Monadic Simultaneous Rigid E-Unification and Related Problems

Yuri Gurevich[1]* and Andrei Voronkov[2]**

[1] EECS Department
University of Michigan
Ann Arbor, MI, 48109-2122, USA
[2] Computing Science Department, Uppsala University
Box 311, S-751 05 Uppsala, Sweden

Abstract. We study the monadic case of a decision problem know as simultaneous rigid E-unification. We show its equivalence to an extension of word equations. We prove decidability and complexity results for special cases of this problem.

1 Introduction

Simultaneous rigid E-unification is a combinatorial problem in equational logic which is closely connected with some formulations of the Herbrand theorem and with automated theorem proving by the tableau method and the connection (or mating) method. In this section we define simultaneous rigid E-unification, discuss its connection with several decision problems in logic and survey some known results.

We shall consider *equational logic,* i.e. logic whose only predicate is the equality predicate $\simeq$. Let $s_1, t_1, \ldots, s_n, t_n, s, t$ be terms. All atomic formulas in equational logic are *equations,* i.e. expressions of the form $s \simeq t$. We do not distinguish an equation $s \simeq t$ from the equation $t \simeq s$. We write $s_1 \simeq t_1, \ldots, s_n \simeq t_n \vdash s \simeq t$ to denote that the formula $\forall(s_1{\simeq}t_1 \wedge \ldots \wedge s_n{\simeq}t_n \supset s{\simeq}t)$ is true, i.e. it is provable in first-order (classical or intuitionistic) logic. Equivalently, we can say that s and t lie in the same class of the congruence induced by $\{s_1{\simeq}t_1, \ldots, s_n{\simeq}t_n\}$.

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

* Partially supported by grants from NSF, ONR and the Faculty of Science and Technology of Uppsala University.
** Supported by a TFR grant.

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

In [2] it is shown that the decidability of SREU is equivalent to the decidability of some other fundamental problems, for example the decidability of the prenex fragment of intuitionistic logic with equality. We refer to [2, 6] for the discussion of these problems.

Best known (un)decidability results on SREU are the following: (i) SREU with ground left-hand sides, two variables and three rigid equation is undecidable (Veanes [16]); (ii) SREU with one variable is DEXPTIME-complete (Degtyarev, Gurevich, Narendran, Veanes and Voronkov [3]). The last two results imply a complete classification of decidable prenex fragments of intuitionistic predicate calculus with equality: the $\exists\exists$ fragment is undecidable and the $\forall^*\exists\forall^*$ fragment is decidable. All the above mentioned undecidability results require that the signature contain a function symbol of arity ≥ 2.

The special case of SREU when all function symbols have arity ≤ 1, is called *monadic SREU.* The decidability of monadic SREU is an open problem. The following facts are known about monadic SREU (Degtyarev, Matiyasevich and Voronkov [4]).

- The word equation problem is effectively reducible to monadic SREU. (This fact shows that if this problem is decidable, its decidability should be uneasy to prove.)
- Monadic SREU with one function symbol is decidable (this fact has a non-elementary proof).
- Monadic SREU is decidable if and only if it is decidable in the signature with two function symbols.

This paper studies monadic SREU. Although the general case remains an open problem, we prove its equivalence to a combinatorial problem of words defined in Section 5. This problem is defined in terms of *ideals* on the set of pairs of words and called *the ideal equation problem.* We prove

Theorem 4 Monadic SREU is decidable if and only if the ideal equation problem is decidable.

We also prove the decidability of some special cases of monadic SREU. In Section 4 we prove a result similar to the main result of [3]:

Theorem 3 Monadic SREU with one variable is PSPACE-complete.

Plaisted [13] proved that SREU with ground left-hand sides is undecidable. The corresponding monadic case is shown to be decidable in Section 3:

Theorem 2 Monadic SREU with ground left-hand sides is decidable.

The complexity of monadic SREU with ground left-hand sides is not known. We prove

Theorem 1 Monadic SREU with one variable and ground left-hand sides is PSPACE-hard.

2 Preliminaries

In this section we introduce basic definitions concerning terms, equations, words, word equations, automata and rewrite rules. We have to define so many concepts since it is unreasonable to expect the reader to know everything. We also assert some statements proved elsewhere and prove some properties of the introduced notions which will be used in subsequent sections.

The symbol $\rightleftharpoons$ means "equal by definition".

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

For any expression E (for example, term, or a set of equations), we denote by E_c^t the expressions obtained from E by the replacement of all occurrences of the constant c by a term t. We write $s[t]$ to denote a particular occurrence of a subterm t of a term s.

In this paper, we shall only consider *monadic signatures* consisting of a finite set $\mathcal{F}$ of unary function symbols and a finite set $\mathcal{C}$ of constants. Such signatures are denoted $(\mathcal{F}, \mathcal{C})$. The set of ground terms of this signature is denoted by $T_{(\mathcal{F},\mathcal{C})}$. We always assume $\mathcal{C} \neq \emptyset$ and hence $T_{(\mathcal{F},\mathcal{C})} \neq \emptyset$. For any set of equations $\mathcal{E}$ we denote by $T(\mathcal{E})$ the set of all terms occurring in $\mathcal{E}$ and their subterms. For example, if $\mathcal{E} = \{f(x) \simeq g(c), c \simeq g(f(x))\}$, then $T(\mathcal{E}) = \{x, f(x), c, g(c), g(f(x))\}$.

We shall denote variables by x, y, z, constants by a, b, c, d, function symbols by f, g, h, terms by r, s, t and substitutions by θ.

We shall use the following statement proved in Kozen [9] or Shostak [15].

Lemma 1 (Derivability of equations is in PTIME) *There is a polynomial-time algorithm checking, by a given finite set of equations $\mathcal{E}$ and terms s, t, whether $\mathcal{E} \vdash s \simeq t$.*

We write $\mathcal{E}' \vdash \mathcal{E}$ iff for any equation $(s \simeq t) \in \mathcal{E}$ we have $\mathcal{E}' \vdash s \simeq t$. In the sequel we shall use the following lemma whose proof is standard.

Lemma 2 (Lemma on constants) *Let $\mathcal{E}$ and $\mathcal{E}'$ be sets of equations. For any constant c and term t, if $\mathcal{E} \vdash \mathcal{E}'$, then $\mathcal{E}_c^t \vdash {\mathcal{E}'}_c^t$.*

Words and finite automata. This section defines *words* and *finite automata.* We shall also introduce a notation for monadic terms which allows us to easily come from terms to words and back.

Let $\mathcal{F}$ be a finite non-empty set, called the *alphabet.* Its elements are called *letters. Words* are finite sequences of letters. We denote words by a juxtaposition of its letters, as $W = a_1 a_2 \ldots a_n$. The natural number n is called the *length* of the word W and denoted $|W|$. We denote by ε the *empty word,* which is the unique word of length zero. The set of all words with letters in $\mathcal{F}$ is denoted by $\mathcal{F}^*$.

It will be convenient for us to use the alphabet $\mathcal{F}$ also as the set of unary function symbols of a monadic signature $(\mathcal{F}, \mathcal{C})$. Every term s in such a signature has the form $f_1(f_2(\ldots f_n(t) \ldots))$ where $n \geq 0$, $f_1, \ldots, f_n$ are unary function symbols and t is a constant or a variable. We shall denote such a term s in the reversed Polish notation, i.e. as $tf_n \ldots f_2 f_1$. Thus, every term can be represented in the form tW, where t is a constant or a variable and W is a word. Similarly, any term of the form $f_1(f_2(\ldots f_n(t) \ldots))$, where t is an arbitrary term, will be written as $tf_n \ldots f_2 f_1$.

A *finite automaton* $\mathcal{A}$ on the alphabet $\mathcal{F}$ is a quadruple (Q, I, T, E), where Q is a finite set, called *the set of states,* I and T are distinguished subsets of Q, called the sets of *initial* and *terminal* states, respectively. The set $E \subseteq Q \times \mathcal{F} \times Q$ is *the set of edges of* $\mathcal{A}$. An edge (p, f, q) is also denoted $p \xrightarrow{f} q$. The automaton is *deterministic* iff whenever $(p, f, q_1) \in E$ and $(p, f, q_2) \in E$, then $q_1 = q_2$.

A word $f_1 \ldots f_n$ is *recognized by an automaton* (Q, I, T, E) iff there is a sequence of states $q_0 \ldots q_n$ such that $q_0 \in I$, $q_n \in T$ and $q_{i-1} \xrightarrow{f_i} q_i$ for all $i \in \{1, \ldots, n\}$. A set of words is *regular* iff it is the set of words recognized by some automaton.

The *intersection nonemptyness of deterministic finite automata problem* is the following decision problem. Given any finite set $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ of deterministic finite automata, is there a word recognized by each automaton in this set. The following statement is proved in Kozen [10]:

Lemma 3 *The intersection nonemptyness of deterministic finite automata problem is PSPACE-complete.*

Word equations. In addition to the alphabet $\mathcal{F}$, we shall also consider a countable set $\mathcal{V}$ of *word variables,* denoted u, v, w. A *word equation* is any expression of the form $V \simeq W$, where $V, W \in (\mathcal{F} \cup \mathcal{V})^*$. A *word substitution* is any expression $\sigma = \{V_1/v_1, \ldots, V_n/v_n\}$, where v_i are word variables and V_i are words in $\mathcal{F}^*$. Its *domain,* denoted $dom(\sigma)$ is the set $\{v_1, \ldots, v_n\}$. The *application of such a word substitution θ to a word $W \in (\mathcal{F} \cup \mathcal{V})^*$,* is the operation of simultaneous replacement of all occurrences of v_i by V_i. The result of the application is the word denoted $W\sigma$. A word substitution σ is a *solution to a word equation* $U \simeq V$ iff all variables in U, V belong to $dom(\sigma)$ and we have $U\sigma = V\sigma$. A *system of word equations* is any finite set of word equations, its *solution* is any substitution solving all equations in the system. Words will be denoted by U, V, W, word variables by u, v, w and word substitutions by ρ, σ, τ.

Makanin [11] proved that word equations are decidable. Analyzing Makanin's algorithm, Schultz [14] proves the following result.

Lemma 4 (Decidability of word equations with regular constraints) *The problem of solvability of word equations where every word variable u_i ranges over a regular set S_i, is decidable.*

It is known that the problem of solvability of word equations is NP-hard. No good upper bound for the complexity of this problem has been obtained so far, it is only known that the problem is in 3-NEXP (Kościelski and Pacholski [7, 8]).

Equational logic and rigid equations. Let $\mathcal{R}$ be a system of rigid equations. The *signature of* $\mathcal{R}$ is defined as the signature consisting of all constants and function symbols occurring in $\mathcal{R}$; and in addition a fixed constant if $\mathcal{R}$ contains no constants. A solution θ to $\mathcal{R}$ is called *grounding for* $\mathcal{R}$ iff for every variable x occurring in $\mathcal{R}$ the term $x\theta$ is ground. A substitution θ is called *relevant* for $\mathcal{R}$ iff for every variable x the term $x\theta$ is in the signature of $\mathcal{R}$.

In the sequel, we shall need the following technical property of systems of rigid equations.

Lemma 5 (Existence of relevant grounding solutions) *Let $\mathcal{R}$ be a solvable system of rigid equations. Then there exists a solution θ to $\mathcal{R}$ that is grounding and relevant for $\mathcal{R}$.*

We shall introduce one particular kind of rigid equations that will be used as a technical tool for proofs in this paper. For any monadic signature $(\mathcal{F}, \mathcal{C})$, any variable x and any constant $c \in C$ introduce the following rigid equation:

$$Gr_{(\mathcal{F},\mathcal{C})}(x) \rightleftharpoons \{d \simeq c \mid d \in \mathcal{C}\} \cup \{cf \simeq c \mid f \in \mathcal{F}\} \vdash_{\forall} x \simeq c$$

We shall use the following obvious lemma:

Lemma 6 *A substitution θ is a solution to $Gr_{(\mathcal{F},\mathcal{C})}(x)$ iff $x\theta \in T_{(\mathcal{F},\mathcal{C})}$.*

As a consequence, we have

Lemma 7 *For any system $\mathcal{R}$ of rigid equations there is a system $\mathcal{R}'$ of rigid equations such that for any substitution θ, θ is a solution to $\mathcal{R}'$ if and only if θ is a grounding relevant solution to $\mathcal{R}$. In addition, $\mathcal{R}'$ can be found by $\mathcal{R}$ using a polynomial-time algorithm; and $\mathcal{R}'$ has ground left-hand sides if $\mathcal{R}$ has ground left-hand sides.*

Proof. Let $x_1, \ldots, x_n$ be all variables in $\mathcal{R}$ and $(\mathcal{F}, \mathcal{C})$ be the signature of $\mathcal{R}$. Define $\mathcal{R}' \rightleftharpoons \mathcal{R} \cup \{Gr_{(\mathcal{F},\mathcal{C})}(x_i) \mid i \in \{1, \ldots, n\}\}$. Then apply Lemma 6.

Rewrite rules. This section introduces a technique standard in the theory of ground systems of rewrite rules. However, we shall use ordinary equations instead of rewrite rules.

Introduce an ordering $\succ$ on terms in $T_{(\mathcal{F},\mathcal{C})}$ in the following way. Let $>$ be any total ordering on $\mathcal{F} \cup \mathcal{C}$ and $s = cf_1 \ldots f_m$, $t = dg_1 \ldots g_n$. Then $s \succ t$ iff one of the following conditions is true:

1. $m > n$;
2. $m = n$ and the string $cf_1 \ldots f_m$ is greater than $dg_1 \ldots g_n$ in the lexicographic ordering induced by $>$.

The ordering $\succ$ is total, noetherian and can be extended to a simplification ordering [1]. Some properties of the ordering formulated below are simple consequence of standard statements in the theory of rewrite systems. Their proofs may be found in e.g. [1]. Note that the ordering $\succ$ depends on the ordering of $>$. In the definitions below we assume that we have chosen a fixed ordering $>$ on $\mathcal{F} \cup \mathcal{C}$, and hence $\succ$ is also fixed.

Let $\mathcal{E}, \mathcal{E}'$ be finite sets of ground equations and $\mathcal{E}$ contains distinct equations $s \simeq t$ and $r[s] \simeq u$. We say that *$\mathcal{E}'$ is obtained from $\mathcal{E}$ by simplification from $s \simeq t$ into $r[s] \simeq u$,* denoted $\mathcal{E} \to \mathcal{E}'$ iff

$$\mathcal{E}' = (\mathcal{E} \setminus \{r[s] \simeq u\}) \cup \{r[t] \simeq u\}$$

The reflexive and transitive closure of the relation $\to$ on sets of ground equations is denoted by $\to^*$. A set of equations $\mathcal{E}$ is called *irreducible* iff there exists no $\mathcal{E}'$ such that $\mathcal{E} \to \mathcal{E}'$.

Let $\mathcal{E}$ be an irreducible set of ground equations. We write $t \to_{\mathcal{E}} t'$ if there exists an equation $(r \simeq s) \in \mathcal{E}$ such that $r \succ s$, and t' is obtained from t by the replacement of one occurrence of the subterm r by s. The relation $\to_{\mathcal{E}}^*$ is the reflexive and transitive closure of $\to_{\mathcal{E}}$. A term t is called *irreducible with respect to $\mathcal{E}$* iff there is no term s such that $t \to_{\mathcal{E}} s$. *The normal form of a term t w.r.t. $\mathcal{E}$,* denoted $t \downarrow_{\mathcal{E}}$, is the term s such that $t \to_{\mathcal{E}}^* s$ and s is irreducible w.r.t $\mathcal{E}$. The normal form of any term exists and is unique. We shall use the following statements which are easy to prove.

Lemma 8 *Let $\mathcal{E}$ be an irreducible set of ground equations and s, t be terms. Then $\mathcal{E} \vdash s \simeq t$ if and only if $s \downarrow_{\mathcal{E}} = t \downarrow_{\mathcal{E}}$.*

Mixing words and rigid equations. We call a *word term,* or simply *w-term,* in the signature $(\mathcal{F}, \mathcal{C})$ any expression of the form cW such that $c \in \mathcal{C}$ and $W \in (\mathcal{F} \cup \mathcal{V})^*$. A *w-equation* is any expression $cV \simeq dW$, where cV and dW are w-terms. A *rigid w-equation* is any expression of the form $\mathcal{W} \vdash_\forall cV \simeq dW$, where $\mathcal{W}$ is a finite set of w-equations, cV and dW are w-terms. A *system of rigid w-equations* is any finite set of rigid w-equations. *The signature of a system of rigid w-equations* is defined similar to that of a system of rigid equations. Sets of w-equations will be denoted by $\mathcal{W}$, and sets of rigid w-equations by $\mathcal{S}$.

A *solution to a rigid w-equation* $\mathcal{W} \vdash_\forall cV \simeq dW$ is any word substitution σ whose domain contains all word variables in $\mathcal{W}, V, W$ such that $\mathcal{W}\sigma \vdash cV\sigma \simeq dW\sigma$. A *solution to a system* $\mathcal{S}$ *of rigid w-equations* is any word substitution that is a solution to every rigid w-equation in $\mathcal{S}$.

Note that a ground w-equation is also an ordinary equation.

In Lemma 9 below we show that one can consider systems of rigid w-equations instead of systems of rigid equations. The following technical lemma is proved in [6]:

Lemma 9 *The problem of solvability of systems of rigid w-equations is polynomial-time reducible to monadic SREU. Monadic SREU is effectively reducible to the problem of solvability of systems of rigid w-equations.*

3 Ground left-hand sides

In this section we prove that monadic SREU with ground left-hand sides is decidable and PSPACE-hard.

SREU with ground left-hand sides is PSPACE-hard.

Lemma 10 *Let* $\mathcal{A} = (Q, I, T, E)$ *be a deterministic finite automaton over* $\mathcal{F}$. *There exists a system* $\mathcal{R}$ *of two monadic rigid equations of one variable* x *with the following properties:*

1. $\mathcal{R}$ *has ground left-hand sides;*
2. *for every solution* θ *to* $\mathcal{R}$ *we have* $x\theta = cW$, *where* $W \in \mathcal{F}^*$ *and* c *is a fixed constant;*
3. *for any word* $W \in \mathcal{F}$, *the substitution* $\{cW/x\}$ *is a solution to* $\mathcal{R}$ *if and only if* W *is recognized by* $\mathcal{A}$.

In addition, $\mathcal{R}$ *can be effectively constructed from* $\mathcal{A}$ *using a polynomial-time algorithm.*

Proof. Without loss of generality we can assume that I consists of one state (see e.g. [12]). By renaming states, we can assume that $I = \{c\}$. Let F be a unary function symbol fresh for $\mathcal{F}$ and d be a constant fresh for Q. Define $\mathcal{R}$ as $\{R_1, R_2\}$, where

$$R_1 = \{pf \simeq q \mid (p \xrightarrow{f} q) \in E\} \cup \{rF \simeq d \mid r \in T\} \vdash_\forall xF \simeq d$$
$$R_2 = Gr_{(\mathcal{F},\{c\})}(x)$$

Consider any substitution $\theta = \{t/x\}$. By Lemma 6, θ is a solution to R_2 if and only if t has the form cW such that $W \in \mathcal{F}^*$. Consider when such substitution $\{cW/x\}$ is also a solution to R_1. By definition, this means

$$\{pf \simeq q \mid (p \xrightarrow{f} q) \in E\} \cup \{rF \simeq d \mid r \in T\} \vdash cWF \simeq d \tag{1}$$

Since the automaton is deterministic, the left-hand side of (1) is irreducible. Using Lemma 8, one can see that (1) holds if and only if W is recognizable by $\mathcal{A}$. Evidently, $\mathcal{R}$ is constructed by $\mathcal{A}$ in polynomial time.

Lemma 11 *The intersection nonemptyness of deterministic finite automata problem is polynomial-time reducible to monadic SREU with one variable and ground left-hand sides.*

Proof. Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be deterministic finite automata. Let $\mathcal{R}_i$, where $i \in \{1, \ldots, n\}$ be the system of rigid equations constructed by $\mathcal{A}_i$ as in Lemma 10. Define $\mathcal{R} = \bigcup_{i=1}^{n} \mathcal{R}_i$. By Lemma 10, every solution to $\mathcal{R}$ has the form $\{cW/x\}$ and any substitution $\{cW/x\}$ is a solution to $\mathcal{R}$ if and only if W is recognized by each $\mathcal{A}_i$. Hence, $\mathcal{R}$ is solvable if and only if there is a word recognizable by all $\mathcal{A}_i$. Evidently, $\mathcal{R}$ is constructed by $\mathcal{A}_1, \ldots, \mathcal{A}_n$ in polynomial time.

Combining Lemmas 3 and 11 we obtain

Theorem 1 *Monadic SREU with one variable and ground left-hand sides is PSPACE-hard.*

Monadic SREU with ground left-hand sides is decidable. A finite set $\mathcal{E}$ of equations is *in the automaton form* iff

1. every equation in $\mathcal{E}$ has the form $cf \simeq d$;
2. for every two w-equations $cf \simeq d_1$ and $cf \simeq d_2$ in $\mathcal{E}$ we have $d_1 = d_2$;

Note that any set of equations in the automaton form is irreducible. The following statement is proved in [6]:

Lemma 12 *Given any rigid w-equation S with ground left-hand side, one can effectively find in polynomial time a rigid w-equation S' with ground left-hand side such that*

1. *S and S' have the same solutions;*
2. *the left-hand side of S' is in the automaton form.*

Let $\mathcal{E}$ be a set of equations in the automaton form and c, d be any constants. Denote by $\mathcal{A}(\mathcal{E}, c, d)$ the following automaton (Q, I, T, E). Its alphabet is the set of function symbols occurring in $\mathcal{E}$. The set of states Q is the set of all constants occurring in $\mathcal{E}, c, d$. The sets of initial states and terminal states are defined by $I \rightleftharpoons \{c\}$ and $T \rightleftharpoons \{d\}$. Finally, the set of edges is defined by

$$E \rightleftharpoons \{a \xrightarrow{f} b \mid (af \simeq b) \in \mathcal{E}\}.$$

Lemma 13 *A word W is recognized by $\mathcal{A}(\mathcal{E}, c, d)$ if and only if $\mathcal{E} \vdash cW \simeq d$.*

Proof. Immediate by Lemma 8.

Lemma 14 *Let $\mathcal{E}$ be a set of equations in the automaton form, $W, W' \in \mathcal{F}^*$ and c, c' be constants. Then $\mathcal{E} \vdash cW \simeq c'W'$ if and only if there is a constant d and words U, U', V such that $W = UV$, $W' = U'V$, U is recognized by $\mathcal{A}(\mathcal{E}, c, d)$ and U' is recognized by $\mathcal{A}(\mathcal{E}, c', d)$.*

Proof.

($\Rightarrow$) We have $\mathcal{E} \vdash cW \simeq c'W'$. By Lemma 8 we have $cW \downarrow_{\mathcal{E}} = c'W' \downarrow_{\mathcal{E}}$. Choose d and V such that $cW \downarrow_{\mathcal{E}} = dV$. Define U and U' such that $W = UV$ and $W' = U'V$. We have $\mathcal{E} \vdash cU \simeq d$ and $\mathcal{E} \vdash c'U' \simeq d$. By Lemma 13 words U and U' are recognized by $\mathcal{A}(\mathcal{E}, c, d)$ and $\mathcal{A}(\mathcal{E}, c', d)$, respectively.

($\Leftarrow$) We have $W = UV$, $W' = U'V$, U is recognized by $\mathcal{A}(\mathcal{E}, c, d)$ and U' is recognized by $\mathcal{A}(\mathcal{E}, c', d)$. By Lemma 13 we have $\mathcal{E} \vdash cU \simeq d$ and $\mathcal{E} \vdash c'U' \simeq d$. Hence, $\mathcal{E} \vdash cUV \simeq dV$ and $\mathcal{E} \vdash c'U'V \simeq dV$. Then $\mathcal{E} \vdash cUV \simeq c'U'V$, i.e. $\mathcal{E} \vdash cW \simeq c'W'$. □

Lemma 15 *The problem of solvability of systems of rigid w-equations with ground left-hand sides effectively reduces to word equations with regular constraints.*

Proof. Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ be such a system of rigid w-equations. By Lemma 12 we can assume that the left-hand sides of all S_i are in the automaton form. Let $S_i = (\mathcal{E}_i \vdash_\forall c_i W_i \simeq c_i' W_i')$, for all $i \in \{1, \ldots, n\}$. Let $u_1, \ldots, u_n$, $v_1, \ldots, v_n$ and $u_1', \ldots, u_n'$ be word variables fresh for $\mathcal{S}$. By Lemma 14, the system $\mathcal{S}$ is solvable if and only if there are constants d_i occurring in S_i, for all $i \in \{1, \ldots, n\}$ such that the following system of word equations and regular constraints is solvable:

$$\begin{array}{ll}
W_1 \simeq u_1 v_1 & u_1 \text{ is recognized by } \mathcal{A}(\mathcal{E}_1, c_1, d_1) \\
\ldots & \ldots \\
W_n \simeq u_n v_n & u_n \text{ is recognized by } \mathcal{A}(\mathcal{E}_n, c_n, d_n) \\
W_1' \simeq u_1' v_1 & u_1' \text{ is recognized by } \mathcal{A}(\mathcal{E}_1, c_1', d_1) \\
\ldots & \ldots \\
W_n' \simeq u_n' v_n & u_n' \text{ is recognized by } \mathcal{A}(\mathcal{E}_n, c_n', d_n)
\end{array}$$

To conclude the proof we note that there is only a finite number of choices for d_i.

Theorem 2 *Monadic SREU with ground left-hand sides is decidable.*

Proof. By Lemma 9 monadic SREU with ground left-hand sides is effectively reducible to the problem of solvability of systems of rigid w-equations. By Lemma 15 the latter problem is effectively reducible to word equations with regular constraints. Then apply Lemma 4.

4 One-variable case

In this section we consider rigid equations with one variable x. We shall write $\mathcal{E}(x)$ to denote all occurrences of a variable x in $\mathcal{E}$, and write $\mathcal{E}(t)$ to denote the set of equations obtained from $\mathcal{E}$ by replacement of all occurrences of x by t. We shall use similar notation for terms, for example $s(x)$. Using this notation, we can write any rigid equation of one variable x as $\mathcal{E}(x) \vdash_\forall s(x) \simeq t(x)$. The following statement is proved in [6]:

Lemma 16 *Let $\mathcal{E}(x)$ be a finite set of equations of one variable x and $s(x), t(x)$ be terms of one variable x such that $\mathcal{E}(x) \not\vdash s(x) \simeq t(x)$. Let c be a constant fresh for $\mathcal{E}(x), s(x), t(x)$ and r be a ground term such that c does not occur in r. If $\mathcal{E}(r) \vdash s(r) \simeq t(r)$, then there exists a ground term $r' \in T(\mathcal{E}(c) \cup \{s(c) \simeq t(c)\})$ such that $\mathcal{E}(c) \vdash r \simeq r'$.*

Lemma 17 *Let $\mathcal{E}(x) \vdash_\forall s(x) \simeq t(x)$ be a rigid equation of one variable x, c be a constant fresh for this rigid equation, r be a ground term in which c does not occur and $\mathcal{E}(x) \not\vdash s(x) \simeq t(x)$. Then the substitution $\theta = \{r/x\}$ is a solution to this rigid equation if and only if there is a ground term $r' \in T(\mathcal{E}(c) \cup \{s(c) \simeq t(c)\})$ such that $\mathcal{E}(c), \mathcal{E}(r') \vdash s(r') \simeq t(r')$ and θ is a solution to $\mathcal{E}(c) \vdash_\forall r' \simeq x$.*

Proof.

- $\Rightarrow$ We have that θ is a solution to $\mathcal{E}(x) \vdash_\forall s(x) \simeq t(x)$. Then $\mathcal{E}(r) \vdash s(r) \simeq t(r)$. By Lemma 16 there is a term $r' \in T(\mathcal{E}(c) \cup \{s(c) \simeq t(c)\})$ such that $\mathcal{E}(c) \vdash r \simeq r'$. Then $\mathcal{E}(r), \mathcal{E}(c) \vdash s(r') \simeq t(r')$.
- $\Leftarrow$ We have $\mathcal{E}(c), \mathcal{E}(r') \vdash s(r') \simeq t(r')$ and $\mathcal{E}(c) \vdash_\forall r' \simeq r$. Then $\mathcal{E}(c), \mathcal{E}(r) \vdash s(r) \simeq t(r)$. By Lemma 2 we can substitute r for c obtaining $\mathcal{E}(r) \vdash s(r) \simeq t(r)$. □

Lemmas 16 and 17 also hold for non-monadic signatures [3].

Lemma 18 *Monadic SREU with one variable is in PSPACE.*

Proof. We shall give a non-deterministic algorithm reducing monadic SREU with one variable to the intersection nonemptyness of deterministic finite automata problem.

Let $\mathcal{R}$ be a system of rigid equations of one variable x whose signature is $(\mathcal{F}, \mathcal{C})$. It has the form

$$\mathcal{E}_1 \vdash_\forall s_1(x) \simeq t_1(x) \quad \cdots \quad \mathcal{E}_n \vdash_\forall s_n(x) \simeq t_n(x)$$

By Lemma 5 we can restrict ourselves to relevant grounding solutions $\theta = \{r/x\}$ only. Let c be a variable fresh for $(\mathcal{F}, \mathcal{C})$. By Lemma 17 θ is a solution to $\mathcal{R}$ if and only if there are ground terms $r'_i \in T(\mathcal{E}_i(c) \cup \{s_i(c) \simeq t_i(c)\})$, where $i \in \{1, \ldots, n\}$ such that $\mathcal{E}(c), \mathcal{E}(r') \vdash s(r') \simeq t(r')$ and θ is a solution to the system

$$\mathcal{E}_1(c) \vdash_\forall r'_1 \simeq x \quad \cdots \quad \mathcal{E}_n(c) \vdash_\forall r'_n \simeq x$$

Nondeterministically select such $r'_1, \ldots, r'_n$ and verify the condition $\mathcal{E}(c), \mathcal{E}(r') \vdash s(r') \simeq t(r')$ (it can be checked in polynomial time using Lemma 1).

Such θ is a solution to this system of rigid equations if and only if there is a constant $d \in \mathcal{C}$ such that the following system of rigid w-equations is solvable:

$$\mathcal{E}_1(c) \vdash_\forall r'_1 \simeq dx \quad \cdots \quad \mathcal{E}_n(c) \vdash_\forall r'_n \simeq dx$$

Nondeterministically select such d. By Lemma 12 we can equivalently replace this system with a system

$$\mathcal{E}'_1 \vdash_\forall c_1 \simeq d_1 x \quad \cdots \quad \mathcal{E}'_n \vdash_\forall c_n \simeq d_n x$$

where $\mathcal{E}'_i$ are in the automaton form. By Lemma 13, this system is solvable if and only if the intersection of automata $\mathcal{A}(\mathcal{E}'_1, d_1, c_1), \ldots, \mathcal{A}(\mathcal{E}'_n, d_n, c_n)$ is non-empty.

We have given a non-deterministic algorithm reducing monadic SREU with one variable to the intersection nonemptyness of deterministic finite automata problem. On each branch, the algorithm makes polynomially many steps. Applying Lemma 3 on the complexity of the intersection nonemptyness of deterministic finite automata problem we get that monadic SREU with one variable is in NPSPACE, and hence in PSPACE.

Combining Theorem 1 and Lemma 18, we obtain

Theorem 3 *Monadic SREU with one variable is PSPACE-complete.*

5 General case

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

$$(U_1, U_2) * (V_1, V_2) \rightleftharpoons \begin{cases} (U_2 W, V_2) \text{ if } V_1 \text{ has the form } U_1 W \\ (V_1, V_2) \quad \text{otherwise} \end{cases}$$

$$(U_1, U_2)^r \rightleftharpoons (U_2, U_1)$$

$$(U_1, U_2) \leq (V_1, V_2) \rightleftharpoons \text{there is a word } W \text{ such that } (V_1, V_2) = (U_1 W, U_2 W)$$

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

An ideal equation is an expression

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

where $n \geq 0$ and $U, V, U_1, \ldots, U_n, V_1, \ldots, V_n \in (\mathcal{F} \cup \mathcal{V})^*$. A *solution to such ideal equation* is any word substitution σ such that

1. words $U\sigma, V\sigma, U_1\sigma, \ldots, U_n\sigma, V_1\sigma, \ldots, V_n\sigma$ are words over $\mathcal{F}$;
2. the word $(U\sigma, V\sigma)$ belongs to the ideal generated by

$$\{(U_1\sigma, V_1\sigma), \ldots, (U_n\sigma, V_n\sigma)\}.$$

A *system of ideal equations* is any finite set of ideal equations. *Solutions to a system of ideal equations* are substitutions that solve each equation in the system. The *ideal equations problem* is the decision problem of solvability of systems of ideal equations. The aim of this section is to show that monadic SREU is equivalent to the ideal equations problem.

The following lemma proved in [6] is the main reason for introducing the notion of an ideal.

Lemma 19 *Let $U_1, \ldots, U_n, V_1, \ldots, V_n, U, V$ be words on $\mathcal{F}$ and a be any constant. Then $aU_1 \simeq aV_1, \ldots, aU_n \simeq aV_n \vdash aU \simeq aV$ if and only if $(U, V) \in$ ideal$(\{(U_1, V_1), \ldots, (U_n, V_n)\})$.*

Theorem 4 *Monadic SREU is decidable if and only if the ideal equation problem is decidable.*

Proof. See [6].

Technical report [6] discusses ideal equations in more detail. In particular, it is shown that ideal equations are decidable if and only if word equations extended by a family of predicates behaving like a greatest common divisor on word are decidable. In addition, the following statement is proved:

Lemma 20 *Ideal equations are decidable if and only if ideal equations with regular constraints and the inequality constraints $U \not\simeq V$ are decidable.*

Acknowledgments. We thank Anatoli Degtyarev and Gennadi Makanin.

References

1. 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.
2. A. Degtyarev, Yu. Gurevich, and A. Voronkov. Herbrand's theorem and equational reasoning: Problems and solutions. In *Bulletin of the European Association for Theoretical Computer Science*, volume 60, page ??? October 1996. The "Logic in Computer Science" column.
3. A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid E-unification with one variable. UPMAIL Technical Report 139, Uppsala University, Computing Science Department, March 1997.
4. 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.
5. A. Degtyarev and A. Voronkov. The undecidability of simultaneous rigid E-unification. *Theoretical Computer Science*, 166(1–2):291–300, 1996.
6. Yu. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. UPMAIL Technical Report 137, Uppsala University, Computing Science Department, February 1997.
7. 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.
8. A. Kościelski and L. Pacholski. Complexity of Makanin's algorithm. *Journal of the Association for Computing Machinery*, 43(4):670–684, 1996.
9. 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.
10. D. Kozen. Lower bounds for natural proof systems. In *Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 254–266, 1977.
11. 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.
12. D. Perrin. Finite automata. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 1, pages 1–57. Elsevier Science, Amsterdam, 1990.
13. 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.
14. K.U. Schulz. Makanin's algorithm: Two improvements and a generalization. In K.U. Schulz, editor, *Word Equations and Related Topics*, volume 572 of *Lecture Notes in Computer Science*, Tübingen, Germany, October 1990.
15. R. Shostak. An algorithm for reasoning about equality. *Communications of the ACM*, 21:583–585, July 1978.
16. M. Veanes. Uniform representation of recursively enumerable sets with simultaneous rigid E-unification. UPMAIL Technical Report 126, Uppsala University, Computing Science Department, 1996.