International Journal of Foundations of Computer Science

RIGID REACHABILITY
THE NON-SYMMETRIC FORM OF RIGID E-UNIFICATION

HARALD GANZINGER
Max-Planck-Institut für Informatik
Im Stadtwald, 66123 Saarbrücken, Germany

and

FLORENT JACQUEMARD*
LORIA and INRIA
615 rue du Jardin Botanique,
B.P. 101, 54602 Villers-les-Nancy Cedex, France

and

MARGUS VEANES*
Microsoft Research
Redmond, WA 98052 USA

Received
Revised
Communicated by

ABSTRACT

We show that rigid reachability, the non-symmetric form of rigid E-unification, is already undecidable in the case of a single constraint. From this we infer the undecidability of a new and 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 PSPACE-complete. Moreover, we identify two decidable non-monadic fragments that are complete for EXPTIME.

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 via $R\sigma$ to $t\sigma$. The term "rigid" refers to the fact that for no rule more than one instance can be used in the rewriting process. Simultaneous rigid reachability (SRR) 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

*This work was done while the author was staying at MPI Informatik.

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 [19] as "simultaneous rigid E-unification" (SREU). (Symmetric systems R arise, for instance, from orienting a given set of equations E in both directions.) It has been shown in [12] that simultaneous rigid E-unification is undecidable, whereas the non-simultaneous case with just one rigid equation to solve is NP-complete [18]. 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 R 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 [27].

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 for ground rewrite systems which is $\mathcal{P}$-complete in the equational case becomes EXPTIME-complete when symmetry is absent.

Another decidability result which we prove in section 6.2 is that SRR with ground rules is EXPTIME-complete for "balanced" systems of reachability constraints. Balanced systems include, in particular, cases where all occurrences of each variable are at the same depth. On the other hand, if variable depths are slightly non-balanced — for instance, when all variables occur at the same depth except for one occurrence of a variable, — the problem becomes undecidable. The decidability result for balanced systems generalizes the related result by Degtyarev, Gurevich, Narendran, Veanes and Voronkov [9] for the up to now largest decidable fragment of SREU with ground rules and implies EXPTIME-completeness of that fragment (which was left open in [9]). For obtaining the decidability results we employ tree automata techniques for product languages in a way similar to their use in chapter 3 of [3].

The, arguably, most difficult remaining open problem regarding SRR and SREU is the decidability of the monadic case where all non-constant function symbols are unary. This fragment is important because of its close relation to word equations [11], and to fragments of intuitionistic logic [13]. What is known about monadic SREU in general is that it reduces to a nontrivial extension of word equations [24]. In the case of ground rules, the decidability of monadic SREU was established in [24] by reducing it to "word equations with regular constraints". The decidability of the latter problem is an extension of Makanin's [29] result by Schulz [32]. Conversely, word equations reduce in polynomial time to monadic SREU [11].

In Section 5 we show that monadic SRR with ground rules is in PSPACE, improving over the EXPTIME result that we have obtained earlier [20]. (The PSPACE-hardness of monadic SREU with ground rules was already shown by Goubault [22].) We conjecture that there is no simple reduction, from monadic SREU to monadic SREU with ground rules, as otherwise one would get a very simple proof for decidability of word unification, compared to Makanin's [29] orig-

inal proof. On the other hand, recent results show that word unification is in PSPACE [30], and it might even be in $\mathcal{NP}$, so that from a complexity-theoretic point of view a reduction is not impossible.

For obtaining the PSPACE result we apply an extension of the intersection nonemptiness problem of a sequence of finite automata that we prove to be in PSPACE. Moreover, using the same proof technique, we can show that simultaneous rigid reachability with ground rules remains in PSPACE, even when just the rules are required to be monadic. Furthermore, in this case PSPACE-hardness holds already for a single constraint with one variable, contrasting the fact that SREU with one variable is solvable in polynomial time [10].

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, a 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 *term language* or simply *language* is a triple $L = (\Sigma, \mathcal{X}, \mathcal{F})$ where (i) Σ is a signature, (ii) $\mathcal{X}$ (with elements denoted by $x, y, x_1, y_1, \ldots$) is a collection of first-order variables, and (iii) $\mathcal{F}$ (with elements denoted by $F, G, F_1, F', \ldots$) is a collection of symbols with fixed arities ≥ 1, called *second-order variables*. The various sets of symbols are assumed to be pairwise disjoint. L is *first-order*, if $\mathcal{F}$ is empty, and L is called *second-order*, otherwise. L is *monadic* if all function symbols in Σ 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 L is denoted by $\mathcal{T}_\Sigma$. A term is called *shallow* if all its variables occur at depth ≤ 1. The size $\|t\|$ of a term t is the number of nodes in its tree representation.

We assume that the reader is familiar with the basic concepts in term rewriting [14, 1]. 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. In case the position p of a subterm occurrence needs to be emphasized, we will use the notation $u[s]_p$. 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 $s \to t$. An equation or a rule is *ground* if its terms are ground. A *(rewrite) system* is a finite set of rewrite rules. 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 $l \to r$ 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 $l \to r$ in R, R also contains its converse $r \to l$. 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 the terms in its rules.

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.

Tree Automata. Tree automata are a generalization of classical automata [15, 34]. Under the rewriting-based view *e.g.* [5, 7] a *(finite bottom-up) tree automaton* (TA) A is a quadruple (Q, Σ, R, F), where (i) Q is a finite set of constants called *states*, (ii) Σ is a finite *signature* that is disjoint from Q, (iii) R is a system of *rules* of the form $f(q_1, \ldots, q_n) \to q$, where $f \in \Sigma$ has arity $n \geq 0$ and $q, q_1, \ldots, q_n \in Q$, and (iv) $F \subseteq Q$ is the set of *final states*. When Q, Σ, R and F are not specified, we denote them respectively Q_A, Σ_A, R_A and F_A. The *size* of a TA A is $\|A\| = |Q| + |\Sigma| + \|R\|$.

We denote by $L(A, q)$ the set $\{t \in \mathcal{T}_\Sigma \mid t \xrightarrow[R]{*} q\}$ of ground terms *accepted* by A in state q. The set of terms *recognized* by the TA A is the set $\bigcup_{q \in F} L(A, q)$.
A set of terms is called *recognizable* or *regular* if it is recognized by some TA. A *monadic* TA is a TA over a monadic signature.

String Automata. For monadic signatures, we use the traditional, equivalent concepts of alphabets, strings (or words), finite automata, and regular expressions. We will identify an NFA A with alphabet Σ with the set of all rules $a(q) \to p$, also written as $q \xrightarrow[A]{a} p$, where there is a transition with label $a \in \Sigma$ from state q to state p in A, and we denote this set of rules also by A. A monadic term $a_1(a_2(\ldots a_n(q)))$ is written, using the *reversed Polish notation*, as the string $qa_n \ldots a_1$.

Then A *accepts a string* $a_1 a_2 \cdots a_n$ if and only if, for some final state q and the initial state q_0 of A, $a_n(\cdots a_2(a_1(q_0)) \cdots) \xrightarrow[A]{*} q$. The set of all strings accepted by A is denoted by $L(A)$.

Product Automata. Let Σ be a signature, m a positive integer, and $\perp$ a new constant. We write $\Sigma_\perp$ for $\Sigma \cup \{\perp\}$ and $\Sigma_\perp^m$ denotes the signature consisting of, for all $f_1, f_2, \ldots, f_m \in \Sigma_\perp$, a unique function symbol $\langle f_1 f_2 \cdots f_m \rangle$ with arity equal to the maximum of the arities of the f_i's.

Let $t_i \in \mathcal{T}_\Sigma \cup \{\perp\}$, $t_i = f_i(t_{i1}, \ldots, t_{ik_i})$, where $k_i \geq 0$, for $1 \leq i \leq m$. Let k be the maximum of all the k_i and let $t_{ij} = \perp$ for $k_i < j \leq k$. The *product* $t_1 \otimes \cdots \otimes t_m$ of $t_1, \ldots, t_m$ is defined by recursion on the subterms:

$$\begin{aligned} t_1 \otimes \cdots \otimes t_m &= \langle f_1 f_2 \cdots f_m \rangle (t_{11} \otimes \cdots \otimes t_{m1}, \ldots, t_{1k} \otimes \cdots \otimes t_{mk}) \text{ if } k > 0 \\ &= \langle t_1 t_2 \cdots t_m \rangle \text{ otherwise} \end{aligned} \quad (1)$$

For example:

$$\begin{aligned} f(c, g(c)) \otimes f(g(d), f(c, g(c))) &= \langle ff \rangle (c \otimes g(d), g(c) \otimes f(c, g(c))) \\ &= \langle ff \rangle (\langle cg \rangle (\bot \otimes d), \langle gf \rangle (c \otimes c, \bot \otimes g(c))) \\ &= \langle ff \rangle (\langle cg \rangle (\langle \bot d \rangle), \langle gf \rangle (\langle cc \rangle, \langle \bot g \rangle (\bot \otimes c))) \\ &= \langle ff \rangle (\langle cg \rangle (\langle \bot d \rangle), \langle gf \rangle (\langle cc \rangle, \langle \bot g \rangle (\langle \bot c \rangle))) \end{aligned}$$

We write ${\mathcal{T}_\Sigma}^m$ for the set of all t in $\mathcal{T}_{\Sigma_\bot^m}$ such that $t = t_1 \otimes \cdots \otimes t_m$ for some $t_1, \ldots, t_m \in \mathcal{T}_\Sigma \cup \{\bot\}$. If $s \in {\mathcal{T}_\Sigma}^m$ and $t \in {\mathcal{T}_\Sigma}^n$, where $s = s_1 \otimes \cdots \otimes s_m$ and $t = t_1 \otimes \cdots \otimes t_n$, then $s \otimes t$ denotes the term $s_1 \otimes \cdots \otimes s_m \otimes t_1 \otimes \cdots \otimes t_n$ in ${\mathcal{T}_\Sigma}^{m+n}$. Given a sequence $\vec{t} = t_1, \ldots, t_m$ of terms in $\mathcal{T}_\Sigma \cup \{\bot\}$, we write $\bigotimes \vec{t}$ for the product term $t_1 \otimes \cdots \otimes t_m$

Given two automata A_1 and A_2 over $\Sigma_\bot^m$ and $\Sigma_\bot^n$, respectively, the *product* of A_1 and A_2 is an automaton $A_1 \otimes A_2$ over $\Sigma_\bot^{m+n}$ such that

$$L(A_1 \otimes A_2) = L(A_1) \otimes L(A_2) = \{t_1 \otimes t_2 : t_1 \in L(A_1), t_2 \in L(A_2)\}$$

The construction of $A_1 \otimes A_2$ is straightforward, with a state $q_{(q_1,q_2)}$ for all states q_1 in A_1 and q_2 in A_2, see *e.g.* [3]. In general, $\bigotimes_{i=1}^n A_i$ is defined accordingly.

We will use the following construction of Dauchet, Heuillard, Lescanne and Tison [8] in our proofs.

Lemma 1 (Dauchet, Heuillard, Lescanne and Tison [8]) *Let R be a ground rewrite system over a signature Σ. There is a TA A such that $L(A) = \{s \otimes t : s, t \in \mathcal{T}_\Sigma,\ s \stackrel{*}{\underset{R}{\to}} t\}$ that can be constructed in polynomial time from R and Σ.*

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 [21] and Farmer [16], 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:

(i) If $s = x$ and $x \in \mathrm{dom}(\theta)$ then $s\theta = \theta(x)$.
(ii) If $s = x$ and $x \notin \mathrm{dom}(\theta)$ then $s\theta = x$.
(iii) If $s = F(t_1, \ldots, t_n)$ and $F \in \mathrm{dom}(\theta)$ then $s\theta = \theta(F)[t_1\theta, \ldots, t_n\theta]$.
(iv) If $s = F(t_1, \ldots, t_n)$ and $F \notin \mathrm{dom}(\theta)$ then $s\theta = F(t_1\theta, \ldots, t_n\theta)$.
(v) 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.

3. Rigid Reachability is Undecidable

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

Lemma 2 (Gurevich and Veanes [23]) *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* Σ_{mv} *and* Σ_{id} *is a binary function symbol* .,[a] *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* $t_{\mathrm{id}} \,.\, s \xrightarrow[\Pi_2]{*} t$.

The main idea behind the proof of Lemma 2 is illustrated in Figure 1. In the rest of this section, we consider fixed A_{mv}, A_{id}, Π_1 and Π_2 as given by Lemma 2.

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. 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

[a] We write . ("dot") as an infix operator.

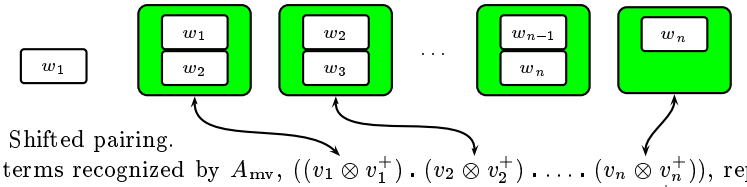

Figure 1: Shifted pairing.

The terms recognized by A_{mv}, $((v_1 \otimes v_1^+) \,.\, (v_2 \otimes v_2^+) \,.\, \ldots \,.\, (v_n \otimes v_n^+))$, represent a sequence of independent 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 $(w_1 \,.\, w_2 \,.\, \ldots \,.\, w_n)$.

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 $t_{\mathrm{id}} \,.\, s$ 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}.

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 3 *Let m, n, k and l be pairwise 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* $t_{\mathrm{id}} \,.\, s \xrightarrow[\Pi_2]{*} t$.

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

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:

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

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:

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

Our goal is to show that solvability of (2), for a given $t_{\mathrm{id}} \in \Sigma_{\mathrm{id}}$, is equivalent to the existence of s and t satisfying the condition in Lemma 2. 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 (2).

Lemma 4 *For every substitution θ, θ solves the constraint (2) if and only if θ solves the system (3)–(6) of constraints.*

$$\left.\begin{array}{llll} (\ R_{\mathrm{mv}}^{(0)}, & x_0, & q_{\mathrm{mv}}^{(0)}\ ) \\ (\ R_{\mathrm{mv}}^{(2)}, & x_2, & q_{\mathrm{mv}}^{(2)}\ ) \\ (\ {\Sigma_{\mathrm{mv}}}^{(0,1)}, & x_0, & x_1\ ) \\ (\ {\Sigma_{\mathrm{mv}}}^{(2,1)}, & x_2, & x_1\ ) \end{array}\right\} \quad (3)$$

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

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

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

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 (3) and (5). The other cases are symmetrical. Now let us assume that θ solves (2).

θ solves (3): We first 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 (the 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 $s \in \mathcal{T}_{\Sigma_{\rm mv}^{(0)} \cup Q_{\rm mv}^{(0)}}$. Hence $l \to r \in R_{\rm mv}^{(0)}$, and consequently $t \xrightarrow[R_{\rm mv}^{(0)}]{*} q_{\rm mv}^{(0)}$.

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

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

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

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

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

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

θ solves (5): We know that $y_4\theta \xrightarrow[R_{\rm u}]{*} y_5\theta$ and $x_0\theta \xrightarrow[R_{\rm u}]{*} y_5\theta$. We first prove that $y_4\theta \xrightarrow[\Sigma_{\rm id}{}^{(4,5)}]{*} y_5\theta$. Suppose, to the contrary, that

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

where $l \to r \in R_{\rm u} \setminus \Sigma_{\rm id}{}^{(4,5)}$. Then either $l \to r \in R_{\rm id}^{(4)}$ or $l \to r \in \Sigma_{\rm id}{}^{(4,3)}$. The former case implies that $Con(y_5\theta) \cap Q_{\rm id}^{(4)} \neq \emptyset$ and the latter case implies that $Con(y_5\theta) \cap \Sigma_{\rm id}^{(3)} \neq \emptyset$. Both cases contradict that $x_0\theta \xrightarrow[R_{\rm u}]{*} y_5\theta$, 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 $Q_{\rm mv}^{(0)}$ to $y_5\theta$ or a constant with label 1 to $y_5\theta$, in both cases contradicting that $y_4\theta \xrightarrow[\Sigma_{\rm id}{}^{(4,5)}]{*} y_5\theta$. □

The following lemma relates the solvability of (2) to the Lemma 2.

Lemma 5 *For $t_{\rm id} \in \mathcal{T}_{\Sigma_{\rm id}}$, the constraint (2) is solvable if and only if there exists $s \in T(A_{\rm mv})$ and $t \in T(A_{\rm id})$ such that $s \xrightarrow[\Pi_1]{*} t$ and $t_{\rm id} \cdot s \xrightarrow[\Pi_2]{*} t$.*

Proof. ($\Leftarrow$) Assuming that we are given s and t with the required properties, we 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 3 and Lemma 4 that θ solves (2).

($\Rightarrow$) Assume that θ solves (2). By Lemma 4, θ solves (3)–(6). First we observe the following facts.

(i) With θ solving (3), there exists $s \in T(A_{\rm mv})$ such that $x_0\theta = s^{(0)}$ and $x_2\theta = s^{(2)}$.

(ii) With θ solving (4), there exists $t \in T(A_{\rm id})$ such that $y_4\theta = t^{(4)}$ and $y_6\theta = t^{(6)}$.

From θ solving (5) and by using (ii), it follows that $y_5\theta = t^{(5)}$. Now, due to the second component of (5) and by using (i), we may infer that $s^{(0)} \xrightarrow[\Pi_1{}^{(0,5)}]{*} t^{(5)}$.

From θ solving (6) and by using (ii), it follows that $y_7\theta = t^{(7)}$. Now, due to the second component of (6) and by using (i), we conclude that $t_{\mathrm{id}}^{(7)} \centerdot s^{(2)} \xrightarrow[\Pi_2^{(2,7)}]{*} t^{(7)}$. Now the result follows from Lemma 3. □

Theorem 1 *Rigid reachability is undecidable. More specifically, there exists a terminating ground rewrite system R, and a term t such that the solvability of constraints of the form (R, s, t), where s and t do not share any variables, is undecidable.*

Proof. The undecidability follows from Lemma 2 and Lemma 5, taking (R, s, t) to be the constraints of the form (2) above, with R_{mv} representing the moves of a universal Turing machine. It is not difficult to show that R_{u} is terminating. □

We have not attempted to minimize the number of variables in the constraints (2). Observe also that all but one of the occurrences of variables in (2) are shallow (the target term is shallow).

4. An Application to Second-Order Unification

As a direct application of the previous result, we prove that second-order unification is already undecidable for unification problems containing just a single second-order variable which, in addition, occurs only twice. This result contrasts a claim to the opposite in [27]. 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 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 (2) is solvable:

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

Lemma 6 *Given $t_{\mathrm{id}} \in \mathcal{T}_{\Sigma_{\mathrm{id}}}$, (2) is solvable if and only if (7) is solvable.*

Proof. The direction '$\Rightarrow$' follows from [35, Lemma 2] and the observation that if θ solves (2) 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 now prove the converse direction. Assume that θ solves (7). We show that θ solves (2). A straightforward inductive argument shows that $F_{\mathrm{u}}\theta$ is an L_1^*-term of rank $m+1$ of the form (recall that z_i denotes is the i'th bound variable of a function)

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

for some $k \geq 1$, by using that $R_{\rm u}$ is ground and that $\underline{\rm cons} \notin \Sigma_{\rm u}$ (see [35, Lemma 1]). Hence, since θ solves (7), it follows that

$$\begin{array}{llll} \underline{\rm cons}(s_1[\vec{l}_{\rm u}, t'], & \ldots & \underline{\rm cons}(s_{i+1}[\vec{l}_{\rm u}, t'], & \ldots \ \underline{\rm cons}(t\theta, \quad \underline{\rm nil}) \cdots) \cdots) = \\ \underline{\rm cons}(s\theta, & \ldots & \underline{\rm cons}(s_i[\vec{r}_{\rm u}, \underline{\rm nil}], & \ldots \ \underline{\rm cons}(s_k[\vec{r}_{\rm u}, \underline{\rm nil}], \ \underline{\rm nil}) \cdots) \cdots), \end{array} \tag{8}$$

where s is the source term of (2), t is the target term of (2), and $t' = \underline{\rm cons}(t\theta, \underline{\rm nil})$. Therefore, there exists a reduction in $R_{\rm u} \cup \{t' \to \underline{\rm nil}\}$ of the following form:

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

In other words, $s\theta \xrightarrow[R_{\rm u} \cup \{t' \to \underline{\rm nil}\}]{*} t\theta$, that is,

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

Next we show that $x_0\theta, x_2\theta, y_4\theta, y_6\theta \in \mathcal{T}_{\Sigma_{\rm u}}$. We observe that

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

It follows by induction on the length of reductions that $t' \to \underline{\rm nil}$ can not be used in these rewritings, since $\underline{\rm nil}$ does not not occur in $R_{\rm u}$. Hence, $x_0\theta, x_2\theta, y_4\theta, y_6\theta \in \mathcal{T}_{\Sigma_{\rm u}}$. This implies that $s\theta$ is in $\mathcal{T}_{\Sigma_{\rm u}}$ so that the rule $t' \to \underline{\rm nil}$ can not be used in the reduction of $s\theta$ to $t\theta$. □

We conclude with the following result, that follows from Lemma 2, Lemma 5, and Lemma 6.

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

The presence of first-order variables in the unification problems is essential for obtaining the undecidability result. 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* [28], and thus is decidable.

5. Monadic Rigid Reachability

In the remainder of the paper we will identify restricted, decidable cases of SRR. The restrictions will be defined by syntactic criteria on either the signature or the form of the source and target terms in constraints. In all cases that we prove to be decidable, the rewrite rules have to be ground. We will start by proving that monadic SRR with ground rules is PSPACE-complete. Our main tool is a decision problem of NFAs that we define next. In this section we consider only *monadic* signatures. The Section 6 will exhibit decidable fragments over non-monadic signatures.

5.1. Constrained Product Non-Emptiness of NFAs

Given a signature Σ and a positive integer m, we want to select only a certain subset from $\Sigma_\perp^m$ through *selection constraints (bounded by m)*. These are unordered pairs of indices written as $i \approx j$, where $1 \leq i, j \leq m$, $i \neq j$. Given a signature Σ and a set I of selection constraints, we write $\Sigma_\perp^m \lfloor I$ for the following subset of $\Sigma_\perp^m$:

$$\Sigma_\perp^m \lfloor I \quad = \quad \left\{ \langle a_1 a_2 \cdots a_m \rangle \in \Sigma_\perp^m \quad : \quad (\forall i \approx j \in I)\ a_i = a_j \right\}$$

For an automaton A, let $A \lfloor I$ denote the reduction of A to the alphabet $\Sigma_\perp^m \lfloor I$. We write also $L(A) \lfloor I$ for $L(A \lfloor I)$. The automaton $A \lfloor I$ has the same states as A, and the transitions of $A \lfloor I$ are precisely all the transitions of A with labels from $\Sigma_\perp^m \lfloor I$.

We consider the following decision problem, that is closely related to the non-emptiness problem of the intersection of a sequence of NFAs. Consider an alphabet Σ. Let $(A_i)_{1 \leq i \leq n}$, $n \geq 1$, be a sequence of (string product) NFAs over the alphabets $\Sigma_\perp^{m_i}$ for $1 \leq i \leq n$, respectively. Let m be the sum of all the m_i and let I be a set of selection constraints. The *constrained product non-emptiness problem of NFAs* is, given $(A_i)_{1 \leq i \leq n}$, and I, to decide if $(\bigotimes_{i=1}^n L(A_i)) \lfloor I$ is nonempty. A key lemma is given next. Its proof is a straightforward extension of the Kozen's [26] PSPACE-completeness result of the intersection non-emptiness problem of DFAs.

Lemma 7 *Constrained product non-emptiness of NFAs (or monadic TAs) is in* PSPACE.

Proof. Let $(A_i)_{1 \leq i \leq n}$, m_i, m, Σ, and I be given as above. Assume, for simplicity, that $m_i = 2$ for $1 \leq i \leq n$, *i.e.*, $m = 2n$ and that each automaton has alphabet $\Sigma_\perp^2$. We may also assume, without loss of generality, that none of the automata accepts the empty string and that, whenever a string v is accepted by A_i also $\langle \perp\perp \rangle v$ is accepted by A_i. Consider the following nondeterministic decision procedure.

I: Initialize

Calculate the number of states in $\bigotimes_{i=1}^n A_i$, which is the product of the number of states in the individual A_i, and save it in `IterationLimit`.

Save in $\mathtt{State}_i$ the initial state of A_i for $1 \leq i \leq n$.

II: Guess the next letter

Select $(a_1, \ldots, a_m) \in \Sigma_\perp^m \lfloor I$ and store a_i in $\mathtt{Letter}_i$.

III: Guess the next transition

For $1 \leq i \leq n$, guess nondeterministically a state $\mathtt{q}_i$ from A_i.

Check that, for $1 \leq i \leq n$, there is a $\langle \mathtt{Letter}_{2i-1}\mathtt{Letter}_{2i} \rangle$-transition in A_i from $\mathtt{State}_i$ to $\mathtt{q}_i$, and if so, save $\mathtt{q}_i$ in $\mathtt{State}_i$. If there is no such transition then terminate and **reject**.

IV: Check acceptance

If, for $1 \leq i \leq n$, $\mathtt{State}_i$ is an accepting state of A_i then terminate and **accept**.

V: Iterate

If `IterationLimit` is 0 then terminate and **reject**, else decrease `IterationLimit` by one and return to Step II.

The procedure traverses the graph of $(\bigotimes_{i=1}^{n} A_i)\lfloor I$, by starting from the initial state, at each step just remembering the current state and guessing a valid transition from that state to the next state. We only need to check if there exists a path of at most `IterationLimit` transitions (as initialized in Step I) in $L(\bigotimes_{i=1}^{n} A_i)\lfloor I$ from the initial state to a final state. It is evident that the procedure always terminates, and that it accepts if and only if $L(\bigotimes_{i=1}^{n} A_i)\lfloor I$ is nonempty.

It is obvious that no more than polynomial space is required for the execution of the procedure. In particular, via the usual binary encoding of numbers, the iteration limit can be calculated in polynomial space. Hence, the procedure runs in non-deterministic polynomial space and thus in PSPACE, by using the result of Savitch [31].

Finally, note that the only difference between NFAs and monadic TAs is that in the latter we may have several transitions of the form $c \to q$, where c is a constant and q a state. This corresponds roughly to allowing several initial states in NFAs. □

The proof of Lemma 7 can be extended in a straightforward manner to finite tree automata. The only difference will be that the algorithm will do "universal choices" when the arity of function symbols (letters) in the component automata is > 1. This leads to *alternating* PSPACE, and thus, by the result of Chandra, Kozen and Stockmeyer [2], to EXPTIME upper bound for the constrained product non-emptiness problem of TAs.

Although we will not use this fact, it is worth noting that the constrained product non-emptiness problem is also PSPACE-hard, and this so already for DFAs (or monadic DTAs). It is easy to see that $\bigcap_{i=1}^{n} L(A_i)$ is nonempty if and only if $L(\bigotimes_{i=1}^{n} A_i)\lfloor\{i \approx i+1 : 1 \leq i < n\}$ is nonempty.

5.2. Monadic SRR with Ground Rules is in PSPACE

We need the following notion of normal form of a system of reachability constraints. We say that a system S of reachability constraints is *flat*, if each constraint in S is either of the form

- (R, x, t), R is nonempty, x is a variable, and t is a ground term or a variable distinct from x, or of the form
- $(\emptyset, x, f(y))$, where x and y are distinct variables and f is a unary function symbol.

Note that the solvability of a reachability constraint with empty rule set is simply the unifiability of the source and the target.

Lemma 8 *Let S be a system of reachability constraints. There is a flat system which can be obtained in polynomial time from S, and that is solvable if and only if S is solvable.*

Proof. Let S be a given system of reachability constraints and consider the following procedure.

(i) Replace each constraint (R, s, t), such that s is not a variable, or $s = t$, by the two constraints (R, x, t) and $(\emptyset, x, s)$, where x is a new variable.

(ii) Replace each constraint (R, x, t), where R is nonempty, x is a variable and t is neither ground nor a variable, by the constraints (R, x, y) and $(\emptyset, y, t)$, where y is a new variable.

(iii) Replace each constraint $(\emptyset, x, f(s))$, where s is not a variable and not ground, by the constraints $(\emptyset, x, f(y))$ and $(\emptyset, y, s)$, where y is a new variable.

(iv) Repeat the above steps until the system is flat.

It is easy to check that each step preserves solvability, and clearly, the time complexity of this procedure is polynomial in the size of S. □

By using the lemmas 7 and 8 we can now show the following theorem which is the main result of this section.

Theorem 3 *Monadic SRR with ground rules is* PSPACE*-complete.*

Proof. The PSPACE-hardness has been proved already for the special case where the rule sets are symmetric [22] and where there is only one variable [24]. We prove membership in PSPACE by giving a polynomial time reduction to the constrained product non-emptiness problem of NFAs.

Let S be a system of reachability constraints with ground rules. Let Σ be the signature of S. We may assume, by using Lemma 8, that S is flat. Enumerate all the constraints in S as $\rho_1, \ldots, \rho_m, \rho_{m+1}, \ldots, \rho_n$, such that the constraints of the form $(\emptyset, x, f(y))$ occur as $\rho_{m+1}, \ldots, \rho_n$. Let $\rho_i = (R_i, x_i, t_i)$ for $1 \leq i \leq m$ and $\rho_i = (\emptyset, x_i, f_i(y_i))$ for $m < i \leq n$.

For $1 \leq i \leq m$, using Lemma 1, construct (in polynomial time) an NFA A_i such that,

$$L(A_i) \quad = \quad \{x_i\theta \otimes t_i\theta \,:\, \theta \text{ solves } \rho_i\}.$$

For $m < i \leq n$, construct an NFA A_i such that

$$L(A_i) \quad = \quad \{x_i\theta \otimes y_i\theta \,:\, \theta \text{ solves } \rho_i\} \quad [= \{f_i(s) \otimes s \,:\, s \in \mathcal{T}_\Sigma\}].$$

This construction is exemplified in Figure 2 and can be done in polynomial time.

Let now I be the set of the following selection constraints (where $1 \leq i, j \leq n$ and $i \neq j$):

(i) If the source of a ρ_i is a variable that occurs as the source of a ρ_j, then $2i - 1 \approx 2j - 1 \in I$.

(ii) If the source of a ρ_i is a variable that occurs in the target of a ρ_j, then $2i - 1 \approx 2j \in I$.

(iii) If the target of a ρ_i is a variable that occurs in the target of a ρ_j, then $2i \approx 2j \in I$.

Clearly, $L(\bigotimes_{i=1}^{n} A_i)\downarrow I$ is nonempty if and only if S is solvable. With this, the theorem follows from Lemma 7. □

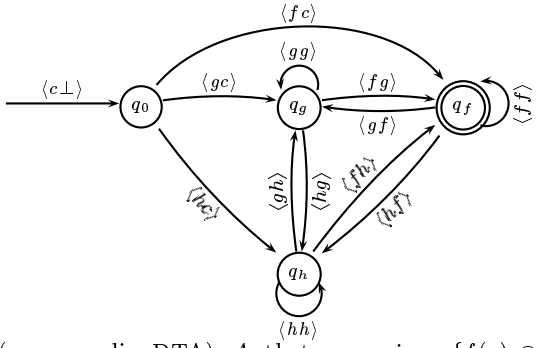

Figure 2: A DFA (or monadic DTA) A that recognizes $\{f(s) \otimes s \ : \ s \in \mathcal{T}_\Sigma\}$, where Σ consists of the unary function symbols f, g, and h, and the constant c. For example A recognizes the string $\langle c\bot\rangle\langle gc\rangle\langle gg\rangle\langle hg\rangle\langle fh\rangle$, *i.e.*, the term $\langle fh\rangle(\langle hg\rangle(\langle gg\rangle(\langle gc\rangle(\langle c\bot\rangle))))$ that is the same as $f(h(g(g(c)))) \otimes h(g(g(c)))$.

The crucial step in the proof of Theorem 3 is the construction of an automaton that recognizes the language $\{f(s) \otimes s \ : \ s \in \mathcal{T}_\Sigma\}$ (cf. Figure 2). The reason why the proof does *not* generalize to TAs is that the language $\{f(s) \otimes s \ : \ s \in \mathcal{T}_\Sigma\}$ is not regular for non-monadic signatures. The next example illustrates how the reduction in the proof of Theorem 3 works.

Example 1 Consider the flat system $S = \{\rho_1, \rho_2, \rho_3\}$ with $\rho_1 = (R, y, x)$, $\rho_2 = (\emptyset, y, f(z))$ and $\rho_3 = (\emptyset, z, g(x))$, over a signature $\Sigma = \{f, g, c\}$, where c is a constant and R is ground. This system is solvable if and only if the constraint $(R, f(g(x)), x)$ is solvable.

The construction in the proof of Theorem 3 gives us the NFAs A_1, A_2 and A_3 such that

$$\begin{aligned} L(A_1) &= \{s \otimes t : s \xrightarrow[R]{*} t, s, t \in \mathcal{T}_\Sigma\}, \\ L(A_2) &= \{f(s) \otimes s : s \in \mathcal{T}_\Sigma\}, \\ L(A_3) &= \{g(s) \otimes s : s \in \mathcal{T}_\Sigma\}, \end{aligned}$$

and a set $I = \{1 \approx 3, 5 \approx 4, 6 \approx 2\}$ of selection constraints. So $L(\bigotimes_{i=1}^{3} A_i)\lfloor I$ is as follows.

$$\begin{aligned} &L(A_1 \otimes A_2 \otimes A_3)\lfloor I = \\ &\quad \left\{s \otimes t \otimes f(u) \otimes u \otimes g(v) \otimes v : s, t, u, v \in \mathcal{T}_\Sigma, \ s \xrightarrow[R]{*} t\right\}\lfloor\{1 \approx 3, 5 \approx 4, 6 \approx 2\} \\ &= \left\{s \otimes t \otimes f(u) \otimes u \otimes g(v) \otimes v : s, t, u, v \in \mathcal{T}_\Sigma, \ s \xrightarrow[R]{*} t, \ s = f(u), \ g(v) = u, \ v = t\right\} \\ &= \left\{f(g(t)) \otimes t \otimes f(g(t)) \otimes g(t) \otimes g(t) \otimes t : t \in \mathcal{T}_\Sigma, \ f(g(t)) \xrightarrow[R]{*} t\right\} \end{aligned}$$

Hence, solvability of S is equivalent to non-emptiness of $L(A_1 \otimes A_2 \otimes A_3)\lfloor I$.

If only the rules are (ground and) monadic but the source and target terms are arbitrary, SRR remains decidable and in PSPACE. Furthermore, using the intersection non-emptiness problem for DFAs one may easily show that PSPACE-hardness of this fragment holds already for a *single* constraint with *one* variable. This is in contrast with the fact that SREU with one variable and a fixed number of constraints can be solved in polynomial time [10].

6. Decidable Non-Monadic Cases

We show that rigid reachability and simultaneous rigid reachability are decidable for arbitrary signatures if the rules are ground and if the source and target terms are suitably restricted. We will consider two kinds of restrictions. In the next section we consider the case where either the source s or the target t of a constraint are linear, and where s and t are *variable-disjoint*, that is, $Var(s) \cap Var(t) = \emptyset$. This fragment 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 [10]. When, additionally, both the source and target terms are linear, then rigid reachability and simultaneous rigid reachability are both $\mathcal{P}$-complete.

In the section 6.2 we will show EXPTIME completeness for the case of balanced constraints which embeds the case where non-linear variables have to occur at the same depth.

6.1. Linear and Variable-Disjoint Sources and Targets

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, ... ,A_n. This intersection emptiness problem is known to be EXPTIME-complete, see [17], [33] and [36]. We may assume the state 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 given signature Σ by a new symbol f of arity n, and assume that $n > 1$.

Lemma 9 *$L(A_1) \cap \ldots \cap L(A_n) \neq \emptyset$ if and only if, the constraint $\big(R_{A_1} \cup \ldots \cup R_{A_n},\ f(x, \ldots, x),\ f(q^{\mathrm{f}}_{A_1}, \ldots, q^{\mathrm{f}}_{A_n})\big)$ has a solution.*

Proof. ($\Rightarrow$) is obvious. For ($\Leftarrow$) we use the fact that the new symbol f 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 facts that $n > 1$ and that the state sets are disjoint make it impossible for states of the automata to appear in $x\theta$. □

Theorem 4 *Rigid reachability is* EXPTIME*-hard even when the rules and the target are ground and the source contains only a single variable.*

For obtaining an EXPTIME upper bound for a somewhat less restrictive case of rigid reachability we will now apply certain 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 [8].

Proposition 1 (Coquidé and Gilleron [4]) *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 of size in $O(\|t\| * \|R\|^2)$.*

Proposition 2 *The subset of $\mathcal{T}_\Sigma$ of ground instances of a given linear term s is recognizable by a tree automaton A_s of size linear in the size of s.*

Theorem 5 *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 upper time bound becomes $O(n^4)$ when the source term is linear, since $k = 0$ in this case.

Proof. Assume to be given a 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 1, recognizing the predecessors with respect to R of the ground instances of t. The size of A is in $O(n^3)$.

If the source s is linear, then there is a solution for (R, s, t) iff $L(A) \cap L(A_s) \neq \emptyset$, where A_s is a tree automaton accepting the ground instances of s, cf. Proposition 2. Since the intersection of recognizable languages is recognizable by a tree automaton whose size is the product of the sizes of the given tree automata, the solvability of the constraint can be decided in time $O(\|s\| * n^3) \subseteq O(n^4)$.

If the source s is not linear, we reduce the problem to $|Q_A|^k$ problems of the above type. We assume wolog 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 $\cap_{l \leq j} L(A, q_l) \neq \emptyset$,[b] and we replace the l-th occurrence of x in s by q_l, for $l \leq j$.

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) $\Rightarrow$ (ii): Assume that σ is a solution of the constraint (R, s, t). This means in particular that $s\sigma \in L(A)$ *i.e.* $s\sigma \xrightarrow[A]{*} q^{\mathrm{f}}$. Let τ be the restriction of 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 $s\theta \xrightarrow[R_A]{*} s_i$, for some 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 may associate (by a substitution θ) a term $s_x \in \cap_{l \leq j} L(A, q_l)$ where $q_1, \ldots, q_j$ are the states occurring in s_i at the occurrences of x in s. Hence $s\tau\theta \xrightarrow[R]{*} t\sigma$ for some grounding substitution σ which is only defined on the variables of t. Since $Var(s) \cap Var(t) = \emptyset$, 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 $|Q_A|^k$, that is, 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 $O(n^{3k+4})$. □

[b] One can decide these emptiness problems in time $\|A\|^k \in O(n^{3k})$.

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 5 according to the (non)-linearity of the target.

As a consequence we obtain these two theorems:

Theorem 6 *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 7 *Rigid reachability is $\mathcal{P}$-complete if the rules are ground, the source and the target are variable-disjoint, either the source or the target is linear, and if the number of occurrences of non-linear variables in the non-linear term 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 [25]) to rigid reachability where rules, source and target are ground. The upper bound has been proved in Theorem 5. □

We now generalize Theorem 7 to the simultaneous case of rigid reachability.

Theorem 8 *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 7 separately to each constraint of the system. □

Similarly, we can prove:

Theorem 9 *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 7) 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 9 by allowing some common variables between the s_i.

Theorem 10 *Simultaneous rigid reachability is in* EXPTIME *when all the rules of a system of constraints $\big((R_1, s_1, t_1), \ldots, (R_m, s_m, t_m)\big)$ are ground, every t_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 (for $i \neq j$), are variable-disjoint.*

Proof. We reduce this problem to an exponential number of problems of the type of Theorem 9.

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 1). The size of each A_i is in $O(\|t_i\| * \|R_i\|^2)$. We may assume that the state sets of the A_i are pairwise disjoint and that the final states sets of the A_i are singletons, say, $F_{A_i} = \{q_i^{\mathrm{f}}\}$. 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:

(i) $(R_{A_1}, s_{1,j_1}, q_1^{\mathrm{f}}), \ldots, (R_{A_m}, s_{1,j_m}, q_m^{\mathrm{f}})$

(ii) for every variable x which occurs k times in $\{s_1, \ldots, s_n\}$, with $k \geq 2$, $\big(R_{A_1} \uplus \ldots \uplus R_{A_m}, f_{\mathrm{u}}^k(x, \ldots, x), f_{\mathrm{u}}^k(q_1, \ldots, q_k)\big)$, 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 $\big((R_1, s_1, t_1), \ldots, (R_n, s_n, t_n)\big)$ has a solution if and only if, 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 satisfies the requirements of Theorem 9, 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. □

The theorem is true, symmetrically, when we exchange the rôles of sources and targets. We conclude this section by mentioning that the only difference between the conditions for undecidability of rigid reachability in Theorem 1 and the condition for decidability in Theorems 5, 6, and 7 is the linearity of source and (or) target terms.

6.2. Balanced Reachability Constraints

In this section, we consider a second form of restrictions on source and target terms which makes non-monadic SRR decidable for ground rules. The restriction will be placed on the depth of non-linear variable occurrences.

6.2.1. Semi-linear sequences of terms

We call a sequence of terms $(t_1, t_2, \ldots, t_m)$ of terms in $\mathcal{T}_\Sigma \cup \{\bot\}$ *semi-linear* if one of the following conditions holds for each t_i:

(i) t_i is a variable, or
(ii) t_i is a linear term and no variable in t_i occurs in t_j for $i \neq j$.

Note that if t_i is ground then it satisfies the second condition trivially.

Lemma 10 *Let $(s_1, s_2, \ldots, s_k)$ be a semi-linear sequence of Σ-terms. Then the subset $\big\{s_1\theta \otimes s_2\theta \otimes \cdots \otimes s_k\theta : \theta$ is a grounding Σ-substitution$\big\} \subseteq {\mathcal{T}_\Sigma}^m$ is recognizable by a TA of size in $O\big((\|s_1\| + \|\Sigma\|) \cdots (\|s_k\| + \|\Sigma\|)\big)$.*

Proof. Let Σ and $\vec{s} = s_1, s_2, \ldots, s_k$ be given. Let A_i be the TA that recognizes $\{s_i\theta : s_i\theta \in \mathcal{T}_\Sigma\}$ for $1 \leq i \leq k$. The desired TA is $(\bigotimes A_i)\lfloor I$, where I is the set of all selection constraints $i \approx j$ such that s_i and s_j are identical variables. □

We shall also use the following lemma.

Lemma 11 *Let A be a TA, $s \in \mathcal{T}_\Sigma$, and $p_1, \ldots, p_k$ be independent positions in s. Then there is a TA A', with $\|A'\| \in O\big(\|A\|^{2k}\big)$, that recognizes the set $\big\{s_1 \otimes \cdots \otimes s_k : s_1, \ldots, s_k \in \mathcal{T}_\Sigma,\ s[s_1]_{p_1} \ldots [s_k]_{p_k} \in L(A)\big\}$*

Proof. For all states $q \in Q_A$, let A_q be the automaton $(Q_A, \Sigma, R_A, \{q\})$. Let $\{\vec{q_i}\}_{1 \leq i \leq m}$ be the collection of all sequences $\vec{q_i} = q_{i1}, \ldots, q_{ik} \in Q_A$ such that, for some $q_{\mathrm{f}} \in F_A$, $s[q_{i1}]_{p_1} \ldots [q_{ik}]_{p_k} \xrightarrow[R_A]{*} q_{\mathrm{f}}$. For all such sequences $\vec{q_i}$, $1 \leq i \leq m$, construct a TA A_i that recognizes

$$L(A_{q_{i1}}) \otimes \cdots \otimes L(A_{q_{ik}}).$$

Here we can assume that each $L(A_{q_{ij}})$ is nonempty, or else $L(A_i)$ is empty. Assume that all the A_i's have disjoint sets of states and let A' be the union of all the A_i's. It is easy to check that A' recognizes the given set of terms. Note that $m \leq |Q_A|^k$. The size of A' is therefore $\|A'\| \leq \sum_{i=1}^{m} \|A_i\| \leq \sum_{i=1}^{m} \|A\|^k \leq |Q_A|^k \times \|A\|^k$ □

6.2.2. Parallel decomposition of sequences of terms

We generalize the notion of a product of terms given in the section 2 by also admitting *non-ground* terms. The resulting term lives in an extended signature with $\otimes$ as an additional variadic function symbol. The definition is the same as given in equation (1) in section 2, with the additional stipulation that if one of the t_i is a variable then

$$t_1 \otimes \cdots \otimes t_m = \otimes(t_1, \ldots, t_m).$$

In other words, if one of the t_i in a product is a variable, $t_1 \otimes \cdots \otimes t_m$ is left as it is and considered a term in the extended language. Suppose that $\vec{s} = s_1, \ldots, s_m$ is a sequence of terms, and let $(\otimes(\vec{t_i}))_{1 \leq i \leq k}$ be the sequence of all the subterms of the product term $\bigotimes \vec{s}$ which have head symbol $\otimes$ (applied to argument lists $\vec{t_i}$). The *parallel decomposition* of $\vec{s} = s_1, \ldots, s_m$, denoted $pd(\vec{s})$, is the sequence $(\vec{t_i})_{1 \leq i \leq k}$ of the argument lists of the $\otimes$ subterms in $\vec{s}$. The positions at which the $\otimes$ subterms occur will be denoted by $pdp(\vec{s})$. More precisely, $pdp(\vec{s})$ is the sequence $(p_i)_{1 \leq i \leq k}$, where p_i is the position of $\otimes(\vec{t_i})$ in $\bigotimes \vec{s}$.

The following example illustrates these new definitions and lemmas and how they are used.

Example 2 Let $s = f(g(z), g(x))$ and $t = f(y, f(x, y))$ be two terms, and let R be a ground rewrite system over Σ. We will show how to capture all the solutions of the reachability constraint (R, s, t) as a certain regular set of $\Sigma_{\perp}^2$-terms. First, construct the product $s \otimes t$.

$$\begin{aligned} s \otimes t &= f(g(z), g(x)) \otimes f(y, f(x, y)) \\ &= \langle ff \rangle (g(z) \otimes y, g(x) \otimes f(x, y)) \\ &= \langle ff \rangle (\otimes(g(z), y), \langle gf \rangle (x \otimes x, \perp \otimes y)) \\ &= \langle ff \rangle (\otimes(g(z), y), \langle gf \rangle (\otimes(x, x), \otimes(\perp, y))) \end{aligned}$$

The $\otimes$-terms in $s \otimes t$ are $\otimes(g(z), y)$, $\otimes(x, x)$, $\otimes(\perp, y)$. Appending their arguments gives us the semi-linear sequence $pd(s, t) = g(z), y, x, x, \perp, y$. (Note that $pdp(s, t)$ is the sequence $1, 21, 22$.) It follows from Lemma 10 that there is a TA A' such that $L(A') = \{g(z\theta) \otimes y\theta \otimes x\theta \otimes x\theta \otimes \perp \otimes y\theta : \theta \text{ is a grounding } \Sigma\text{-substitution}\}$.

Now, consider a TA A_R that recognizes the relation of $\xrightarrow[R]{*}$, see Lemma 1, *i.e.*, $L(A_R) = \{u \otimes v : u \xrightarrow[R]{*} v,\ u, v \in \mathcal{T}_\Sigma\}$. From A_R we can, by using Lemma 11, construct a TA A'' such that

$$L(A'') = \{s_1 \otimes s_{21} \otimes s_{22} : s_1, s_{21}, s_{22} \in {\mathcal{T}_\Sigma}^2,\ \langle ff \rangle (s_1, \langle gf \rangle (s_{21}, s_{22})) \in L(A_R)\}$$

This is the language which results from $L(A_R)$ by projecting to the subterms at the positions $pdp(s,t)$. Let A recognize $L(A') \cap L(A'')$. We get that

$$
\begin{aligned}
L(A) &= L(A') \cap L(A'') \\
&= \left\{ s_1 \otimes s_{21} \otimes s_{22} : \begin{array}{l} (\exists x\theta, y\theta, z\theta \in \mathcal{T}_\Sigma) \\ s_1 = g(z\theta) \otimes y\theta,\ s_{21} = x\theta \otimes x\theta,\ s_{22} = \bot \otimes y\theta, \\ \langle ff \rangle (s_1, \langle gf \rangle (s_{21}, s_{22})) \in L(A_R) \end{array} \right. \\
&= \big\{ g(z\theta) \otimes \cdots \otimes y\theta : \langle ff \rangle (g(z\theta) \otimes y\theta, \langle gf \rangle (x\theta \otimes x\theta, \bot \otimes y\theta)) \in L(A_R) \big\} \\
&= \big\{ g(z\theta) \otimes \cdots \otimes y\theta : f(g(z\theta), g(x\theta)) \xrightarrow[R]{*} f(y\theta, f(x\theta, y\theta)) \big\} \\
&= \big\{ g(z\theta) \otimes \cdots \otimes y\theta : \theta \text{ solves } (R, s, t) \big\}
\end{aligned}
$$

Hence $L(A) \neq \emptyset$ if and only if (R, s, t) is solvable.

The crucial property that is needed in the example to decide the rigid reachability problem is that the parallel decomposition of the sequence consisting of its source and target terms is semi-linear. This observation leads to the following definition.

6.2.3. Balanced systems of reachability constraints

A system $(R_i, s_i, t_i)_{1 \leq i \leq n}$ of reachability constraints with ground rewrite systems R_i is called *balanced* if the parallel decomposition $pd(s_1, t_1, s_2, t_2, \ldots, s_n, t_n)$ is semi-linear. The proof of Lemma 12 is a generalization of the construction in Example 2.

Lemma 12 *From every balanced system S of reachability constraints we can construct in* EXPTIME *a TA A such $L(A) \neq \emptyset$ iff S is satisfiable.*

Proof. Let $S = \big((R_1, s_1, t_1), \ldots, (R_n, s_n, t_n)\big)$ be a given a balanced system of reachability constraints. Let $U = s_1 \otimes t_1 \otimes \ldots \otimes s_n \otimes t_n$ and $(p_1, \ldots, p_k) = pdp(s_1, t_1, \ldots, s_n, t_n)$.
By definition, the sequence $pd(s_1, t_2, \ldots, s_n, t_n) = (u_1, \ldots, u_{2kn})$ is semi-linear. Therefore, it follows from Lemma 10 that there is a TA A' such that

$$L(A') = \big\{ u_1\theta \otimes \ldots \otimes u_{2kn}\theta : \theta \text{ is a grounding } \Sigma\text{-substitution} \big\}$$

Using Lemma 1, we can associate a TA A_i to each R_i $(i \leq n)$ such that

$$L(A_i) = \big\{ u \otimes v : u \xrightarrow[R_i]{*} v,\ u, v \in \mathcal{T}_\Sigma \big\}$$

We can use Lemma 11 to construct a TA A'' such that

$$L(A'') = \Big\{ v_1 \otimes \ldots \otimes v_k : v_1, \ldots, v_k \in {\mathcal{T}_\Sigma}^{2n}, U[v_1]_{p_1} \ldots [v_k]_{p_k} \in L(\bigotimes_{i=1}^{n} A_i) \Big\}$$

Note that both $L(A')$ and $L(A'')$ are subsets of ${\mathcal{T}_\Sigma}^{2kn}$. Let A be a TA recognizing $L(A') \cap L(A'')$. We observe that $L(A) \neq \emptyset$ if and only if, S is satisfiable. Let t be a term in ${\mathcal{T}_\Sigma}^{2kn}$.

Now, $t \in L(A)$

iff $t = u_1\theta \otimes \ldots \otimes u_{2kn}\theta$ for some grounding Σ-substitution θ (as t is in $L(A')$), and $U[w_1]_{p_1} \ldots [w_k]_{p_k} \in L(\bigotimes_{i=1}^{n} A_i)$, where $w_i = \bigotimes_{j=(i-1)n+1}^{i \cdot n} u_j\theta$ (as t is in $L(A'')$),

iff $(s_1 \otimes t_1 \otimes \ldots \otimes s_n \otimes t_n)[w_1]_{p_1} \ldots [w_k]_{p_k} \in L(\bigotimes_{i=1}^{n} A_i)$,

iff $s_1\theta \otimes t_1\theta \otimes \ldots \otimes s_n\theta \otimes t_n\theta \in L(\bigotimes_{i=1}^{n} A_i)$, because every variable of S occurs in one of the $u_1, \ldots, u_{2kn}$, by definition of pd,

iff $s_1\theta \xrightarrow[R_1]{*} t_1\theta, \ldots, s_n\theta \xrightarrow[R_n]{*} t_n\theta$.

Let us, finally, calculate the size of A, as the complexity of its construction depends linearly on its size. For each $i \leq n$, the size of A_i is polynomial in $\|R_i\|$, thus $\left\|\bigotimes_{i=1}^{n} A_i\right\| \leq M^{cn}$ where $M = \max\{\|R_i\| : i \leq n\}$ and c is a constant independent of the problem size. Therefore, $\|A'\| \leq M^{2cnk}$, cf. Lemma 11. According to Lemma 10, $\|A''\| \leq \|u_1\| \times \ldots \times \|u_{kn}\| \leq \Pi_{i=1}^{n}\|s_i\| \times \Pi_{i=1}^{n}\|t_i\| \leq N^{2n}$, where $N = \max\{\|s_i\|, \|t_i\| : i \leq n\}$. Hence,

$$\|A\| = \|A''\| \times \|A'\| \leq N^{2n} \times M^{2cnk} \leq \|S\|^{2n(ck+1)}.$$

□

Theorem 11 *Simultaneous rigid reachability is* EXPTIME*-complete for balanced systems with ground rules.*

Proof. The EXPTIME hardness follows from the lemma 9, and the membership in EXPTIME is a direct consequence of Lemma 12. □

The theorem can also be used to show the decidability of the following variation of the fragment. Suppose that for each variable x there exists an integer d_x such that x occurs only at positions of length d_x, as is the case for $s_1 = f(x, g(y))$, $t_1 = f(f(y, y), x)$, $s_2 = g(x)$, and $t_2 = g(f(a, y))$. To reduce the problem to balanced case, one simply non-deterministically guesses terms with new variables to be substituted for x (in the example we might guess the terms a, $g(x_1)$, or $f(x_2, x_3)$, among others) such that the outcome is a balanced system where all variables occur at the same depth. Every solution of the original system arises as the composition of the guessed substitution with a solution of the balanced system.

7. Conclusion

We have shown that absence of symmetry makes solving of rigid reachability constraints 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 for fragments with ground rules, including the monadic case and certain non-monadic cases with restrictions on non-linear occurrences of variables.

Acknowledgments

The authors wish to thank the referees for their useful remarks and suggestions, which led to significant improvements of the results. The PSPACE upper bound of monadic SRR with ground rules has been obtained by joint work with Veronique Cortier, and was published in [6].

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
2. A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, January 1981.
3. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. http://www.grappa.univ-lille3.fr/tata, 1999.
4. J.-L. Coquidé and R. Gilleron. Proofs and reachability problem for ground rewrite systems. In *Proc. of the 6th International Meeting of Young Computer Scientists*, volume 464 of *LNCS*, pages 120–129, 1990.
5. J.L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi. Bottom-up tree pushdown automata: classification and connection with rewrite systems. *Theoretical Computer Science*, 127:69–98, 1994.
6. V. Cortier, H. Ganzinger, F. Jacquemard, and M. Veanes. Decidable fragments of simultaneous rigid reachability. In *Proc. 26th Int. Coll. on Automata, Languages and Programming, ICALP99, Prague*, volume 1644 of *Lecture Notes in Computer Science*. Springer-Verlag, july 1999.
7. M. Dauchet. Rewriting and tree automata. In H. Comon and J.P. Jouannaud, editors, *Term Rewriting (French Spring School of Theoretical Computer Science)*, volume 909 of *Lecture Notes in Computer Science*, pages 95–113. Springer Verlag, Font Romeux, France, 1993.
8. M. Dauchet, T. Heuillard, P. Lescanne, and S. Tison. Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems. *Information and Computation*, 88:187–201, 1990.
9. A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. Decidability and complexity of simultaneous rigid E-unification with one variable and related results. *Theoretical Computer Science*, 1998. To appear.
10. A. Degtyarev, Yu. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid E-unification with one variable. In T. Nipkow, editor, *Rewriting Techniques and Applications*, volume 1379 of *Lecture Notes in Computer Science*, pages 181–195. Springer Verlag, 1998.
11. A. Degtyarev, Yu. Matiyasevich, and A. Voronkov. Simultaneous rigid E-unification and related algorithmic problems. In *Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 494–502, New Brunswick, NJ, July 1996. IEEE Computer Society Press.
12. A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. UPMAIL Technical Report 105, Uppsala University, Computing Science Department, May 1995.
13. A. Degtyarev and A. Voronkov. Decidability problems for the prenex fragment of intuitionistic logic. In *Eleventh Annual IEEE Symposium on Logic in Computer*

Science (LICS'96), pages 503–512, New Brunswick, NJ, July 1996. IEEE Computer Society Press.

14. 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.
15. J. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.
16. W.M. Farmer. Simple second-order languages for which unification is undecidable. *Theoretical Computer Science*, 87:25–41, 1991.
17. T. Frühwirth, E. Shapiro, M.Y. Vardi, and E. Yardemi. Logic programs as types for logic programs. In *6th IEEE Symp. Logic In Computer Science*, pages 300–309, 1991.
18. J.H. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Rigid E-unification is NP-complete. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, July 1988.
19. J.H. Gallier, S. Raatz, and W. Snyder. Theorem proving using rigid E-unification: Equational matings. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, 1987.
20. H. Ganzinger, F. Jacquemard, and M. Veanes. Rigid reachability. In *Proc. Asian Computing Science Conference (ASIAN'98)*, volume 1538 of *Lecture Notes in Computer Science*, Berlin, 1998. Springer-Verlag.
21. W.D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.
22. J. Goubault. Rigid $\vec{E}$-unifiability is DEXPTIME-complete. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*. IEEE Computer Society Press, 1994.
23. Y. Gurevich and M. Veanes. Partisan corroboration, and shifted pairing. Research Report MPI-I-98-2-014, Max-Planck-Institut für Informatik, September 1998.
24. Y. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. In P. Degano, R. Corrieri, and A. Marchetti-Spaccamella, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 154–165. Springer Verlag, 1997.
25. D.C. Kozen. Complexity of finitely presented algebras. In *Proc. 9th STOC*, pages 164–177, 1977.
26. D. Kozen. Lower bounds for natural proof systems. In *Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 254–266, 1977.
27. J. Levy. Decidable and undecidable second-order unification problems. In T. Nipkow, editor, *Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March/April 1998, Proceedings*, volume 1379 of *Lecture Notes in Computer Science*, pages 47–60. Springer Verlag, 1998.
28. J. Levy and M. Veanes. On the undecidability of second-order unification. Invited paper for a special issue of *Information and Computation* on term rewriting, 1998.
29. 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.
30. W. Plandowski. Satisfiability of word equations with constants is in PSPACE. In P. Beame, editor, *Proceedings of the Fortieth Annual IEEE Symposium on Founda-*

tions of Computer Science (FOCS'99), New York City, NY, 1999. IEEE Computer Society Press.

31. W.J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.
32. K.U. Schulz. Makanin's algorithm: Two improvements and a generalization. In K.U. Schulz, editor, *Proceedings of the First International Workshop on Word Equations and Related Topics, Tübingen*, number 572 in Lecture Notes in Computer Science, 1990.
33. H. Seidl. Haskell overloading is dexptime-complete. *Inf. Process. Letters*, 52:57–60, 1994.
34. J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.
35. M. Veanes. 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)*, pages 264–275. IEEE Computer Society Press, 1998.
36. M. Veanes. On computational complexity of basic decision problems of finite tree automata. Technical Report 133, Computing Science Department, Uppsala University, jan 1997.