I N F O R M A T I K

Rigid Reachability

Harald Ganzinger
Florent Jacquemard
Margus Veanes

MPI–I–98–2–013 September 1998

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Im Stadtwald 66123 Saarbrücken Germany

Authors' Addresses

Harald Ganzinger
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
hg@mpi-sb.mpg.de

Florent Jacquemard
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
florent@mpi-sb.mpg.de

Margus Veanes
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
veanes@mpi-sb.mpg.de

Publication Notes

A short version of this paper appears in the proceedings of the fourth Asian Computing Science Conference, ASIAN'98.

Acknowledgements

The authors wish to thank their reviewers at the fourth Asian Computing Science Conference whose remarks led to significant improvements of the conference paper version of this report and finally this report itself.

Abstract

We show that rigid reachability, the non-symmetric form of rigid E-unification, is undecidable already in the case of a single constraint. From this we infer the undecidability of a new rather restricted kind of second-order unification. We also show that certain decidable subclasses of the problem which are $\mathcal{P}$-complete in the equational case become EXPTIME-complete when symmetry is absent. By applying automata-theoretic methods, simultaneous monadic rigid reachability with ground rules is shown to be in EXPTIME.

Keywords

Rigid Unification, Second Order Unification, Term Rewriting, Reachability.

1 Introduction

Rigid reachability is the problem, given a rewrite system R and two terms s and t, whether there exists a ground substitution σ such that $s\sigma$ rewrites in some number of steps via $R\sigma$ into $t\sigma$. The term "rigid" stems from the fact that for no rule more than one instance can be used in the rewriting process. Simultaneous rigid reachability is the problem in which a substitution is sought which simultaneously solves each member of a system of reachability constraints (R_i, s_i, t_i). A special case of [simultaneous] rigid reachability arises when the R_i are symmetric, containing for each rule $l \to r$ also its converse $r \to l$. The latter problem was introduced in (Gallier, Raatz & Snyder 1987) as "simultaneous rigid E-unification". (Symmetric systems R arise, for instance, from orienting a given set of equations E in both directions.) It has been shown in (Degtyarev & Voronkov 1995) that simultaneous rigid E-unification is undecidable, whereas the non-simultaneous case with just one rigid equation to solve is NP-complete (Gallier, Narendran, Plaisted & Snyder 1988). The main result in this paper is that for non-symmetric rigid reachability already the case of a single reachability constraint is undecidable, even when the rule set is ground. From this we infer undecidability of a rather restricted form of second-order unification for problems which contain just a single second-order variable which, in addition, occurs at most twice in the unification problem. The latter result contrasts a statement in (Levy 1998) to the opposite.

The absence of symmetry makes the problem much more difficult. This phenomenon is also observed in decidable cases which we investigate in the second part of the paper. For instance we prove that a certain class of rigid problems which is $\mathcal{P}$-complete in the equational case becomes EXPTIME-complete when symmetry is absent. Our results demonstrate a very thin borderline between the decidable and the undecidable fragments of rigid reachability with respect to several syntactical criteria. In particular, for ground R and variable-disjoint s and t, the problem is undecidable, whereas it becomes *decidable* when, in addition, either s or t is linear.

In the Section 6 we will apply automata-theoretic methods to the monadic case and establish an EXPTIME upper bound for monadic simultaneous rigid reachability for ground rewrite systems. This generalizes the analogous result of (Gurevich & Voronkov 1997) for simultaneous rigid E-unification. Also, our proof is more direct and provides a better upper bound, closer to the PSPACE lower bound given in (Gurevich & Voronkov 1997). A PSPACE upper bound for this problem has been proved more recently in a joint work with Cortier (Cortier 1998).

2 Preliminaries

A *signature* Σ is a collection of *function symbols* with fixed arities ≥ 0 and, unless otherwise stated, Σ is assumed to contain at least one *constant*, that is, one function symbol with arity 0. The set of all constants in Σ is denoted by $Con(\Sigma)$. We use $a, b, c, d, a_1, \ldots$ for constants and $f, g, f_1, \ldots$ for function symbols in general. A designated constant in Σ is denoted by c_Σ.

A *term language* or simply *language* is a triple $L = (\Sigma_L, \mathcal{X}_L, \mathcal{F}_L)$ where (i) Σ_L is a signature, (ii) $\mathcal{X}_L$ $(x, y, x_1, y_1, \ldots)$ is a collection of first-order variables, and (iii) $\mathcal{F}_L$ $(F, G, F_1, F', \ldots)$ is a collection of symbols with fixed arities ≥ 1, called *second-order variables*. The various sets of symbols are assumed to be pairwise disjoint. Let L be a language. L is *first-order*, if $\mathcal{F}_L$ is empty; L is *second-order*, otherwise. L is *monadic* if all function symbols in Σ_L have arity ≤ 1. The set of all terms in a language L, or L-*terms*, is denoted by $\mathcal{T}_L$. We use $s, t, l, r, s_1, \ldots$ for terms. We usually omit mentioning L when it is clear from the context. The set of first-order variables of a term t is denoted by $Var(t)$. A *ground* term is one that contains no variables. The set of all ground terms in a language L is denoted by $\mathcal{T}_{\Sigma_L}$. A term is called *shallow* if all variables in it occur at depth ≤ 1. The size $\|t\|$ of a term t is defined recursively by: $\|t\| = 1$ if $t \in \mathcal{X}_L \cup Con(\Sigma_L)$ and $\|f(t_1, \ldots, t_n)\| = \|t_1\| + \ldots + \|t_n\| + 1$ when $f \in \Sigma_L \cup \mathcal{F}_L$.

We assume that the reader is familiar with the basic concepts in term rewriting (Dershowitz & Jouannaud 1990, Baader & Nipkow 1998). We write $u[s]$ when s occurs as a subterm of u. In that case $u[t]$ denotes the replacement of the indicated occurrence of s by t. An *equation in L* is an unordered pair of L-terms, denoted by $s \approx t$. A *rule in L* is an ordered pair of L-terms, denoted by $s \rightarrow t$. An equation or a rule is *ground* if the terms in it are ground. A *system* is a finite set. Let R be a system of ground rules, and s and t two ground terms. Then s *rewrites* in R to t, denoted by $s \xrightarrow[R]{} t$, if t is obtained from s by replacing an occurrence of a term l in s by a term r for some rule $l \rightarrow r$ in R. The term s *reduces* in R to t, denoted by $s \xrightarrow[R]{*} t$, if either $s = t$ or s rewrites to a term that reduces to t. R is called *symmetric* if, with any rule $l \rightarrow r$ in R, R also contains its converse $r \rightarrow l$. Below we shall not distinguish between systems of equations and symmetric systems of rewrite rules. The *size* of a system R is the sum of the sizes of its components: $\|R\| = \sum_{l \rightarrow r \in R} (\|l\| + \|r\|)$.

Rigid Reachability. Let L be a first-order language. A *reachability constraint*, or simply a constraint in L is a triple (R, s, t) where R is a set of rules in L, and s and t are terms in L. We refer to R, s and t as the *rule set*, the *source term* and the *target term*, respectively, of the constraint. A substitution θ in L *solves* (R, s, t) (in L) if θ is grounding for R, s and t, and $s\theta \xrightarrow[R\theta]{*} t\theta$. The problem of solving constraints (in L) is called *rigid reachability* (for L). A system of constraints is *solvable* if there exists a substitution that solves all constraints in that system. *Simultaneous rigid reachability* or *SRR* is the problem of solving systems of constraints. *Monadic* (simultaneous) rigid reachability is (simultaneous) rigid reachability for monadic languages.

Rigid E-unification is rigid reachability for constraints (E, s, t) with sets of equations E. *Simultaneous Rigid E-unification* or *SREU* is defined accordingly.

Finite Tree Automata. Finite bottom-up tree automata, or simply, tree automata, from here on, are a generalization of classical automata (Doner 1970, Thatcher & Wright 1968). Using a rewrite rule based definition (Coquidé, Dauchet, Gilleron & Vágvölgyi 1994, Dauchet 1993), a *tree automaton* (or *TA*) A is a quadruple $(Q_A, \Sigma_A, R_A, F_A)$, where (i) Q_A is a finite set of constants called *states*, (ii) Σ_A is a *signature* that is disjoint from Q_A, (iii) R_A is a system of *rules* of the form $f(q_1, \ldots, q_n) \to q$, where $f \in \Sigma_A$ has arity $n \geq 0$ and $q, q_1, \ldots, q_n \in Q_A$, and (iv) $F_A \subseteq Q_A$ is the set of *final states*. The *size* of a TA A is $\|A\| = |Q_A| + \|R_A\|$.

We denote by $T(A, q)$ the set $\{t \in \mathcal{T}_{\Sigma_A} \mid t \xrightarrow[R_A]{*} q\}$ of ground terms *accepted* by A in state q. The set of terms *recognized* by the TA A is the set $\bigcup_{q \in F_A} T(A, q)$. A set of terms is called *recognizable* or *regular* if it is recognized by some TA.

Word automata. In monadic signatures, every function symbol has an arity at most 1, thus terms are words. For monadic signatures, we thus use the traditional, equivalent concepts of alphabets, words, finite automata, and regular expressions. A word with a variable $a_1 \ldots a_n x$ corresponds to the term $a_1(a_2(\ldots a_n(x))) \in \mathcal{T}_\Sigma$. The substitution of x by a term u is the same as the concatenation of the respective words. A finite (word) automaton A is a tuple $(Q_A, \Sigma_A, R_A, q^{\mathrm{i}}_A, F_A)$ where the components Q_A, Σ_A, R_A, F_A have the same form and meaning as the corresponding components of a tree automaton over a monadic signature, and where, additionally, q^{i}_A is the initial state. A transition $a(q) \to q'$ of R_A ($a \in \Sigma_A$, $q, q' \in Q_A$) is denoted $q \xrightarrow{a} q'$.

Second-Order Unification. Second-order unification is unification for second-order terms. For representing unifiers, we need expressions representing functions which, when applied, produce instances of a term in the given language L. Following Goldfarb (Goldfarb 1981) and Farmer (Farmer 1991), we, therefore, introduce the concept of an *expansion* L^* of L. Let $\{z_i\}_{i \geq 1}$

be an infinite collection of new symbols not in L. The language L^* differs from L by having $\{z_i\}_{i\geq 1}$ as additional first-order variables, called *bound variables*. The *rank* of a term t in L^*, is either 0 if t contains no bound variables (*i.e.*, $t \in \mathcal{T}_L$), or the largest n such that z_n occurs in t. Given terms t and $t_1, t_2, \ldots, t_n$ in L^*, we write $t[t_1, t_2, \ldots, t_n]$ for the term that results from t by simultaneously replacing z_i in t by t_i for $1 \leq i \leq n$. An L^*-term is called *closed* if it contains no variables other than bound variables. Note that closed L^*-terms of rank 0 are ground L-terms.

A *substitution in* L is a function θ with finite domain $\mathrm{dom}(\theta) \subseteq \mathcal{X}_L \cup \mathcal{F}_L$ that maps first-order variables to L-terms, and n-ary second-order variables to L^*-terms of rank $\leq n$. The result of applying a substitution θ to an L-term s, denoted by $s\theta$, is defined by induction on s:

1. If $s = x$ and $x \in \mathrm{dom}(\theta)$ then $s\theta = \theta(x)$.

2. If $s = x$ and $x \notin \mathrm{dom}(\theta)$ then $s\theta = x$.

3. If $s = F(t_1, \ldots, t_n)$ and $F \in \mathrm{dom}(\theta)$ then $s\theta = \theta(F)[t_1\theta, \ldots, t_n\theta]$.

4. If $s = F(t_1, \ldots, t_n)$ and $F \notin \mathrm{dom}(\theta)$ then $s\theta = F(t_1\theta, \ldots, t_n\theta)$.

5. If $s = f(t_1, \ldots, t_n)$ then $s\theta = f(t_1\theta, \ldots, t_n\theta)$.

We also write $F\theta$ for $\theta(F)$, where F is a second-order variable. A substitution is called *closed*, if its range is a set of closed terms. Given a term t, a substitution θ is said to be *grounding for* t if $t\theta$ is ground, similarly for other L-expressions. Given a sequence $\vec{t} = t_1, \ldots, t_n$ of terms, we write $\vec{t}\theta$ for $t_1\theta, \ldots, t_n\theta$.

Let E be a system of equations in L. A *unifier* of E is a substitution θ (in L) such that $s\theta = t\theta$ for all equations $s \approx t$ in E. E is *unifiable* if there exists a unifier of E. Note that if E is unifiable then it has a closed unifier that is grounding for E, since $\mathcal{T}_{\Sigma_L}$ is nonempty. The *unification problem for* L is the problem of deciding whether a given equation system in L is unifiable. In general, the *second-order unification* problem or *SOU* is the unification problem for arbitrary second-order languages. *Monadic* SOU is SOU for monadic second-order languages. By SOU *with one second-order variable* we mean the unification problem for second-order languages L such that $|\mathcal{F}_L| = 1$.

Following common practice, by an *exponential* function we mean an integer function of the form $f(n) = 2^{P(n)}$ where P is a polynomial. The complexity class EXPTIME is defined accordingly.

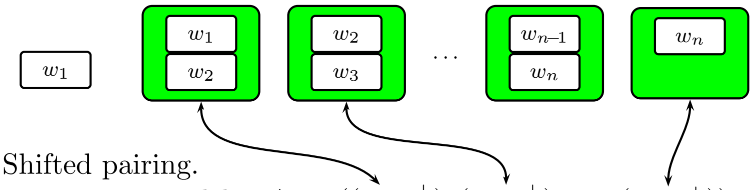


Figure 1: Shifted pairing.

The terms recognized by A_{mv}, $((v_1, v_1^+), (v_2, v_2^+), \dots, (v_n, v_n^+))$, represent a sequence of moves of a given Turing machine, where v_i^+ is the successor of v_i according to the transition function of the TM.

Each term t recognized by A_{id} represents a sequence of IDs of the TM $(w_1, w_2, \dots, w_n)$.

The two rewrite systems Π_1 and Π_2 are such that s reduces in Π_1 to t if and only if $v_i = w_i$ for $1 \leq i \leq k = n$, and $t_{\mathrm{id}} \centerdot s$ reduces in Π_2 to t if and only if t_{id} represents w_1, $v_i^+ = w_{i+1}$ for $1 \leq i < n$, and w_n is the final ID of the TM. It follows that such s and t exist if and only if the TM accepts the input string represented by t_{id}.

---

3 Rigid Reachability is Undecidable

We prove that rigid reachability is undecidable. The undecidability holds already for constraints with some fixed ground rule set which is, moreover, terminating. Our main tool in proving the undecidability result is the following statement.

Lemma 1 ((Gurevich & Veanes 1998)) *One can effectively construct two tree automata* $A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\})$, $A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\})$, *and two canonical systems of ground rules* $\Pi_1, \Pi_2 \subseteq \mathcal{T}_{\Sigma_{\mathrm{mv}}} \times \mathcal{T}_{\Sigma_{\mathrm{id}}}$, *where the only common symbol in* A_{mv} *and* A_{id} *is a binary function symbol* $\centerdot$,[1] *such that, it is undecidable whether, given* $t_{\mathrm{id}} \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, *there exists* $s \in T(A_{\mathrm{mv}})$ *and* $t \in T(A_{\mathrm{id}})$ *such that* $s \xrightarrow[\Pi_1]{*} t$ *and* $t_{\mathrm{id}} \centerdot s \xrightarrow[\Pi_2]{*} t$.

The main idea behind the proof of Lemma 1 is illustrated in Figure 1. In the rest of this section, we consider fixed A_{mv}, A_{id}, Π_1 and Π_2 given by Lemma 1.

Undecidability of simultaneous rigid E-unification follows from this lemma by viewing the rules R_{mv} and R_{id} of the automata A_{mv} and A_{id}, respectively, as well as the rewrite systems Π_1 and Π_2, as sets of equations, and by formulating the reachability constraints between s and t as a system of rigid equations. It is *not* possible, though, to achieve the same effect by a *single* rigid E-unification constraint for a combined system of equations.

[1] We write $\centerdot$ ("dot") as an infix operator.

The interference between the component systems cannot be controlled due to the symmetry of equality. This is different for reachability where rewrite rules are only applied from left to right. In fact, our main idea in the undecidability proof is to combine the four rewrite systems R_{mv}, R_{id}, Π_1, and Π_2 into a single system and achieve mutual non-overlapping of rewrite rules by renaming the constants in the respective signatures.

3.1 Renaming of Constants

For any integer m and a signature Σ we write $\Sigma^{(m)}$ for the constant-disjoint copy of Σ where each constant c has been replaced with a new constant $c^{(m)}$, we say that $c^{(m)}$ *has label* m. Note that non-constant symbols are not renamed. For a ground term t and a set of ground rules R over Σ, we define $t^{(m)}$ and $R^{(m)}$ over $\Sigma^{(m)}$ accordingly.

Given a signature Σ and two different integers m and n, we write $\Sigma^{(m,n)}$ for the following set of rules that simply replaces each label m with label n:

$$\Sigma^{(m,n)} = \{\, c^{(m)} \to c^{(n)} \mid c \in \mathit{Con}(\Sigma) \,\}.$$

We write $\Pi^{(m,n)}$, where Π is either Π_1 or Π_2, for the following set of rules:

$$\Pi^{(m,n)} = \{\, l^{(m)} \to r^{(n)} \mid l \to r \in \Pi \,\}.$$

Lemma 2 *Let m, n, k and l be distinct integers. The statements (i) and (ii) are equivalent for all all $s \in \mathcal{T}_{\Sigma_{\mathrm{mv}}}$ and $t_{\mathrm{id}}, t \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$.*

(i) $s \xrightarrow[\Pi_1]{*} t$ *and* $t_{\mathrm{id}} \centerdot s \xrightarrow[\Pi_2]{*} t$.

(ii) $s^{(m)} \xrightarrow[\Pi_1{}^{(m,n)}]{*} t^{(n)}$ *and* $t_{\mathrm{id}}^{(l)} \centerdot s^{(k)} \xrightarrow[\Pi_2{}^{(k,l)}]{*} t^{(l)}$.

Proof. The left-hand sides of the rules in Π_1 and Π_2 are terms in $\mathcal{T}_{\Sigma_{\mathrm{mv}}}$ and the right-hand sides of the rules in Π_1 and Π_2 are terms in $\mathcal{T}_{\Sigma_{\mathrm{id}}}$. But Σ_{mv} and Σ_{id} are constant-disjoint. ⊠

3.2 The Main Construction

Let R_{u} be the following system of ground rules:

$$\begin{aligned} R_{\mathrm{u}} \;=\; & R_{\mathrm{mv}}^{(0)} \cup R_{\mathrm{mv}}^{(2)} \cup {\Sigma_{\mathrm{mv}}}^{(0,1)} \cup {\Sigma_{\mathrm{mv}}}^{(2,1)} \cup R_{\mathrm{id}}^{(4)} \cup R_{\mathrm{id}}^{(6)} \cup {\Sigma_{\mathrm{id}}}^{(4,3)} \cup {\Sigma_{\mathrm{id}}}^{(6,3)} \cup \\ & {\Sigma_{\mathrm{id}}}^{(4,5)} \cup {\Pi_1}^{(0,5)} \cup {\Sigma_{\mathrm{id}}}^{(6,7)} \cup {\Pi_2}^{(2,7)} \end{aligned}$$

Note that constants with odd labels occur only in the right-hand sides of rules and can, once introduced, subsequently not be removed by R_{u}. Let f_{u} be a new function symbol with arity 12. We consider the following constraint:

$$\begin{pmatrix} R_{\mathrm{u}}, & f_{\mathrm{u}}(& x_0, & x_2, & x_0, & x_2, & y_4, & y_6, & y_4, & y_6, & y_4, & x_0, & y_6, & t_{\mathrm{id}}^{(7)} \centerdot x_2 &), \\ & f_{\mathrm{u}}(& q_{\mathrm{mv}}^{(0)}, & q_{\mathrm{mv}}^{(2)}, & x_1, & x_1, & q_{\mathrm{id}}^{(4)}, & q_{\mathrm{id}}^{(6)}, & y_3, & y_3, & y_5, & y_5, & y_7, & y_7 &) \end{pmatrix} \quad (1)$$

Our goal is to show that solvability of (1), for a given $t_{\mathrm{id}} \in \Sigma_{\mathrm{id}}$, is equivalent to the existence of s and t satisfying the condition in Lemma 1. Note that, for all ground terms t_i and s_i, for $1 \leq i \leq 12$,

$$f_{\mathrm{u}}(t_1, \ldots, t_{12}) \xrightarrow[R_{\mathrm{u}}]{*} f_{\mathrm{u}}(s_1, \ldots, s_{12}) \quad \Leftrightarrow \quad t_i \xrightarrow[R_{\mathrm{u}}]{*} s_i \text{ (for } 1 \leq i \leq 12).$$

As a first step, we prove a lemma that allows us to separate the different subsystems of R_{u} that are relevant for the reductions between the corresponding arguments of f_{u} in the source term and the target term of (1).

Lemma 3 *For every substitution θ, θ solves the constraint (1) if and only if θ solves the system (2)–(5) of constraints.*

$$\left.\begin{array}{llll} (& R_{\mathrm{mv}}^{(0)}, & x_0, & q_{\mathrm{mv}}^{(0)} &) \\ (& R_{\mathrm{mv}}^{(2)}, & x_2, & q_{\mathrm{mv}}^{(2)} &) \\ (& \Sigma_{\mathrm{mv}}{}^{(0,1)}, & x_0, & x_1 &) \\ (& \Sigma_{\mathrm{mv}}{}^{(2,1)}, & x_2, & x_1 &) \end{array}\right\} \quad (2)$$

$$\left.\begin{array}{llll} (& R_{\mathrm{id}}^{(4)}, & y_4, & q_{\mathrm{id}}^{(4)} &) \\ (& R_{\mathrm{id}}^{(6)}, & y_6, & q_{\mathrm{id}}^{(6)} &) \\ (& \Sigma_{\mathrm{id}}{}^{(4,3)}, & y_4, & y_3 &) \\ (& \Sigma_{\mathrm{id}}{}^{(6,3)}, & y_6, & y_3 &) \end{array}\right\} \quad (3)$$

$$\left.\begin{array}{llll} (& \Sigma_{\mathrm{id}}{}^{(4,5)}, & y_4, & y_5 &) \\ (& \Pi_1{}^{(0,5)}, & x_0, & y_5 &) \end{array}\right\} \quad (4)$$

$$\left.\begin{array}{llll} (& \Sigma_{\mathrm{id}}{}^{(6,7)}, & y_6, & y_7 &) \\ (& \Pi_2{}^{(2,7)}, & t_{\mathrm{id}}^{(7)} \centerdot x_2, & y_7 &) \end{array}\right\} \quad (5)$$

Proof. The direction '$\Leftarrow$' is immediate, since if θ solves a constraint (R, s, t) then obviously it solves any constraint (R', s, t) where $R \subseteq R'$. We prove the direction '$\Rightarrow$', by showing that θ solves the subsystems (2) and (4). The other cases are symmetrical. So assume that θ solves (1).

Proof of θ solving (2)

1. We show that $x_i\theta \xrightarrow[R_{\mathrm{mv}}^{(i)}]{*} q_{\mathrm{mv}}^{(i)}$ for $i = 0$. (by symmetry, this also proves the case $i = 2$.) We know that $x_0\theta \xrightarrow[R_{\mathrm{u}}]{*} q_{\mathrm{mv}}^{(0)}$. We prove by induction on the length of reductions that, for all t, if $t \xrightarrow[R_{\mathrm{u}}]{*} q_{\mathrm{mv}}^{(0)}$ then $t \xrightarrow[R_{\mathrm{mv}}^{(0)}]{*} q_{\mathrm{mv}}^{(0)}$.

   The base case (reduction is empty) holds trivially. If the reduction is nonempty, then we have for some $l \to r \in R_{\mathrm{u}}$, and by using the induction hypothesis, that

   $$t \xrightarrow[l \to r]{} s \xrightarrow[R_{\mathrm{mv}}^{(0)}]{*} q_{\mathrm{mv}}^{(0)}.$$

   Therefore, all constants in r have label 0, since r is a subterm of s and $s \in \mathcal{T}_{\Sigma_{\mathrm{mv}}^{(0)} \cup Q_{\mathrm{mv}}^{(0)}}$. Hence $l \to r \in R_{\mathrm{mv}}^{(0)}$, and consequently $t \xrightarrow[R_{\mathrm{mv}}^{(0)}]{*} q_{\mathrm{mv}}^{(0)}$.

2. We prove that $x_i\theta \xrightarrow[{\Sigma_{\mathrm{mv}}}^{(i,1)}]{*} x_1\theta$ for $i = 0$. (The proof is symmetrical for $i = 2$.) We know that $x_i\theta \xrightarrow[R_{\mathrm{u}}]{*} x_1\theta$ for $i = 0, 2$. Suppose (by contradiction) that

   $$x_0\theta \xrightarrow[{\Sigma_{\mathrm{mv}}}^{(0,1)}]{*} s \xrightarrow[l \to r]{} t \xrightarrow[R_{\mathrm{u}}]{*} x_1\theta,$$

   where $l \to r \in R_{\mathrm{u}} \setminus {\Sigma_{\mathrm{mv}}}^{(0,1)}$. All constants in s and thus in l have label 0 or 1, since all constants in $x_0\theta$ have label 0 by above. It follows that $l \to r \in R_{\mathrm{mv}}^{(0)}$ or $l \to r \in {\Pi_1}^{(0,5)}$. We consider both cases separately.

   (a) Assume that $l \to r \in R_{\mathrm{mv}}^{(0)}$. Then $r \in Q_{\mathrm{mv}}^{(0)}$, and thus $Con(t) \cap Q_{\mathrm{mv}}^{(0)} \neq \emptyset$. Hence $Con(x_1\theta) \cap Q_{\mathrm{mv}}^{(0)} \neq \emptyset$. This contradicts that $x_2\theta \xrightarrow[R_{\mathrm{u}}]{*} x_1\theta$, because all constants in $x_2\theta$ have label 2 by above.

   (b) Assume that $l \to r \in {\Pi_1}^{(0,5)}$. Then $x_1\theta$ contains a constant with label 5, contradicting again that $x_2\theta \xrightarrow[R_{\mathrm{u}}]{*} x_1\theta$.

   It follows that $x_0\theta \xrightarrow[{\Sigma_{\mathrm{mv}}}^{(0,1)}]{*} x_1\theta$.

Proof of θ solving (4) We know that $y_4\theta \xrightarrow[R_{\mathrm{u}}]{*} y_5\theta$ and $x_0\theta \xrightarrow[R_{\mathrm{u}}]{*} y_5\theta$. We prove first that $y_4\theta \xrightarrow[{\Sigma_{\mathrm{id}}}^{(4,5)}]{*} y_5\theta$. Suppose (by contradiction) that:

$$y_4\theta \xrightarrow[{\Sigma_{\mathrm{id}}}^{(4,5)}]{*} s \xrightarrow[l \to r]{} t \xrightarrow[R_{\mathrm{u}}]{*} y_5\theta,$$

where $l \to r \in R_{\mathrm{u}} \setminus {\Sigma_{\mathrm{id}}}^{(4,5)}$. Then either $l \to r \in R_{\mathrm{id}}^{(4)}$ or $l \to r \in {\Sigma_{\mathrm{id}}}^{(4,3)}$. The former case implies that $Con(y_5\theta) \cap Q_{\mathrm{id}}^{(4)} \neq \emptyset$ and the latter case implies that $Con(y_5\theta) \cap \Sigma_{\mathrm{id}}^{(3)} \neq \emptyset$. Both cases contradict that $x_0\theta \xrightarrow[R_{\mathrm{u}}]{*} y_5\theta$, because all constants in $x_0\theta$ have label 0.

To prove that $x_0\theta \xrightarrow[\Pi_1{}^{(0,5)}]{*} y_5\theta$, note that any rule outside $\Pi_1{}^{(0,5)}$ with the left-hand side having constants with label 0 would either introduce a constant from $Q_{\mathrm{mv}}^{(0)}$ to $y_5\theta$ or a constant with label 1 to $y_5\theta$, in both cases contradicting that $y_4\theta \xrightarrow[\Sigma_{\mathrm{id}}{}^{(4,5)}]{*} y_5\theta$. ⊠

The following lemma relates the solvability of (1) to the Lemma 1.

Lemma 4 *For $t_{\mathrm{id}} \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, the constraint (1) is solvable if and only if there exists $s \in T(A_{\mathrm{mv}})$ and $t \in T(A_{\mathrm{id}})$ such that $s \xrightarrow[\Pi_1]{*} t$ and $t_{\mathrm{id}} \centerdot s \xrightarrow[\Pi_2]{*} t$.*

Proof. ($\Leftarrow$) Assuming that given s and t exist, define $x_i\theta = s^{(i)}$ for $i \in \{0,1,2\}$ and $y_i\theta = t^{(i)}$ for $i \in \{3,4,5,6,7\}$. It follows easily from Lemma 2 and Lemma 3 that θ solves (1).

($\Rightarrow$) Assume that θ solves (1). By Lemma 3, θ solves (2)–(5). First we observe the following facts.
(i) From θ solving (2), it follows that there exists $s \in T(A_{\mathrm{mv}})$ such that $x_0\theta = s^{(0)}$ and $x_2\theta = s^{(2)}$.
(ii) From θ solving (3), it follows that there exists $t \in T(A_{\mathrm{id}})$ such that $y_4\theta = t^{(4)}$ and $y_6\theta = t^{(6)}$.
From θ solving (4) and by using (ii), it follows that $y_5\theta = t^{(5)}$. Now, due to the second component of (4) and by using (i), we get that: $s^{(0)} \xrightarrow[\Pi_1{}^{(0,5)}]{*} t^{(5)}$.
From θ solving (5) and by using (ii), it follows that $y_7\theta = t^{(7)}$. Now, due to the second component of (5) and by using (i), we get that: $t_{\mathrm{id}}^{(7)} \centerdot s^{(2)} \xrightarrow[\Pi_2{}^{(2,7)}]{*} t^{(7)}$.
Finally, use Lemma 2. ⊠

We conclude with the following result.

Theorem 1 Rigid reachability is undecidable. Specifically, it is undecidable already under the following restrictions:

- the rule set is some fixed ground terminating rewrite system;
- there are at most eight variables;
- each variable occurs at most three times;
- the source term and the target term do not share variables.

Proof. The undecidability follows from Lemma 1 and Lemma 4. The system R_{u} is easily seen to be terminating, by simply examining the subsystems. The other restrictions follow immediately as properties of (1). ⊠

We have not attempted to minimize the number of variables in (1). Observe also that all but one of the occurrences of variables are shallow (the target term is shallow).

4 A New Undecidability Result for SOU

We prove that SOU is undecidable already when unification problems contain just a single second-order variable which, in addition, occurs twice. This result contrasts a claim to the opposite in (Levy 1998). Let Σ_{u} be the signature consisting of the symbols in R_{u} and the symbol f_{u}. Let $R_{\mathrm{u}} = \{\, l_i \to r_i \mid 1 \leq i \leq m \,\}$. Let $\vec{l}_{\mathrm{u}}$ denote the sequence $l_1, l_2, \ldots, l_m$ and $\vec{r}_{\mathrm{u}}$ the sequence $r_1, r_2, \ldots, r_m$. Let L_{u} be the following language:

$$L_{\mathrm{u}} = (\Sigma_{\mathrm{u}}, \{x_0, x_1, x_2, y_3, y_4, y_5, y_6, y_7\})$$

Let F_{u} be a new second-order variable with arity $m+1$. Let $\underline{\mathrm{cons}}$ be a new binary function symbol and $\underline{\mathrm{nil}}$ a new constant. The language L_1 is defined as the following expansion of L_{u}:

$$L_1 = (\Sigma_{\mathrm{u}} \cup \{\underline{\mathrm{cons}}, \underline{\mathrm{nil}}\}, \mathcal{X}_{L_{\mathrm{u}}}, \{F_{\mathrm{u}}\}).$$

We can show that, given $t_{\mathrm{id}} \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, the following second-order equation in L_1 is solvable if and only if the constraint (1) is solvable:

$$\begin{aligned} &F_{\mathrm{u}}(\vec{l}_{\mathrm{u}}, \underline{\mathrm{cons}}(f_{\mathrm{u}}(q^{(0)}_{\mathrm{mv}}, q^{(2)}_{\mathrm{mv}}, x_1, x_1, q^{(4)}_{\mathrm{id}}, q^{(6)}_{\mathrm{id}}, y_3, y_3, y_5, y_5, y_7, y_7), \underline{\mathrm{nil}})) \approx \\ &\qquad \underline{\mathrm{cons}}(f_{\mathrm{u}}(x_0, x_2, x_0, x_2, y_4, y_6, y_4, y_6, y_4, x_0, y_6, t^{(7)}_{\mathrm{id}} \centerdot x_2), F_{\mathrm{u}}(\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}})) \end{aligned} \tag{6}$$

Lemma 5 *Given $t_{\mathrm{id}} \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, (1) is solvable if and only if (6) is solvable.*

Proof. The direction '$\Rightarrow$' follows from (Veanes 1998, Lemma 2) and the observation that if θ solves (1) then $x\theta \in \mathcal{T}_{\Sigma_{\mathrm{u}}}$ for all $x \in \mathcal{X}_{L_{\mathrm{u}}}$. In particular, it is not possible that $\underline{\mathrm{cons}}$ or $\underline{\mathrm{nil}}$ appear in the terms that are substituted for $\mathcal{X}_{L_{\mathrm{u}}}$.

We prove the other direction. Assume that θ solves (6). We show that θ solves (1). A straightforward inductive argument shows that $F_{\mathrm{u}}\theta$ is an L_1^*-term of rank $m+1$ of the following form: (recall that z_i is the i'th bound variable)

$$F_{\mathrm{u}}\theta = \underline{\mathrm{cons}}(s_1, \underline{\mathrm{cons}}(s_2, \ldots, \underline{\mathrm{cons}}(s_k, z_{m+1}) \cdots)),$$

for some $k \geq 1$, by using that R_{u} is ground and that $\underline{\mathrm{cons}} \notin \Sigma_{\mathrm{u}}$ (see (Veanes 1998, Lemma 1)). Hence, since θ solves (6), it follows that

$$\begin{array}{llll} \underline{\mathrm{cons}}(s_1[\vec{l}_{\mathrm{u}}, t'], & \ldots\ \underline{\mathrm{cons}}(s_{i+1}[\vec{l}_{\mathrm{u}}, t'], & \ldots\ \underline{\mathrm{cons}}(t\theta, & \underline{\mathrm{nil}}) \cdots) \cdots) = \\ \underline{\mathrm{cons}}(s\theta, & \ldots\ \underline{\mathrm{cons}}(s_i[\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}}], & \ldots\ \underline{\mathrm{cons}}(s_k[\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}}], & \underline{\mathrm{nil}}) \cdots) \cdots), \end{array} \tag{7}$$

where s is the source term of (1), t is the target term of (1), and $t' = \underline{\mathrm{cons}}(t\theta, \underline{\mathrm{nil}})$. So there exists a reduction in $R_{\mathrm{u}} \cup \{t' \to \underline{\mathrm{nil}}\}$ of the following form:

$$\begin{array}{ccccccc} s_1[\vec{l}_{\mathrm{u}}, t'] & & s_2[\vec{l}_{\mathrm{u}}, t'] & & s_k[\vec{l}_{\mathrm{u}}, t'] & & t\theta \\ \shortparallel & \searrow^{*} & \shortparallel & \cdots & \shortparallel & \searrow^{*} & \shortparallel \\ s\theta & & s_1[\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}}] & & s_{k-1}[\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}}] & & s_k[\vec{r}_{\mathrm{u}}, \underline{\mathrm{nil}}] \end{array}$$

This means that $s\theta \xrightarrow[R_{\mathrm{u}} \cup \{t' \to \underline{\mathrm{nil}}\}]{*} t\theta$, *i.e.*,

$$\begin{array}{l} f_{\mathrm{u}}(\ x_0,\ \ x_2,\ \ x_0,\ x_2,\ \ y_4,\ \ y_6,\ y_4,\ y_6,\ y_4,\ x_0,\ y_6,\ t_{\mathrm{id}}^{(7)} \centerdot x_2\ )\theta \\ \qquad\qquad\qquad \xrightarrow[R_{\mathrm{u}} \cup \{t' \to \underline{\mathrm{nil}}\}]{*} \\ f_{\mathrm{u}}(\ q_{\mathrm{mv}}^{(0)},\ q_{\mathrm{mv}}^{(2)},\ x_1,\ x_1,\ q_{\mathrm{id}}^{(4)},\ q_{\mathrm{id}}^{(6)},\ y_3,\ y_3,\ y_5,\ y_5,\ y_7,\ \ y_7\ )\theta \end{array}$$

It is sufficient to show that $x_0\theta, x_2\theta, y_4\theta, y_6\theta \in \mathcal{T}_{\Sigma_{\mathrm{u}}}$. Because then $s\theta \in \mathcal{T}_{\Sigma_{\mathrm{u}}}$ and the rule $t' \to \underline{\mathrm{nil}}$ can not be used in the reduction of $s\theta$ to $t\theta$, since $\underline{\mathrm{nil}}$ does not occur in Σ_{u}. To begin with, we observe that

$$x_i\theta \xrightarrow[R_{\mathrm{u}} \cup \{t' \to \underline{\mathrm{nil}}\}]{*} q_{\mathrm{mv}}^{(i)} \quad (i = 0, 2) \quad \text{and that} \quad y_i\theta \xrightarrow[R_{\mathrm{u}} \cup \{t' \to \underline{\mathrm{nil}}\}]{*} q_{\mathrm{id}}^{(i)} \quad (i = 4, 6).$$

It follows by easy induction on the length of reductions that $t' \to \underline{\mathrm{nil}}$ can not be used in these reductions, since $\underline{\mathrm{nil}}$ does not not occur in R_{u}. Hence, $x_0\theta, x_2\theta, y_4\theta, y_6\theta \in \mathcal{T}_{\Sigma_{\mathrm{u}}}$, as needed. $\boxtimes$

We conclude with the following result, that follows from Lemma 1, Lemma 4, and Lemma 5.

Theorem 2 Second-order unification is undecidable with one second-order variable that occurs at most twice.

The role of first-order variables in the above undecidability result is important. Without first-order variables, and if there is only one second-order variable that occurs at most twice, second-order unification reduces to *ground reachability*, (Levy & Veanes 1998), and is thus decidable.

5 Decidable Cases

We show that rigid reachability and simultaneous rigid reachability are decidable when the rules are all ground, either the source s or the target t of any constraint is linear, and the source s and the target t are *variable-disjoint*, that is, $Var(s) \cap Var(t) = \emptyset$. The non-simultaneous case then turns out to be EXPTIME-complete. EXPTIME-hardness holds already with just a single variable. This contrasts with the fact that rigid E-unification with one variable is $\mathcal{P}$-complete (Degtyarev, Gurevich, Narendran, Veanes & Voronkov 1998). When additionally both the source and target terms are linear, then rigid reachability and simultaneous rigid reachability are both $\mathcal{P}$-complete.

Note that the only difference between the conditions for undecidability of rigid reachability in Theorem 1 and the condition for decidability in Theorems 6, 7, and 8 is the linearity of source and (or) target terms. In the rest of the section, we assume fixed a signature Σ.

5.1 Decidable Cases of Rigid Reachability

We begin with defining a reduction from rigid reachability to the emptiness problem of the intersection of n regular languages recognized by tree automata $A_1,\ldots,A_n$. This intersection emptiness problem is known to be EXPTIME-complete, see (Frühwirth, Shapiro, Vardi & Yardemi 1991), (Seidl 1994) and (Veanes 1997). We may assume the states sets of the $A_1,\ldots,A_n$ to be disjoint and that each of these tree automata has only one final state. We call these final states, respectively, $q^{\mathrm{f}}_{A_1},\ldots, q^{\mathrm{f}}_{A_n}$. For stating the following lemma, we extend the signature Σ by a new symbol f of arity n, and assume that $n > 1$.

Lemma 6 *$T(A_1) \cap \ldots \cap T(A_n) \neq \emptyset$ iff the following constraint: $(R_{A_1} \cup \ldots \cup R_{A_n},\ f(x,\ldots,x),\ f(q^{\mathrm{f}}_{A_1},\ldots,q^{\mathrm{f}}_{A_n}))$ has a solution.*

Proof. $(\Rightarrow)$ is obvious. For $(\Leftarrow)$ we use the fact that the new symbol does not occur in any transition rule of the $A_1,\ldots,A_n$. Therefore, and since the state sets are disjoint, any reduction in $f(x,\ldots,x)\theta \xrightarrow[R_{A_1}\cup\ldots\cup R_{A_n}]{*} f(q^{\mathrm{f}}_{A_1},\ldots,q^{\mathrm{f}}_{A_n})$ (where θ is a solution) takes place in one of the arguments of $f(x,\ldots,x)\theta$. Moreover, if the reduction is in the i-th subterm, it corresponds to the application of a rule in R_{A_i}. (It is possible, though, to apply a start rule in R_{A_j} within the i-th subterm, with $i \neq j$. But any reduction of this form blocks in that the final state $q^{\mathrm{f}}_{A_i}$ can not be reached from the reduct.) The fact that $n > 1$ prohibits states of the automata to appear in $x\theta$. ⊠

Theorem 3 Rigid reachability is EXPTIME-hard even when the rules and the target are ground and the source contains only a single variable.

For obtaining also an EXPTIME upper bound for a somewhat less restrictive case of rigid reachability we will apply tree automata techniques. In particular, we will exploit the following fact of preservation of recognizability under rewriting, which is a direct consequence of results in (Dauchet, Heuillard, Lescanne & Tison 1988).

Proposition 4 ((Coquidé & Gilleron 1990)) Let R be a ground rewrite system and t a linear term. The set $\{u \in \mathcal{T}_\Sigma \mid u \xrightarrow[R]{*} t\sigma, t\sigma \text{ ground}\}$ is recognizable by a tree automaton A the size of which is in $O(\|t\| * \|R\|^2)$.

Proposition 5 The subset of $\mathcal{T}_\Sigma$ of ground instances of a given linear term s is recognizable by a tree automaton A_s the size of which is linear in the size of s.

Theorem 6 Rigid reachability, when rules are ground, the target is linear and the source and the target are variable-disjoint, can be decided in time $O(n^{3k+4})$, where n is the size of the constraint, and k is the total number of occurrences of non-linear variables in the source term.

Observe that the time upper bound becomes $O(n^4)$ when s is linear, since $k = 0$ in this case.

Proof. Assume to be given a reachability constraint (R, s, t) of the required form. We first construct a tree automaton A from t and R with the properties as provided by Proposition 4, that is, recognizing the predecessors with respect to R of the ground instances of t. The size $\|A\|$ of A is in $O(n^3)$.

If the source s is linear, then there is a solution for (R, s, t) iff $T(A) \cap T(A_s) \neq \emptyset$, where A_s is the tree automaton of Proposition 5. Since the intersection of recognizable languages is recognizable by a tree automaton the size of which is the product of the sizes of the original tree automata, the solvability of the given constraint can be decided in time $O(\|s\| * n^3) \subseteq O(n^4)$.

If the source s is not linear, we reduce our rigid reachability problem to $|Q_A|^k$ problems of the above type. We assume wlog that A has only one final state q^{f}. Let (s_i) be the finite sequence of terms which can be obtained from the source s by the following replacements: for every variable x which occurs $j \geq 2$ times in s, we choose a tuple $(q_1, \ldots, q_j)$ of states of A such that[2] $\cap_{i \leq j} T(A, q_i) \neq \emptyset$, and we replace the ith occurrence of x with q_i for $i \leq j$ in s.

Then the two following statements are equivalent:

(i) the constraint (R, s, t) has a solution.

(ii) one of the constraints $(R_A, s_i, q^{\mathrm{f}})$ has a solution.

(i) ⇒ (ii): Assume that σ is a solution of the constraint (R, s, t). This means in particular that $s\sigma \in T(A)$ *i.e.* $s\sigma \xrightarrow[A]{*} q^{\mathrm{f}}$. Let τ be the restriction of

[2]One can decide this property in time $\|A\|^k \in O(n^{3k})$.

σ to the set of linear variables of s and θ be its restriction to the set of non-linear variables of s. We have $s\theta \xrightarrow[R_A]{*} s_i$ by construction and τ is a solution of the constraint $(R_A, s_i, q^{\mathrm{f}})$.

(ii) $\Rightarrow$ *(i):* Assume $s_i\tau \xrightarrow[R_A]{*} q^{\mathrm{f}}$ for some i and some grounding substitution τ. To each non-linear variable x of s, we associate a term $s_x \in \cap_{i\leq j} T(A, q_i)$ (it exists by construction) where $q_1,\ldots,q_j$ are the states occurring in s_i at positions corresponding to x in s. This defines a substitution θ on the non-linear variables of s (by $x\theta = s_x$) such that $s\tau\theta \in T(A)$. Hence $s\tau\theta \xrightarrow[R]{*} t\sigma$ for some grounding substitution σ which is only defined on the variables of t. Since $\mathit{Var}(s) \cap \mathit{Var}(t) = \emptyset$, the domains of θ, τ and σ are pairwise disjoint and $\tau \cup \theta \cup \sigma$ is indeed a solution to the constraint (R, s, t).

Complexity: The number of possible s_i is smaller than $|Q_A|^k$ *i.e.* it is in $O(n^{3k})$. Rigid reachability for one constraint (A, s_i, q^{f}) can be decided in time $O(n^4)$, according to the first part of this proof. Altogether, this gives a decision time in $O(n^{3k+4})$. ⊠

By symmetry, rigid reachability is also decidable when rules are ground, the source is linear and the source and the target are variable-disjoint, with the same complexities as in Theorem 6 according to the (non)-linearity of the target.

As a consequence we obtain these two theorems:

Theorem 7 Rigid reachability is EXPTIME-complete when rules are ground, the source and the target are variable-disjoint, and either the source or the target is linear.

Theorem 8 Rigid reachability is $\mathcal{P}$-complete when the rules are ground, the source and the target are variable-disjoint, one of the source or the target is linear, and the number of occurrences of non-linear variables in the other is bounded by some fixed constant k independent from the problem.

Note that the linear case corresponds to $k = 0$.

Proof. For obtaining the lower bound, one may reduce the $\mathcal{P}$-complete uniform ground word problem (see (Kozen 1977)) to rigid reachability where rules, source and target are ground. The upper bound has been proved in Theorem 6. ⊠

5.2 Decidable Cases of Simultaneous Rigid Reachability

We now generalize Theorem 8 to the simultaneous case of rigid reachability.

Theorem 9 Simultaneous rigid reachability is $\mathcal{P}$-complete for systems of pairwise variable-disjoint constraints with ground rules, and sources and targets that are variable-disjoint and linear.

Proof. Apply Theorem 8 separately to each constraint of the system. $\boxtimes$

Similarly, we can prove:

Theorem 10 Simultaneous rigid reachability is EXPTIME-complete for systems of pairwise variable-disjoint constraints with ground rules, and sources and targets that are variable-disjoint and such that at least one of them is linear for each constraint.

The problem remains in $\mathcal{P}$ (see Theorem 8) if there is a constant k independent from the problem and for each s_i (resp. t_i) which is non-linear, the total number of occurrences of non-linear variable in s_i (resp. t_i) is smaller than k.

We can relax the conditions in the above Theorem 10 by allowing some common variables between the s_i.

Theorem 11 Simultaneous rigid reachability is in EXPTIME when all the rules of a system of constraints $((R_1, s_1, t_1), \ldots, (R_m, s_m, t_m))$ are ground, either every t_i is linear or every s_i is linear, and for all $i, j \leq m$, the terms s_i and t_j and respectively the terms t_i and t_j (when $i \neq j$) are variable-disjoint.

Proof. We prove for the case where every t_i is linear, the other case follows by symmetry. We reduce this problem to an exponential number of problems of the type of Theorem 10.

We associate a TA A_i to each pair (t_i, R_i) which recognizes the language $\{u \in \mathcal{T}_\Sigma \mid u \xrightarrow[R_i]{*} t_i\sigma, t_i\sigma \text{ ground}\}$ (see Proposition 4). The size of each A_i is in $O(\|t_i\| * \|R_i\|^2)$. We assume wlog that the states sets of the A_i are pairwise disjoint and that the final states sets of the A_i are singletons, namely $F_{A_i} = \{q_i^{\mathrm{f}}\}$. We construct for each $i \leq m$ a sequence of terms $(s_{i,j})$ obtained by replacement of variables occurrences in s_i (regardless of linearity) by states of A_i. To each m-tuple $(s_{1,j_1}, \ldots, s_{m,j_m})$, we associate a system which contains the constraints:

1. $(R_{A_1}, s_{1,j_1}, q_1^{\mathrm{f}}), \ldots, (R_{A_m}, s_{1,j_m}, q_m^{\mathrm{f}})$
2. for every variable x which occurs k times in $\{s_1, \ldots, s_n\}$, with $k \geq 2$, $(R_{A_1} \uplus \ldots \uplus R_{A_m}, f_{\mathrm{u}}^k(x, \ldots, x), f_{\mathrm{u}}^k(q_1, \ldots, q_k))$, where f_{u}^k is a new function symbol of arity k and $q_1, \ldots, q_k$ are the states occurring in $s_{1,j_1}, \ldots, s_{1,j_m}$ at the positions corresponding to x in $s_1, \ldots, s_m$.

Then the system $((R_1, s_1, t_1), \ldots, (R_n, s_n, t_n))$ has a solution iff one of the above systems has a solution. Each of these systems has a size which is polynomial in the size of the original system and moreover, each fulfills the hypothesis of Theorem 10 and can thus be decided in EXPTIME. Since the number of the above systems is exponential (in the size of the initial problem), we have an EXPTIME upper bound for the decision problem. ⊠

6 Monadic Simultaneous Rigid Reachability

Our second main decidability result generalizes the decidability proof of Monadic SREU for ground rules (Gurevich & Voronkov 1997). Moreover, our proof gives an EXPTIME upper bound to monadic SREU for ground rules. Although the lower bound is known to be PSPACE (Gurevich & Voronkov 1997), no interesting upper bound has been known before for this problem. We shall use basic automata theory for obtaining our result. More recently, and using different techniques, monadic rigid reachability with ground rules was found to be decidable also in PSPACE (Cortier 1998). The presentation in this section will be in terms of words rather than monadic terms.

Recognizing Substitution Instances. We will first show that n-tuples of substitution instances of monadic terms are recognizable. For this purpose we let automata compute on $((\Sigma \cup \{\perp\})^n)^*$, where $\perp$ is a new symbol. The representation of a pair of words of Σ^* as a word $((\Sigma \cup \{\perp\})^2)^*$ is given by the product $\otimes$ defined as follows:

$$\begin{array}{rcll} a_1 \ldots a_n \otimes b_1 \ldots b_m & = & \langle a_1, b_1\rangle \ldots \langle a_n, b_n\rangle\langle\perp, b_{n+1}\rangle \ldots \langle\perp, b_m\rangle & \text{if } n < m \\ a_1 \ldots a_n \otimes b_1 \ldots b_m & = & \langle a_1, b_1\rangle \ldots \langle a_m, b_m\rangle\langle a_{m+1}, \perp\rangle \ldots \langle a_n, \perp\rangle & \text{if } n \geq m \end{array}$$

We extend this definition of $\otimes$ associatively to tuples of arbitrary length in the obvious way.

Lemma 7 *Let $L_1, \ldots, L_n$ be recognizable subsets of Σ^*. Then $L_1 \otimes \ldots \otimes L_n$ is recognizable (in $((\Sigma \cup \{\perp\})^n)^*$). The size of the product automaton can be bounded by the product of the sizes of its factor automata.*

Proof. We present the construction for $n = 2$. It may be generalized by a straightforward induction argument to arbitrary n. Let A_1 and A_2, respectively, recognize $L_1\perp^*$ and $L_2\perp^*$. We define:

$$A_1 \otimes A_2 := \Big((Q_{A_1} \cup \{q_\perp\}) \times (Q_{A_2} \cup \{q_\perp\}), \Sigma, R, \langle q^{\mathrm{i}}_{A_1}, q^{\mathrm{i}}_{A_2}\rangle, F_{A_1} \times F_{A_2}\Big)$$

where $q_\perp$ is a new state symbol ($q_\perp \notin Q_{A_1} \cup Q_{A_2}$) and R is such that:

- if $p \xrightarrow{a} p' \in R_{A_1}$ then $\langle p, q_\perp \rangle \xrightarrow{\langle a,\perp \rangle} \langle p', q_\perp \rangle \in R$
- if $q \xrightarrow{b} q' \in R_{A_2}$ then $\langle q_\perp, q \rangle \xrightarrow{\langle \perp,b \rangle} \langle q_\perp, q' \rangle \in R$
- if $p \xrightarrow{a} p' \in R_{A_1}$ and $q \xrightarrow{b} q' \in R_{A_2}$ then $\langle p, q \rangle \xrightarrow{\langle a,b \rangle} \langle p', q' \rangle \in R$

$\boxtimes$

When constructing an automaton for $\Sigma^* \otimes L$ from an automaton A for L, the size of the product automaton can be bounded by $c * \|A\|$, for some (small) constant c if the alphabet Σ is assumed to be fixed.

Theorem 12 Given p monadic Σ-terms s_i, the set of tuples of their ground instances $\{s_1\theta \otimes \ldots \otimes s_p\theta \mid \theta \text{ ground } \}$ is recognizable by an automaton with size exponential in $\sum_{j=1}^{p} |s_j|$.

The proof of Theorem 12 will be based on three technical lemmas.

Lemma 8 *Let L be a language of tuples of the form $a_1 \otimes \ldots \otimes a_n$ that is recognized by A. Then, for any permutation π, the language*

$$L_\pi = \{a_{\pi 1} \otimes \ldots \otimes a_{\pi n} \mid a_1 \otimes \ldots \otimes a_n \in L\}$$

can be recognized by an automaton of the same size as A.

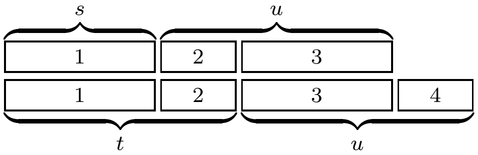
Proof. We apply the permutation π to the tuples of symbols appearing in the rules of R_A. $\boxtimes$

Lemma 9 *Given $s, t \in \Sigma^*$, the set $\{su \otimes tu \mid u \in \Sigma^*\}$ is recognizable by an automaton with size exponential in $|s| + |t|$.*

Proof. Let $s = s_0 \ldots s_n$ and $t = t_0 \ldots t_m$. By symmetry, we assume that $m \geq n$ and we let $p = \sum_{i=0}^{m-n} (|\Sigma| - 1)^i$. The automaton we are going to construct has $n + 2p + 1$ states. The n first states are denoted $q_0, \ldots, q_n$ and the others are $[w]$ and $[\![w]\!]$ where $w \in \Sigma^{m-n}$.

The initial state of A is q_0.

We classify the transitions of A in 4 categories, corresponding to the following decomposition of the word $su \otimes tu$ in 4 segments:



For category 1, there are $n+1$ transitions between states q_i, in order to recognize a first segment of length n of $su \otimes tu$. This gives the transitions:

$$q_i \xrightarrow{\langle s_i, t_i \rangle} q_{i+1} \text{ for } i < n \tag{8}$$

We ensure the transition between states q_i and states $[w]$ with the epsilon transition:

$$q_{n+1} \rightarrow [\,] \tag{9}$$

The purpose of the rules in category 2 is to memorize (into states $[w]$) the first symbols of u, in segment 2, while checking the rest of t. These rules are:

$$[w] \xrightarrow{\langle a, t_i \rangle} [wa] \text{ where } a \in \Sigma, |w| < m-n, i = n + |w| + 1 \tag{10}$$

It may also be the case that the length of u is smaller than $m-n$ (the length of segment 2). This corresponds to the apparition of some $\langle \perp, t_i \rangle$ in segment 2, which causes a jump to category 4. The rules for this case are:

$$[w] \xrightarrow{\langle \perp, t_i \rangle} [\![t_{i+1} \dots t_m w]\!] \text{ where } |w| < m-n-1, i = n + |w| + 1 \tag{11}$$

$$[w] \xrightarrow{\langle \perp, t_m \rangle} [\![w]\!] \text{ where } |w| = m-n-1 \tag{12}$$

The most important segment in the above decomposition is the third one. Note that it is the only segment which is not bounded in the length of s and t. Thus, the transitions of A which go through this segment have to make possible some loop between the states $[w]$ where $|w| = m-n$. The purpose of these transitions is to check that the u is the same in su and tu, with a delay $m-n$.

$$[a_1 \dots a_{m-n}] \xrightarrow{\langle a, a_1 \rangle} [a_2 \dots a_{m-n} a] \tag{13}$$

Finally, the transitions of the fourth category are supposed to finish the computation of A.

$$[a_1 \dots a_i] \xrightarrow{\langle \perp, a_1 \rangle} [\![a_2 \dots a_i]\!] \text{ where } i = m-n \tag{14}$$

$$[\![a_1 \dots a_i]\!] \xrightarrow{\langle \perp, a_1 \rangle} [\![a_2 \dots a_i]\!] \text{ where } i < m-n \tag{15}$$

The automaton A with transitions 8–15 recognizes $\{su \otimes tu \mid u \in \Sigma^*\}$. $\boxtimes$

Example 1 *Let $\Sigma = \{0,1\}$, $s = 0$ and $t = 101$, thus $n = 0$, $m = 2$. The automaton recognizing $\{su \otimes tu\}$ is depicted in figure 2. The categories to which the transition rules belong are indicated at the bottom of the figure.*

The construction underlying the proof of Lemma 9 cannot be generalized to the case of non-monadic signatures. However, one can generalize it (in the monadic case) to the product of an arbitrary number p of words.

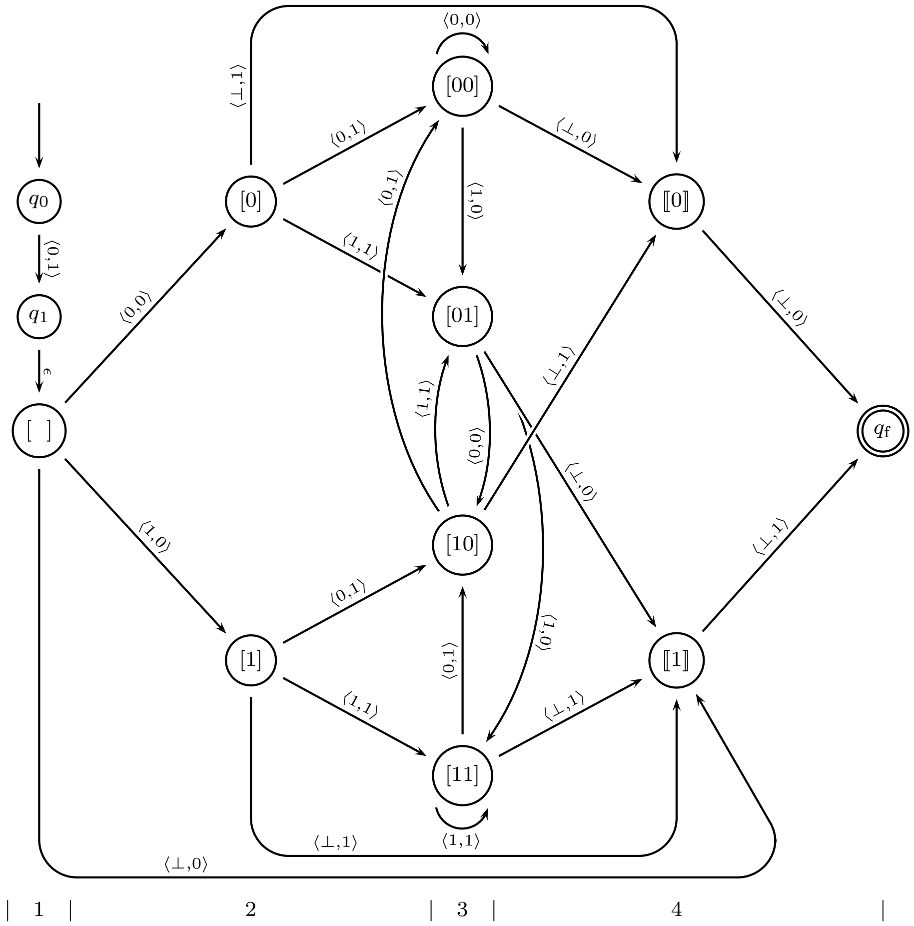


Figure 2: An example illustrating the proof of Lemma 9.

Lemma 10 *Given $p \geq 1$, the set $\{s_1 u \otimes \ldots \otimes s_p u \mid u \in \Sigma^*\}$ is recognizable by an automaton with size exponential in $\sum_{j=1}^{p} |s_j|$.*

Proof. By induction on p. The base case $p = 1$ is trivial, and the case $p = 2$ is proved in Lemma 9. Assume that $p \geq 2$ and that we have an automaton A with $T(A) = \{s_1 u \otimes \ldots \otimes s_p u \mid u \in \Sigma^*\}$ and one more word s_{p+1}. Let A'' be an automaton such that $T(A'') = \{s_p u \otimes s_{p+1} u \mid u \in \Sigma^*\}$. A'' may be obtained by applying Lemma 9 again. Clearly,

$$(T(A) \otimes \Sigma^*) \cap (\underbrace{\Sigma^* \otimes \ldots \otimes \Sigma^*}_{p-1} \otimes T(A'')) = \{s_1 u \otimes \ldots \otimes s_{p+1} u \mid u \in \Sigma^*\}.$$

According to Lemma 7, this language is recognizable by an automaton A', and, by Lemmas 7 and 9, $\|A'\|$ is of the order $2^{\sum_{j=1}^{p} |s_j|} * 2^{|s_p| + |s_{p+1}|}$. ⊠

Now we are ready to prove the Theorem 12 which says that, given p monadic Σ-terms s_i, the set of tuples of their ground instances

$$\{s_1\theta \otimes \ldots \otimes s_p\theta \mid \theta \text{ ground }\}$$

is recognizable by an automaton with size exponential in $\sum_{j=1}^{p} |s_j|$.

Proof. The terms s_i are either ground or have the form $s_i(x_i)$ with one occurrence of a variable x_i "at the end". Let, for any variable x occurring in any of the terms, $s_{i_1}, \ldots, s_{i_n}$ be those terms among the s_i which contain x. According to Lemma 10, the language

$$L_1^x = \{s_{i_1}\theta \otimes \ldots \otimes s_{i_n}\theta \mid \theta \text{ ground }\}$$

is recognizable by an automaton of size exponential in $\sum_j |s_{i_j}|$. From Lemma 7 we infer that $L_2^x = L_1^x \otimes \Sigma^* \otimes \ldots \otimes \Sigma^*$, with $p-n$ factors of Σ^*, is recognizable by an automaton with size exponential in $\sum_j |s_{i_j}|$. Finally, $L^x = (L_2^x)_\pi$, with π a permutation which maps the first n indices j to i_j, that is, puts the s_{i_j} into their right place in the sequence $1 \ldots p$, is also recognizable by an automaton of the same size, see Lemma 8. Moreover, it is not difficult to see that

$$L_g = \{t_1 \otimes \ldots \otimes t_p \mid t_i = s_i \text{ if } s_i \text{ ground, and } t_i \in \Sigma^*, \text{ otherwise}\}$$

is recognizable by an automaton with size polynomial in $\max |s_i|$. The desired language arises as the intersection of the languages L^x and L_g so that recognizability with the stated complexity bound follows. ⊠

For solving reachability constraints, we also need to recognize rewriting relations.

Theorem 13 ((Dauchet et al. 1988)) Given a ground rewrite system R on Σ^*, the set $\{u \otimes v \mid u \xrightarrow[R]{*} v\}$ is recognizable by an automaton the size of which is polynomial in the size of R.

Theorem 14 Rigid reachability in monadic signatures is in EXPTIME when the rules are ground.

Proof. Let (R, s, t) be a constraint over the monadic signature Σ. We show that the set of "solutions" $\mathcal{S} = \{s\theta \otimes t\theta \mid s\theta \xrightarrow[R]{*} t\theta\}$ is recognizable. s and t may contain at most one variable which we denote by x and y, respectively. These two variables may or may not be identical. Applying Theorem 12, we may infer that $\{s\theta \otimes t\theta \mid x\theta \text{ and } y\theta \text{ ground }\}$ is recognizable by an automaton A with size exponential in $|s| + |t|$. By Theorem 13, the set $\{u \otimes v \mid u, v \in \Sigma^*, u \xrightarrow[R]{*} v\}$ is recognizable by an automaton A' with size polynomial in the size of R. Clearly $\mathcal{S} = T(A) \cap T(A')$, and emptiness is decidable in time linear in the size of the corresponding intersection automaton (which is exponential in $|s| + |t|$ and polynomial in the size of R). ⊠

The extension to the simultaneous case of Theorem 14 generalizes and improves a result of (Gurevich & Voronkov 1997).

Theorem 15 Simultaneous rigid reachability in monadic signatures is decidable in EXPTIME when the rules are ground.

Proof. The construction is a generalization of the one for Theorem 14. Suppose we are given the system of constraints (R_i, s_i, t_i), $1 \leq i \leq n$. We first construct an automaton A_i for each $i \leq n$ such that $T(A_i) = \{u \otimes v \mid u, v \in \Sigma^*, u \xrightarrow[R_i]{*} v\}$. Then $A = \bigotimes_{i=1}^n A_i$ (see Lemma 7) recognizes the language:

$$T(A) = \{u_1 \otimes v_1 \otimes u_2 \otimes \ldots \otimes u_n \otimes v_n \mid \text{ for all } i \leq n, u_i, v_i \in \Sigma^*, u_i \xrightarrow[R_i]{*} v_i\}.$$

The size of A is the product of the sizes of the A_i, hence of order M^n where M is the maximum of the sizes of the A_i. In Theorem 12 we have shown that the language

$$L^G = \{s_1\theta \otimes t_1\theta \otimes \ldots \otimes s_n\theta \otimes t_n\theta \mid \theta \text{ ground}\}$$

is recognizable by an automaton A^G of size exponential in $\sum_i |s_i| + |t_i|$. The simultaneous reachability constraint is solvable if and only the intersection $L^G \cap T(A)$ is non-empty. According to the respective sizes of the automata in the above intersection, this gives an EXPTIME upper-bound for deciding simultaneous rigid reachability. ⊠

7 Conclusion

We have shown that absence of symmetry makes the solving of rigid reachability constraints in general much harder. In the non-simultaneous case one jumps from decidability to undecidability. In the case of ground rewrite rules, source terms with just a single variable, and ground target terms, the complexity increases from $\mathcal{P}$-completeness to EXPTIME-completeness. The undecidability of rigid reachability implies a new undecidability result for second-order unification problems with just a single second-order variable that occurs twice. We have also seen that automata-theoretic methods provide us with rather simple proofs of upper bounds in the monadic case.

Acknowledgments

The authors wish to thank the referees for their useful remarks and suggestions, which led to significant improvements of the results presented in this paper.

References

Baader, F. & Nipkow, T. (1998), *Term Rewriting and All That*, Cambridge University Press.

Coquidé, J., Dauchet, M., Gilleron, R. & Vágvölgyi, S. (1994), 'Bottom-up tree pushdown automata: classification and connection with rewrite systems', *Theoretical Computer Science* **127**, 69–98.

Coquidé, J.-L. & Gilleron, R. (1990), Proofs and reachability problem for ground rewrite systems, *in* 'Proc. of the 6th International Meeting of Young Computer Scientists', Vol. 464 of *LNCS*, pp. 120–129.

Cortier, V. (1998), Pspace-completeness of monadic simultaneous rigid reachability, Unpublished manuscript.

Dauchet, M. (1993), Rewriting and tree automata, *in* H. Comon & J. Jouannaud, eds, 'Term Rewriting (French Spring School of Theoretical Computer Science)', Vol. 909 of *Lecture Notes in Computer Science*, Springer Verlag, Font Romeux, France, pp. 95–113.

Dauchet, M., Heuillard, T., Lescanne, P. & Tison, S. (1988), The confluence of ground term rewriting systems is decidable, *in* 'Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh'.

Degtyarev, A. & Voronkov, A. (1995), Simultaneous rigid E-unification is undecidable, UPMAIL Technical Report 105, Uppsala University, Computing Science Department.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M. & Voronkov, A. (1998), The decidability of simultaneous rigid E-unification with one variable, *in* T. Nipkow, ed., 'Rewriting Techniques and Applications', Vol. 1379 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 181–195.

Dershowitz, N. & Jouannaud, J.-P. (1990), Rewrite systems, *in* J. Van Leeuwen, ed., 'Handbook of Theoretical Computer Science', Vol. B: Formal Methods and Semantics, North Holland, Amsterdam, chapter 6, pp. 243–309.

Doner, J. (1970), 'Tree acceptors and some of their applications', *Journal of Computer and System Sciences* **4**, 406–451.

Farmer, W. (1991), 'Simple second-order languages for which unification is undecidable', *Theoretical Computer Science* **87**, 25–41.

Frühwirth, T., Shapiro, E., Vardi, M. Y. & Yardemi, E. (1991), Logic programs as types for logic programs, *in* '6th IEEE Symp. Logic In Computer Science', pp. 300–309.

Gallier, J., Narendran, P., Plaisted, D. & Snyder, W. (1988), Rigid E-unification is NP-complete, *in* 'Proc. IEEE Conference on Logic in Computer Science (LICS)', IEEE Computer Society Press, pp. 338–346.

Gallier, J., Raatz, S. & Snyder, W. (1987), Theorem proving using rigid E-unification: Equational matings, *in* 'Proc. IEEE Conference on Logic in Computer Science (LICS)', IEEE Computer Society Press, pp. 338–346.

Goldfarb, W. (1981), 'The undecidability of the second-order unification problem', *Theoretical Computer Science* **13**, 225–230.

Gurevich, Y. & Veanes, M. (1998), Partisan corroboration, and shifted pairing, Research Report MPI-I-98-2-014, Max-Planck-Institut für Informatik.

Gurevich, Y. & Voronkov, A. (1997), Monadic simultaneous rigid E-unification and related problems, *in* P. Degano, R. Corrieri & A. Marchetti-Spaccamella, eds, 'Automata, Languages and Programming, 24th International Colloquium, ICALP'97', Vol. 1256 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 154–165.

Kozen, D. (1977), Complexity of finitely presented algebras, *in* 'Proc. 9th STOC', pp. 164–177.

Levy, J. (1998), Decidable and undecidable second-order unification problems, *in* T. Nipkow, ed., 'Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March/April 1998, Proceedings', Vol. 1379 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 47–60.

Levy, J. & Veanes, M. (1998), On the undecidability of second-order unification, Invited paper for a special issue of *Information and Computation* on term rewriting.

Seidl, H. (1994), 'Haskell overloading is dexptime-complete', *Inf. Process. Letters* **52**, 57–60.

Thatcher, J. & Wright, J. (1968), 'Generalized finite automata theory with an application to a decision problem of second-order logic', *Mathematical Systems Theory* **2**(1), 57–81.

Veanes, M. (1997), On computational complexity of basic decision problems of finite tree automata, Technical Report 133, Computing Science Department, Uppsala University.

Veanes, M. (1998), The relation between second-order unification and simultaneous rigid E-unification, *in* 'Proc. Thirteenth Annual IEEE Symposium on Logic in Computer Science, June 21–24, 1998, Indianapolis, Indiana (LICS'98)', IEEE Computer Society Press, pp. 264–275.