(y) \wedge \psi_2(u) \wedge \varphi(c,y,d,u))$$

In classical logic, every formula algorithmically reduces to a prenex formula and thus the provability problem for prenex formulas is undecidable, even for pure predicate logic (no equality or function symbols), even for "tiny" fragments of pure predicate logic; see e.g. [7]. In intuitionistic logic, this is not the case which diminishes of course the role of prenex formulas. The explanation is that classical equivalences like

$$\varphi \supset \exists x \psi(x), \text{ where } x \text{ does not occur in } \varphi, \text{ is equivalent to } \exists x(\varphi \supset \psi(x))$$

do not hold in intuitionistic logic. For example, applying this equivalence to intuitionistically unprovable formula (1.1), we obtain the intuitionistically provable formula $P(a) \vee P(b) \supset \exists x P(x)$. For formulas without equality, Prenex Intuitionistic is decidable and PSPACE-complete (Voronkov [49]).

There are additional interesting problems that the Herbrand theorem gives rise to. One example is related to the *derivation skeletons* as defined in Voronkov [50]. A derivation skeleton is obtained from a derivation by omitting all formulas (or sequents) and keeping only the names of inference rules (but ignoring the equality replacement rule ($\simeq$)) [50]. Consider an example. In this and further examples we often omit parentheses in terms with unary function symbols (for example we write ffb instead of $f(f(b))$).

Example 1.1 Figure 1.1 shows a derivation of the formula

$$\forall x(hx \simeq a) \supset \exists y((ha \not\simeq y \vee y \simeq ghy) \wedge (hb \not\simeq fy \vee y \simeq gfy))$$

in the cut-free sequent calculus of Kanger [32] (with some formulas omitted for simplicity) and its skeleton.

The following decision problem naturally arises:

Problem 5 (Skeleton Instantiation)
Given a formula φ and a derivation skeleton $\mathcal{S}$, is there a derivation of φ with the skeleton $\mathcal{S}$?

Skeleton Instantiation is very sensitive to the choice of the calculus and the precise notion of a skeleton. In the case of formal arithmetic, Skeleton Instantiation may lead to the second-order unification problem (Goldfarb [23]). However, for some cut-free sequent calculi and appropriate notions of skeleton, Skeleton Instantiation is equivalent to Prenex Intuitionistic. Their equivalence can be shown using the so-called constraint technique of Voronkov [50]. In the sequel, we speak about the version of Skeleton Instantiation which is equivalent to Prenex Intuitionistic.

Let us now come back to Formula Instantiation. This problem has several reformulations which serve as a basis for automated deduction methods exploiting the matrix characterization of validity introduced by Prawitz [46]: in particular the method of matings (Andrews [1]) and the connection method (Bibel [6]). The Herbrand theorem in these methods is reformulated in terms of matrices and paths. It can also be formulated in a similar way in terms of tableaux and branches. In matrix-based methods, given a formula, one constructs a sequence of *matrices* and tries to find a substitution that makes all vertical paths in some matrix of this sequence inconsistent.

Example 1.2 The formula of Example 1.1 can be transformed to a matrix:

$$\begin{bmatrix} \multicolumn{2}{c}{hx_1 \simeq a} \\ \multicolumn{2}{c}{hx_2 \simeq a} \\ \begin{bmatrix} ha \simeq y \\ y \not\simeq ghy \end{bmatrix} & \begin{bmatrix} hb \simeq fy \\ y \not\simeq gfy \end{bmatrix} \end{bmatrix}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{ha \simeq a, hb \simeq a, a \simeq ga \to ga \simeq ga}{ha \simeq a, hb \simeq a, a \simeq ga \to ga \simeq gha}\,(\simeq)}{ha \simeq a, hb \simeq a, a \simeq ga \to ga \simeq ghga}\,(\simeq)}{ha \simeq a, hb \simeq a \to a \not\simeq ga, ga \simeq ghga}\,(\to\neg)}{ha \simeq a, hb \simeq a \to a \not\simeq ga \vee ga \simeq ghga}\,(\to\vee) \qquad \dfrac{\dfrac{\dfrac{\dfrac{ha \simeq a, hb \simeq a, hb \simeq fga \to ga \simeq ga}{ha \simeq a, hb \simeq a, hb \simeq fga \to ga \simeq ghb}\,(\simeq)}{ha \simeq a, hb \simeq a, hb \simeq fga \to ga \simeq gfga}\,(\simeq)}{ha \simeq a, hb \simeq a \to hb \not\simeq fga \vee ga \simeq gfga}\,(\to\neg)}{ha \simeq a, hb \simeq a \to hb \not\simeq fga \vee ga \simeq gfga}\,(\to\vee)}{ha \simeq a, hb \simeq a \to (a \not\simeq ga \vee ga \simeq ghga) \wedge (hb \not\simeq fga \vee ga \simeq gfga)}\,(\to\wedge)}{ha \simeq a, hb \simeq a \to (ha \not\simeq ga \vee ga \simeq ghga) \wedge (hb \not\simeq fga \vee ga \simeq gfga)}\,(\simeq)}{ha \simeq a, hb \simeq a \to \exists y((ha \not\simeq y \vee y \simeq ghy) \wedge (hb \not\simeq fy \vee y \simeq gfy))}\,(\to\exists)}{ha \simeq a, \forall x(hx \simeq a) \to \exists y((ha \not\simeq y \vee y \simeq ghy) \wedge (hb \not\simeq fy \vee y \simeq gfy))}\,(\forall\to)}{\forall x(hx \simeq a) \to \exists y((ha \not\simeq y \vee y \simeq ghy) \wedge (hb \not\simeq fy \vee y \simeq gfy))}\,(\forall\to)}{\to \forall x(hx \simeq a) \supset \exists y((ha \not\simeq y \vee y \simeq ghy) \wedge (hb \not\simeq fy \vee y \simeq gfy))}\,(\to\supset)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\quad}{\quad}\,(\to\neg)}{\quad}\,(\to\vee) \qquad \dfrac{\dfrac{\quad}{\quad}\,(\to\neg)}{\quad}\,(\to\vee)}{\quad}\,(\to\wedge)}{\quad}\,(\to\exists)}{\quad}\,(\forall\to)}{\quad}\,(\forall\to) \;\; \dfrac{\quad}{\quad}\,(\to\supset)$$

Figure 1.1: Derivation and its skeleton,

The matrix has two vertical paths:

$$\{hx_1 \simeq a, hx_2 \simeq a, ha \simeq y, y \not\simeq ghy\},$$
$$\{hx_1 \simeq a, hx_2 \simeq a, hb \simeq fy, y \not\simeq gfy\}.$$

The substitution $\{a/x_1, b/x_2, ga/y\}$ makes both vertical path inconsistent. (This substitution has been used to instantiate variables in quantifier rules in the derivation of Figure 1.1).

Matrix-based characterization of validity gives rise to the following decision problem:

Problem 6 (Matrix Instantiation)
Given a matrix M, is there a substitution θ such that every vertical path through M is inconsistent?

Matrix Instantiation is also equivalent to Formula Instantiation. Recall that Formula Instantiation is decidable in the case of logic without equality. It follows that all six problems are decidable in the case of logic without equality.

1.2 Simultaneous rigid E-unification

For logic with equality, the matrix instantiation problem naturally leads to the following combinatorial problem (Gallier, Raatz and Snyder [19]).

Problem 7 (SREU)
Given finite sets of equations E_i and equations $s_i \simeq t_i$, where $i \in \{1, \ldots, n\}$, is there a substitution θ such that $E_i\theta \vdash (s_i\theta \simeq t_i\theta)$ for all $i \in \{1, \ldots, n\}$?

Let us give formal definitions.

Definition 1 A *rigid equation* is an expression of the form $E \vdash_\forall s \simeq t$, where E is a finite set of equations and s, t are terms. The set E is *the left-hand side* of this rigid equation, and $s \simeq t$ is its *right-hand side.*

A *solution* to such a rigid equation is any substitution θ such that $E\theta \vdash (s\theta \simeq t\theta)$. (In this case, classical provability $\vdash$ is equivalent to intuitionistic). The *rigid E-unification* problem is the problem of determining whether a given rigid equation has a solution.

A *system of rigid equation* is any finite set of rigid equations. A *solution* to a system R of rigid equations is any substitution which solves every rigid equation in R. The *simultaneous rigid E-unification* problem (SREU) is the problem of determining whether a given system of rigid equations has a solution.

It is not difficult to see that SREU is equivalent to Matrix Instantiation. For example, Matrix Instantiation for the matrix of Example 1.2 is equivalent to the following instance of SREU:

$$\begin{array}{rcl} hx_1 \simeq a, hx_2 \simeq a, ha \simeq y & \vdash_\forall & y \simeq ghy \\ hx_1 \simeq a, hx_2 \simeq a, hb \simeq fy & \vdash_\forall & y \simeq gfy \end{array}$$

They are equivalent in a very strong sense: every solution to this system of rigid equations makes all vertical paths in the matrix inconsistent, and vice versa.

In general, an instance of Matrix Instantiation reduces to a disjunction of instances of SREU. For example, Matrix Instantiation for the matrix

$$\left[\begin{bmatrix} a \simeq b \\ b \not\simeq c \\ a \not\simeq x \end{bmatrix} \qquad x \not\simeq b \right]$$

has a solution if and only if one of the following instances of SREU has a solution:

$$\begin{array}{rcl} a \simeq b & \vdash_\forall & b \simeq c \\ & \vdash_\forall & x \simeq b \end{array} \qquad\qquad \begin{array}{rcl} a \simeq b & \vdash_\forall & a \simeq x \\ & \vdash_\forall & x \simeq b \end{array}$$

Conversely, SREU is reducible to Matrix Instantiation.

Summarizing the above considerations, we have

Theorem 1 The seven decision problems described above are all reducible to each other. If one of them is decidable (undecidable) then so are the others.

> Of course it is undecidable, because E-unification is undecidable.
> *Remark of a student of Linköping University.*

In the rest of this section, we will talk mostly about (non-simultaneous) rigid E-unification, i.e. the case of *one* rigid equation. Simultaneous rigid E-unification will be considered in the next section.

The reader may have heard about E-unification. Here E stand for *equational.* The word "rigid" is introduced to distinguish rigid E-unification from E-unification. The latter problem can be formulated as follows. Given a finite set of equations $E = \{s_1 \simeq t_1, \ldots, s_n \simeq t_n\}$ and an equation $s \simeq t$, find a substitution θ such that $\forall(s_1 \simeq t_1), \ldots, \forall(s_n \simeq t_n) \vdash s\theta \simeq t\theta$, where each $\forall$ binds all variables in its range. For rigid E-unification, we try to find a substitution θ such that $\vdash s_1\theta \simeq t_1\theta \wedge \ldots \wedge s_n\theta \simeq t_n\theta \supset s\theta \simeq t\theta$. In E-unification, all variables in the equations $s_i \simeq t_i$ are treated as universal, while all variables in rigid equations are *rigid:* if we substitute a term t for a variable v in any part of a rigid equation, we must substitute t for v in the whole rigid equation. The word "simultaneous" means that we have to find a simultaneous solution to several rigid equations.

Consider an example: $x \cdot y \simeq y \cdot x \vdash_\forall a \cdot (b \cdot a) \simeq (a \cdot b) \cdot a$. This rigid equation has no solutions. However, we have $\forall x \forall y(x \cdot y \simeq y \cdot x) \vdash a \cdot (b \cdot a) \simeq (a \cdot b) \cdot a$, and hence the terms $a \cdot (b \cdot a)$ and $(a \cdot b) \cdot a$ are E-unifiable in the theory consisting of one axiom $\forall x \forall y(x \cdot y \simeq y \cdot x)$. In the derivation of $a \cdot (b \cdot a) \simeq (a \cdot b) \cdot a$ from $\forall x \forall y(x \cdot y \simeq y \cdot x)$ one needs to use at least two copies of the equation $x \cdot y \simeq y \cdot x$. The corresponding rigid equation with the two copies in the left-hand side:

$$x_1 \cdot y_1 \simeq y_1 \cdot x_1, x_2 \cdot y_2 \simeq y_2 \cdot x_2 \vdash_\forall a \cdot (b \cdot a) \simeq (a \cdot b) \cdot a$$

also has a solution, namely the substitution $\{a/x_1, a/x_2, b \cdot a/y_1, b/y_2\}$.

When the left-hand side E of a rigid equation $E \vdash_\forall s \simeq t$ is ground, then rigid E-unification coincides with E-unification: any solution to this rigid equation is a E-unifier of $s \simeq t$ and vice versa. Hence, several results proved for E-unification are also valid for rigid E-unification. For example, in Hullot [30] it is shown that in the case of ground left-hand sides there is a finite complete set of E-unifiers and in Kozen [35] it is shown that the E-unification problem is NP-complete in the case of ground left-hand sides. We demonstrate that this problem is NP-hard, even for some fixed left-hand side. Take any propositional formula p using only $\neg$ and $\wedge$. The satisfiability problem for such formulas is NP-complete. Consider the following rigid equation:

$$\begin{array}{l} \neg\texttt{true} \simeq \texttt{false}, \neg\texttt{false} \simeq \texttt{true}, \\ \texttt{false} \wedge \texttt{false} \simeq \texttt{false}, \texttt{false} \wedge \texttt{true} \simeq \texttt{false}, \texttt{true} \wedge \texttt{false} \simeq \texttt{false}, \texttt{true} \wedge \texttt{true} \simeq \texttt{true} \\ \quad \vdash_\forall p \simeq \texttt{true} \end{array}$$

It is easy to see that p is satisfiable if and only if this rigid equation has a solution.

When left-hand sides are non-ground, rigid E-unification is fundamentally different from E-unification. E-unification is in general undecidable. In contrast, rigid E-unification is decidable and NP-complete (Gallier et.al. [20, 21, 22]). The first procedure for rigid E-unification, described in [20], was not intended to be most efficient; the purpose was to prove the NP-completeness. Later, several papers presented other algorithms for rigid E-unification (Goubault [24], Becher and Petermann [3], De Kogel [18], Plaisted [45]). These newer algorithms have been presented as either more efficient or simpler than the original algorithm of [20]. Notice, however, that application-wise the more important problem is SREU, simultaneous rigid E-unification. The six decision problems of Subsection 1.1, equivalent to SREU, witness this fact.

Section 2

Solutions

In this section we review recent results on the decidability of simultaneous rigid E-unification and its fragments. The relation of these results to automated reasoning may be found in the survey [17].

2.1 The undecidability

Question: *We have seen papers asserting that simultaneous rigid E-unification is NP-complete and NEXPTIME-complete. Which is right?*
Answer: *Both are wrong.*
Andrei Voronkov, private email communication.

Research on simultaneous rigid E-unification started in 1987 and most results have been proven in 1995–1996. Surprisingly, rigid variables in the context of resolution theorem proving have been used much earlier by Chang and Lee [8, 36] in the so-called V-resolution and V-paramodulation rules. They tried to use rigid variables in order to capture Prawitz's procedure by resolution and to formalize the idea of reasoning with bounded resources.

In 1988–92, several papers gave faulty proofs of the decidability of simultaneous rigid E-unification. Finally, in 1995, the problem was proven to be undecidable by Degtyarev and Voronkov in [12] by a reduction from the monadic semi-unification problem whose undecidability was proven by Baaz [2]. Degtyarev and Voronkov gave more comprehensive undecidability proofs in [15, 14] using reductions from second-order unification, proven undecidable by Goldfarb [23], and from Hilbert's tenth problem. We shall explain the ideas of the undecidability proof in [15] and then consider further results on simultaneous rigid E-unification below.

The undecidability proof in [15] gives a direct encoding of second-order unification by simultaneous rigid E-unification. We shall present second-order unification rather informally; for more details see Goldfarb [23] or Degtyarev and Voronkov [15]. Second-order unification differs from ordinary unification in that it allows second-order variables. A second-order variable x of arity n may occur in terms in the form $x(t_1, \ldots, t_n)$. Second-order substitutions substitute for such variables λ-terms of the form $\lambda w_1 \ldots w_n.t$, where t is any first-order term. The application of the substitution $\{\lambda w_1 \ldots w_n.t/x\}$ to a term $x(t_1, \ldots, t_n)$ yields the term $t\{t_1/w_1, \ldots, t_n/w_n\}$. Ordinary first-order variables can be viewed as second-order variables of arity 0.

By introducing sufficiently many new variables, any instance of second-order unification can be reduced to a set of equations of the form $x(t_1, \ldots, t_n) \simeq t$, where $n \geq 0$ and $t, t_1, \ldots, t_n$ are first-order terms. For example, an equation $f(z(y(a))) \simeq z(y(b))$ can be reduced to

$$\{y(a) \simeq u_1, z(u_1) \simeq u_2, y(b) \simeq u_3, z(u_3) \simeq f(u_2)\}.$$

Consider an example second-order equation

$$x(y, g(y)) \simeq f(g(z), a, y) \tag{2.1}$$

We are interested in ground solutions, i.e. in solutions θ such that $x\theta, y\theta, z\theta$ are ground terms. Furthermore, we restrict attention to solutions that use only those function symbols of positive arity which are present in the given equation. Among the admissible solutions for the example equation are:

$$\{\lambda w_1 w_2 . f(g(b), a, b)/x, b/y, b/z\} \tag{2.2}$$
$$\{\lambda w_1 w_2 . f(w_2, a, w_1)/x, b/y, b/z\} \tag{2.3}$$
$$\{\lambda w_1 w_2 . f(w_2, w_1, a)/x, a/y, a/z\} \tag{2.4}$$
$$\{\lambda w_1 w_2 . f(w_1, a, g(b))/x, g(b)/y, b/z\} \tag{2.5}$$

In order to encode second-order unification, we use some properties of simultaneous rigid E-unification [15]. First, an appropriate rigid equation $Gr(t, \Upsilon)$ allows one to indicate precisely those substitutions which make a given term t a ground term of a given signature Υ provided that Υ contains at least one constant c:

$$Gr(t, \Upsilon) \rightleftharpoons \{f(c, \ldots, c) \simeq c \mid f \in \Upsilon\} \vdash_\forall t \simeq c.$$

Lemma 2.1 A substitution θ solves $Gr(t, \Upsilon)$ if and only if $t\theta$ is a ground term of the signature Υ.

We have already used this property of rigid equations (and even sketched a proof of it) in Section 1 when we discussed the alternative Skolemization in intuitionistic logic (after presenting Problem 4). In fact, rigid equations allow one to represent any regular set (Goubault [25], Plaisted [45] and Veanes [47]).

Second, simultaneous rigid E-unification can be used to represent application of substitutions. We shall temporarily use substitution notation $\{t_1/c_1, \ldots, t_n/c_n\}$ where c_i are constants. This denotes the operation of the simultaneous replacement of *all* occurrences of c_i by t_i, $i \in \{1, \ldots, n\}$.

Lemma 2.2 Let $c_1, \ldots, c_n$ be different constants and $t_1, \ldots, t_n$ first-order terms such that no c_i occurs in any t_j. Then

$$c_1 \simeq t_1, \ldots, c_n \simeq t_n \vdash s_1 \simeq s_2 \ \text{ iff } \ s_1\{t_1/c_1, \ldots, t_n/c_n\} = s_2\{t_1/c_1, \ldots, t_n/c_n\}.$$

The two lemmas suffice to represent second-order unification. Indeed, consider any second-order equation of the form $x(t_1, \ldots, t_n) \simeq t$. Without loss of generality, we may assume that the signature Υ of the equation contains at least one constant. Let $c_1, \ldots, c_n$ be new constants. Form the following system R of rigid equations:

$$\begin{array}{l} Gr(t_1, \Upsilon) \\ \ldots \\ Gr(t_n, \Upsilon) \\ Gr(t, \Upsilon) \\ Gr(x, \Upsilon \cup \{c_1, \ldots, c_n\}) \\ c_1 \simeq t_1, \ldots, c_n \simeq t_n \vdash_\forall x = t \end{array}$$

and assume that a substitution θ solves R. By Lemma 2.1, all terms $t_1\theta, \ldots, t_n\theta, t\theta$ are ground terms in the signature Υ, and $x\theta$ is a ground term over $\Upsilon \cup \{c_1, \ldots, c_n\}$. Since $c_1, \ldots, c_n$ do not occur in $t_1\theta, \ldots, t_n\theta, t\theta$, we can apply Lemma 2.2 to the last rigid equation in R, obtaining

$$x\theta\{t_1\theta/c_1, \ldots, t_n\theta/c_n\} = t\theta$$

Let $\theta = \{s_1/x_1, \ldots, s_m/x_m, s/x\}$. It is easy to see that θ solves R if and only if the substitution $\{s_1/x_1, \ldots, s_m/x_m, \lambda c_1 \ldots c_n.s/x\}$ solves the second-order equation $x(t_1, \ldots, t_n) \simeq t$. Thus, we can encode second-order unification by simultaneous rigid E-unification in quite a natural way.

If we apply this encoding to our example second-order equation (2.1), we obtain the following set of rigid equations:

$$\begin{array}{l}
f(a,a,a) \simeq a, g(a) \simeq a, b \simeq a \vdash_\forall y \simeq a \\
f(a,a,a) \simeq a, g(a) \simeq a, b \simeq a \vdash_\forall z \simeq a \\
f(a,a,a) \simeq a, g(a) \simeq a, b \simeq a, c_1 \simeq a, c_2 \simeq a \vdash_\forall x \simeq a \\
c_1 \simeq y, c_2 \simeq g(y) \vdash_\forall x \simeq f(g(z), a, y)
\end{array}$$

Some solutions of this system of rigid equations are

$$\begin{array}{l}
\{f(g(b), a, b)/x, b/y, b/z\} \\
\{f(c_2, a, c_1)/x, b/y, b/z\} \\
\{f(c_2, c_1, a)/x, a/y, a/z\} \\
\{f(c_1, a, g(b))/x, g(b)/y, b/z\}
\end{array}$$

The reader can compare them with solutions (2.2)–(2.5) of the original second-order equation.

Thus second-order unification effectively reduces to simultaneous rigid E-unification. This reduction, the Goldfarb result mentioned above and Theorem 1 yield

Theorem 2 Problems 1–7 are undecidable.

Voda and Komara [48] considered a specialization of Herbrand Skeleton of the following form. Suppose that $n \geq 1$ is fixed. Given a quantifier-free formula $\varphi(\bar{x})$, are there term sequences $\bar{t}_1, \ldots, \bar{t}_n$ such that the formula $\varphi(\bar{t}_1) \vee \ldots \vee \varphi(\bar{t}_n)$ is provable? They prove that this problem is undecidable, for every n.

2.2 Special cases of simultaneous rigid E-unification

Currently, there are several proofs of the undecidability of simultaneous rigid E-unification. Besides the proofs mentioned above, there are proofs of Voda and Komara [48], Plaisted [45] and Veanes [47]. Plaisted proved a stronger result.

Theorem 3 (Plaisted [45])
Simultaneous rigid E-unification is undecidable even when the left-hand sides of rigid equations are ground.

The technique used by Plaisted is different from the previously used techniques. He reduced the Post correspondence problem to simultaneous rigid E-unification with ground left-hand sides; furthermore Plaisted uses only systems of six rigid equations with three variables for the reduction.

Veanes [47] improved the construction of Plaisted. He has shown how to simulate arbitrary Turing machine using six equations and two variables.

Theorem 4 (Veanes [47])
Simultaneous rigid E-unification remains undecidable under the following restrictions:

1. the signature contains, in addition to constants, only one function symbol and the arity of that symbol is ≥ 2;

2. the left-hand sides of the rigid equations are ground;

3. there are only two variables;

4. there are only six rigid equations[1].

This implies the undecidability of Problem 1–7 for any signature having, in addition to constants, at least one function symbol of arity ≥ 2. The restriction on the ground left-hand sides gives rise to the following restrictions in other problems:

1. For Herbrand Skeleton, Formula Instantiation and Existential Intuitionistic: all negative atoms are ground.

2. For Prenex Intuitionistic: every variable occurring in a negative atom is bound by a universal quantifier.

3. For Skeleton Instantiation: every variable occurring in a negative atom is bound by either a positive universal quantifier or a negative existential quantifier.

4. For Matrix Instantiation: all positive atoms are ground.

The restriction of two on the number of variables yields the following restrictions.

1. For Formula Instantiation: φ has at most two variables.

2. For Existential Intuitionistic and Prenex Intuitionistic: the $\exists\exists$-fragment of intuitionistic logic is undecidable.

3. For Skeleton Instantiation: there are at most two rules marked $(\rightarrow \exists)$ or $(\forall \rightarrow)$;

4. For Matrix Instantiation: the matrix has at most two variables.

These results leave open the decision problem for the one-variable case and for the case with function symbols of arity ≤ 1 only (we call it the monadic case). The monadic case will be addressed in the next subsection. The case of one variable has been proven decidable in [9]; the complexity of the one-variable case is addressed there as well. The decidability of the one-variable fragment of SREU gives rise to decidable fragments of the other decision problems. Taking into account the restrictions imposed by the two-variable condition, it is not difficult to see what those decidable fragments are. One less obvious case is that of Prenex Intuitionistic where the implication is this: the $\forall^*\exists\forall^*$-fragment of intuitionistic logic with equality is decidable.

[1] It was noted recently by Gurevich and Veanes that three equations suffice

2.3 The monadic case

We shall distinguish monadic languages by the number of unary function symbols. Following Degtyarev, Matiyasevich and Voronkov [11], denote by Σ_n the signature with n unary function symbols and a countable number of constants. Let SREU_n be the fragment of simultaneous rigid E-unification restricted to Σ_n. First, we note that the case $n > 2$ is equivalent to the case $n = 2$.

Theorem 5 SREU_n reduces to SREU_2.

The proof uses a standard coding of words in the arbitrary finite alphabet by words in a two-letter alphabet. Let S be an instance of SREU_n with variables $x_1, \ldots, x_m$ and constants $c_1, \ldots, c_k$ where $k \geq 1$. Code a unary function symbol f_i in Σ_n by the sequence $f_1 f_2^i$ of the unary symbols in Σ_2.

Let S' be obtained from S by replacing each f_i with $f_1 f_2^i$ and then adding, for each $j \in \{1, \ldots, m\}$, an additional rigid equation

$$\{f_1 f_2^i c_1 \simeq c_1 \mid i \in \{1, \ldots, n\}\} \cup \{c_1 \simeq c_2, \ldots, c_1 \simeq c_k\} \vdash_\forall x_j \simeq c_1$$

The additional rigid equations restrict the values of variables to ground Σ_2-terms of the form $f_1 f_2^{i_1} \ldots f_1 f_2^{i_p} c_q$, i.e. to the codes of Σ_n-terms. It is easy to check that S is solvable if and only if S' is solvable and that solutions for S' are exactly the codes of the corresponding solutions for S.

Thus, it remains to consider the cases $\mathrm{SREU}_0, \mathrm{SREU}_1, \mathrm{SREU}_2$.

For SREU_0, simultaneous rigid E-unification is NP-complete (Degtyarev, Matiyasevich and Voronkov [11]). There is a simple non-deterministic polynomial time algorithm for finding a solution σ to any SREU_0 system S of rigid equations. Just guess, for each variable x in S, a constant $x\sigma$ among the constants of S; this gives you a candidate σ. (If there are no constants in S, then S is trivially solvable.) Then verify the ground system of rigid equations $S\sigma$ using a known polynomial time algorithm (a congruence closure algorithm), e.g. the one of Kozen [34].

To demonstrate the NP-hardness of SREU_0, we reduce the well-known SAT problem to SREU_0. Let C be a set of propositional clauses with variables $x_1, \ldots, x_n$. Without loss of generality we may assume that Σ_0 contains constants *true* and *false*. Define, for any literal L, Σ_0-equations L^+ and L^- as follows:

$$\begin{aligned} x_i^+ &\rightleftharpoons x_i \simeq \mathit{true}; \\ x_i^- &\rightleftharpoons x_i \simeq \mathit{false}; \\ (\neg x_i)^+ &\rightleftharpoons x_i \simeq \mathit{false}; \\ (\neg x_i)^- &\rightleftharpoons x_i \simeq \mathit{true}. \end{aligned}$$

For any clause $c = (L_1 \vee \ldots \vee L_m)$, define a rigid equation $R(c) \rightleftharpoons (L_1^-, \ldots, L_{m-1}^- \vdash_\forall L_m^+)$. For example, if c is $x_4 \vee \neg x_3 \vee \neg x_1$, then $R(c)$ is $x_4 \simeq \mathit{false}, x_3 \simeq \mathit{true} \vdash_\forall x_1 \simeq \mathit{false}$. It is evident that σ is a solution to $\{R(c) \mid c \in C\}$ if and only if σ satisfies all clauses in C.

It has been noted in Degtyarev, Matiyasevich and Voronkov [10, 11] that the famous *word equation problem* has a simple reduction to SREU_2. The study of word equations (also called *equations in a free semigroup* or *unification under associativity*) was initiated by Markov at the end of the 1950s in connection with the then still unsolved Hilbert'a tenth problem [43]. The problem happened to be very hard and its decidability has been proven only in 1977 by Makanin [38]. The precise definition may be found e.g. in [39, 40], we shall give here an informal definition of the word equation problem.

Given an equality $V \simeq W$ of words V, W in the alphabet $\{f_1, \ldots, f_n\} \cup \{x_1, \ldots, x_m\}$, is there a substitution $\sigma = \{U_1/x_1, \ldots, U_m/x_m\}$ of words for variables such that $U_1, \ldots, U_m$ are words in the alphabet $\{f_1, \ldots, f_n\}$ and $U\sigma = V\sigma$? For example, the word equation $f_2 f_3 x f_3 \simeq y f_1 y$ has solutions

$$\{f_1 f_2/x, f_2 f_3/y\}$$
$$\{f_2 f_3 f_1 f_2 f_3 f_2/x, f_2 f_3 f_2 f_3/y\}$$

Hmelevskiĭ noticed that a finite system of word equations reduces to a single word equation [29].

The reduction of word equations to monadic simultaneous rigid E-unification is similar to the representation of second-order unification, but is restricted to monadic signatures. We shall use the symbols $f_1, \ldots, f_n$ both as elements of the original alphabet to represent words and as unary function symbols to write terms.

Here we only show how to represent the concatenation of words. Suppose that x_i, x_j and x_k are different variables. We define the system of rigid equations R in the following way: R consists of all rigid equations expressing that x_l is a ground term in the signature $\{f_1, \ldots, f_n, c_l\}$, for all $l \in \{i, j, k\}$ and, in addition, the rigid equation $c_i \simeq x_j, c_j \simeq c_k \vdash_\forall x_i \simeq x_k$. By routine inspection, we can check that the solutions to R are substitutions of the form $\{Vc_i/x_i, Wc_j/x_j, VWc_k/x_k\}$, where V, W are arbitrary words in $f_1, \ldots, f_n$.

This theorem implies that there is hardly a simple decidability proof for SREU_2 because solvability of the word equations is a very hard combinatorial problem (a mathematically dense proof in Makanin [38] occupies 88 journal pages). No interesting upper bounds for complexity of this problem are yet known[2]. It is only known that the problem is NP-hard (see e.g. Benanav, D. Kapur and P. Narendran [5]). Makanin's algorithm [38] leads to a tower of several exponentials (see Kościelski and Pacholski [33]).

In addition to word equations, some other relations on words can be represented by monadic simultaneous rigid E-unification. For example, suppose that R is a system of rigid equations expressing that x and y have forms Wc and Vc, where W, V are arbitrary words and c is a constant. Extend R by the rigid equation $x \simeq c \vdash_\forall y \simeq c$, obtaining a system R'. One can see that the solutions of R' are those substitutions σ for which $x\sigma = Wc$ and $y\sigma = W^m c$ for some word W and natural number m. Thus we can express that a word V is an exponent W^m of the word W. In [26] Gurevich and Voronkov show the equivalence of SREU_2 to an extension of word equations with some extra relations.

Now, let us consider SREU_1. The connection of word equations and SREU_2 remains valid in this case but now we are talking about word equations restricted over a one-letter alphabet $\{s\}$. Representing a natural number m by the word s^m, we immediately obtain that SREU_1 can express the addition $k + m$ of numbers (corresponding to the concatenation of $s^k s^m$) and the divisibility predicate (if $s^k = (s^l)^m$ for some m then k divides l, denoted $k \mid l$).

The decision problem of satisfiability of conjunctions of atomic statements in the language $\{0, 1, +, |, \simeq\}$ over natural numbers is known as the *Diophantine problem for addition and divisibility.* This problem has been shown decidable in Bel'tyukov [4], Mart'yanov [42] and Lipshitz [37]. Thus, there is a reduction of the Diophantine problem for addition and divisibility to SREU_1.

Degtyarev, Matiyasevich and Voronkov [11] give the converse reduction of SREU_1 to the Diophantine problem for addition and divisibility, thus establishing the decidability of SREU_1. This reduction is not easy to explain here in detail, so we only describe some more essential ideas. Since every ground term in Σ_1 has the form $s^n c$ for some natural number n and constant c, it suffices to seek solutions among substitutions θ such that each $x_i\theta$ has the form $s^{n_i} c_i$. The algorithm of

[2] A.Kościelski, G.Makanin, L.Pacholski, K.Schulz, J.Siekmann (private communications).

[11] tries to find appropriate n_i and c_i for every such variable x_i. Since only a finite number of constants occur in the system of equations, constants c_i can be "guessed" in the beginning. Then the problem reduces to finding the numbers n_i. The algorithm of [11] "simplifies" rigid equations constructing some conditions on n_i. We consider one case which illustrates how the divisibility predicate appears in the conditions. For example, suppose that some rigid equations has the form $\{s^{n_i}c \simeq c, s^{n_j}c \simeq c\} \cup R \vdash_\forall t_1 \simeq t_2$. One can notice that for all numbers n_i, n_j, the system of two equations $\{s^{n_i}c \simeq c, s^{n_j}c \simeq c\}$ is equivalent to one equation $\{s^{\mathrm{gcd}(n_i,n_j)}c \simeq c\}$, where gcd means the greatest common divisor. Thus, the original rigid equation $\{s^{n_i}c \simeq c, s^{n_j}c \simeq c\} \cup R \vdash_\forall t_1 \simeq t_2$ can be replaced by a "simpler" rigid equation $\{s^k c \simeq c\} \cup R \vdash_\forall t_1 \simeq t_2$ if we add the condition $k = \mathrm{gcd}(n_i, n_j)$. From elementary number-theoretic facts it follows that gcd can be represented via addition and divisibility: $k = \mathrm{gcd}(l, m)$ if and only if $k \mid l \wedge k \mid m \wedge \exists u \exists v(l \mid u \wedge m \mid v \wedge u + k = v)$. Using similar "simplification" rules, every system of rigid equations can be reduced to a disjunction of instances of the Diophantine problem for addition and divisibility. This yields

Theorem 6 SREU_1 is decidable.

This also implies the decidability of Problems 1–7 in the signature Σ_1. The complexity of SREU_1 is investigated in Gurevich and Voronkov [27].

Acknowledgments

We thank Faron Moller for his remarks on the preliminary version of this article. We acknowledge the Faculty of Science and Technology of Uppsala University for providing funds for the visit of the second author to Uppsala University in 1996.

Bibliography

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

[2] M. Baaz. Note on the existence of most general semi-unifiers. In *Arithmetic, Proof Theory and Computation Complexity*, volume 23 of *Oxford Logic Guides*, pages 20–29. Oxford University Press, 1993.

[3] G. Becher and U. Petermann. Rigid unification by completion and rigid paramodulation. In B. Nebel and L. Dreschler-Fischer, editors, *KI-94: Advances in Artificial Intelligence. 18th German Annual Conference on Artificial Intelligence*, volume 861 of *Lecture Notes in Artificial Intelligence*, pages 319–330, Saarbrücken, Germany, September 1994. Springer Verlag.

[4] 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.

[5] D. Benanav, D. Kapur, and P. Narendran. Complexity of matching problems. *Journal of Symbolic Computations*, 3:203–216, 1987.

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

[7] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Springer Verlag, 1996. To appear.

[8] C.L. Chang. Theorem proving with variable-constrained resolution. *Information Sciences*, 4:217–231, 1972.

[9] A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid E-unification with one variable. UPMAIL Technical Report, Uppsala University, Computing Science Department. In preparation.

[10] A. Degtyarev, Yu. Matiyasevich, and A. Voronkov. Simultaneous rigid E-unification is not so simple. UPMAIL Technical Report 104, Uppsala University, Computing Science Department, April 1995.

[11] 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.

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

[13] 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.

[14] 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.

[15] A. Degtyarev and A. Voronkov. The undecidability of simultaneous rigid E-unification. *Theoretical Computer Science*, 166:10, 1996.

[16] A. Degtyarev and A. Voronkov. What you always wanted to know about rigid E-unification. In J.J. Alferes, L. Moniz Pereira, and E. Orlowska, editors, *Logics in Artificial Intelligence (JELIA'96)*, volume 1126 of *Lecture Notes in Artificial Intelligence*, 20 pages, Evora, Portugal, 1996.

[17] A. Degtyarev and A. Voronkov. Equality reasoning in sequent-based calculi: a tutorial. UPMAIL Technical Report, Uppsala University, Computing Science Department. To appear.

[18] E. De Kogel. Rigid E-unification simplified. In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *Theorem Proving with Analytic Tableaux and Related Methods*, number 918 in Lecture Notes in Artificial Intelligence, pages 17–30, Schloß Rheinfels, St. Goar, Germany, May 1995.

[19] 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.

[20] J.H. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Rigid E-unification is NP-complete. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, July 1988.

[21] 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.

[22] 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.

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

[24] J. Goubault. A rule-based algorithm for rigid E-unification. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Computational Logic and Proof Theory. Proceedings of the Third Kurt Gödel Colloquium, KGC'93*, volume 713 of *Lecture Notes in Computer Science*, pages 202–210, Brno, August 1993.

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

[26] Y. Gurevich and A. Voronkov. The monadic case of simultaneous rigid E-unification. Upmail technical report, Uppsala University, Computing Science Department, 1996. In preparation.

[27] Y. Gurevich and A. Voronkov. Simultaneous rigid E-unification in the case of one unary function. Upmail technical report, Uppsala University, Computing Science Department, 1996. In preparation.

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

[29] Ju.I. Hmelevskiǐ. Equations in free semigroups (in Russian). *Dokl. Akad. Nauk SSSR*, 156:749–751, 1964. English translation in Soviet Math. Dokl. 5 (1964).

[30] J.M. Hullot. Canonical forms and unification. In *5th CADE*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334, 1980.

[31] S. Kanger. *Provability in Logic*, volume 1 of *Studies in Philosophy*. Almquist and Wicksell, Stockholm, 1957.

[32] 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.

[33] A. Kościelski and L. Pacholski. Complexity of unification in free groups and free semigroups. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 824–829, Los Alamitos, 1990.

[34] 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.

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

[36] R.C.T. Lee and C.L. Chang. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

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

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

[39] G.S. Makanin. Equations in a free semigroup. *American Mathematical Society Translations*, 117:1–6, 1981.

[40] G.S. Makanin. Investigations on equations in a free group. In K.U. Schulz, editor, *Word Equations and Related Topics*, volume 572 of *Lecture Notes in Computer Science*, pages 1–12, Tübingen, Germany, October 1990.

[41] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, 1982.

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

[43] Yu.V. Matiyasevič. A connection between systems of word and length equations and Hilbert's tenth problem (in Russian). *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 8:132–144, 1968. English Translation in: Sem. Math. V.A. Steklov Math. Inst. Leningrad 8 (1968).

[44] M. Paterson and M. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16:158–167, 1978.

[45] 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.

[46] 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.

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

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

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

[50] A. Voronkov. Proof search in intuitionistic logic with equality, or back to simultaneous rigid E-unification. In M.A. McRobbie and J.K. Slaney, editors, *Automated Deduction — CADE-13*, volume 1104 of *Lecture Notes in Computer Science*, pages 32–46, New Brunswick, NJ, USA, 1996.