Decidable fragments of simultaneous rigid reachability*

Veronique Cortier[1], Harald Ganzinger[2], Florent Jacquemard[3], and Margus Veanes[2]

[1] École Normale Supérieure de Cachan, dpt Mathématiques
61 Avenue du Président Wilson, 94235 Cachan Cedex, France
`Veronique.Cortier@dptmaths.ens-cachan.fr`
[2] Max-Planck-Institut für Informatik
Im Stadtwald, 66123 Saarbrücken, Germany
`{hg,veanes}@mpi-sb.mpg.de`
[3] LORIA and INRIA, 615 rue du Jardin Botanique,
B.P. 101, 54602 Villers-les-Nancy Cedex, France
`Florent.Jacquemard@loria.fr`

Abstract. In this paper we prove decidability results of restricted fragments of *simultaneous rigid reachability* or *SRR*, that is the nonsymmetrical form of *simultaneous rigid E-unification* or *SREU*. The absence of symmetry enforces us to use different methods, than the ones that have been successful in the context of SREU (for example *word equations*). The methods that we use instead, involve finite (tree) automata techniques, and the decidability proofs provide precise computational complexity bounds. The main results are 1) *monadic SRR with ground rules* is PSPACE-complete, and 2) *balanced SRR with ground rules* is EXPTIME-complete. These upper bounds have been open already for corresponding fragments of SREU, for which only the hardness results have been known. The first result indicates the difference in computational power between fragments of SREU with *ground* rules and *nonground* rules, respectively, due to a straightforward encoding of word equations in monadic SREU (with nonground rules). The second result establishes the decidability and precise complexity of the largest known subfragment of nonmonadic SREU.

1 Introduction

Rigid reachability (RR) is the problem, given a rewrite system R and two terms s and t, whether there exists a substitution θ such that $s\theta$, $t\theta$, and $R\theta$ are ground, and $s\theta$ rewrites in some number of steps via $R\theta$ into $t\theta$. 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 (SRR) is the problem in which a substitution is sought which simultaneously solves each member of

* Full version of this paper is available as: Research Report MPI-I-1999-2-004, Max-Planck-Institut für Informatik.

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 $s \to t$ also its converse $t \to s$. Such systems arise for example by orienting a set of equations in both directions. The latter problem was introduced by Gallier, Raatz & Snyder [1987] as "simultaneous rigid E-unification" (SREU) in the context of extending tableaux or matrix methods in automated theorem proving to logic with equality. Rigid reachability was initially studied in the context of second-order unification [Farmer 1991, Levy 1998].

Even though the non-simultaneous case of SREU (rigid E-unification) was proved NP-complete by Gallier, Narendran, Plaisted & Snyder [1988], SREU in general was shown by Degtyarev & Voronkov [1995] to be undecidable. Further implications of the latter result are discussed in [Degtyarev, Gurevich & Voronkov 1996]. In a series of papers, SREU has been studied extensively and several sharp boundaries have been laid between its decidable and undecidable fragments. Most recent developments are discussed by Voronkov [1998] and Veanes [1998]. Rigid reachability was shown undecidable by Ganzinger, Jacquemard & Veanes [1998].

The, arguably, most difficult remaining open problem regarding SREU is the decidability of "monadic" SREU, or SREU restricted to signatures where all nonconstant function symbols are unary. The importance of this fragment stems from its close relation to word equations [Degtyarev, Matiyasevich & Voronkov 1996], and to fragments of intuitionistic logic [Degtyarev & Voronkov 1996]. What is known about monadic SREU in general, is that it reduces to a non-trivial extension of word equations [Gurevich & Voronkov 1997]. In the case of ground rules, the decidability of monadic SREU was established in [Gurevich & Voronkov 1997] by reducing it to "word equations with regular constraints". The decidability of the latter problem is an extension of Makanin's [1977] result by Schulz [1990]. Conversely, word equations reduce in polynomial time to monadic SREU [Degtyarev, Matiyasevich & Voronkov 1996]. The first main result of this paper (in Section 3), is that monadic SRR with ground rules is in PSPACE, improving the EXPTIME result in Ganzinger et al. [1998]. Hence, it is unlikely that there is a simple reduction, if any reduction at all, from monadic SREU to monadic SREU with ground rules, or else one would get a considerable simplification of Makanin's [1977] proof. The PSPACE-hardness of monadic SREU with ground rules was shown by Goubault [1994].

To obtain the PSPACE result we use an extension of the intersection non-emptiness 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 [Degtyarev, Gurevich, Narendran, Veanes & Voronkov 1998*b*].

Our second main result concerns (nonmonadic) SRR with ground rules. In section 4, we show that SRR with ground rules is EXPTIME-complete for "balanced" systems of reachability constraints. Under balanced systems fall for example systems where all occurrences of each variable are at the same depth. It is possible to obtain undecidability of (nonsimultaneous) rigid reachability with ground rules where all but one occurrence of all variables occur at the same depth [Ganzinger et al. 1998]. Moreover, this result generalizes the decidability result by Degtyarev, Gurevich, Narendran, Veanes & Voronkov [1998*a*] of the largest known decidable fragment of SREU with ground rules and implies EXPTIME-completess of the complexity of this fragment (which is left open in [Degtyarev et al. 1998*a*]). The key characteristic of solving balanced systems involves finite tree automata techniques over product languages, where it is not necessary to search for solutions encoding a product of a term with its proper subterm. This property is also important in decision procedures for "automata with constraints between brothers" [see, e.g. Comon, Dauchet, Gilleron, Lugiez, Tison & Tommasi 1998].

2 Preliminaries

Rigid Reachability. A *reachability constraint*, or simply a constraint, in a signature Σ, is a triple (R, s, t) where R is a set of rules in Σ, and s and t are Σ-terms. 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 Σ, *solves* (R, s, t) if θ is grounding for R, s and t, and $s\theta \xrightarrow[R\theta]{*} t\theta$. The problem of solving constraints is called *rigid reachability.* 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 signatures.

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 [e.g. Coquidé, Dauchet, Gilleron & Vágvölgyi 1994, Dauchet 1993], a *tree automaton* (or *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.* 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 with a monadic signature.

Finite 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* $a_1a_2\cdots a_n$ if and only if, for some final state q and the initial state q_0 of A, $a_n(\cdots a_2(a_1(q_0))\cdots) \xrightarrow[A]{*} q$, i.e.,

$$q_0 \xrightarrow[A]{a_1} q_1 \xrightarrow[A]{a_2} \cdots \xrightarrow[A]{a_n} q.$$

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:

$$t_1 \otimes \cdots \otimes t_m = \langle f_1 f_2 \cdots f_m\rangle(t_{11} \otimes \cdots \otimes t_{1k}, \ldots, t_{m1} \otimes \cdots \otimes t_{mk}) \qquad (1)$$

For example:

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

We write $\mathcal{T}_\Sigma^m$ for the set of all t in $\mathcal{T}_{\Sigma_\perp^m}$ such that $t = t_1 \otimes \cdots \otimes t_m$ for some $t_1, \ldots, t_m \in \mathcal{T}_\Sigma \cup \perp$. 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 $\boldsymbol{t} = t_1, \ldots, t_m$ of terms in $\mathcal{T}_\Sigma \cup \perp$, we write $\bigotimes \boldsymbol{t}$ for the product term $t_1 \otimes \cdots \otimes t_m$

Given two automata A_1 and A_2 over $\Sigma_\perp^m$ and $\Sigma_\perp^n$, respectively, the *product* of A_1 and A_2 is an automaton $A_1 \otimes A_2$ over $\Sigma_\perp^{m+n}$ 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. Comon et al. 1998]. In general, $\bigotimes_{i=1}^n A_i$ is defined accordingly.

We will use the following construction of Dauchet, Heuillard, Lescanne & Tison [1990] in our proofs.

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

3 Monadic SRR

We prove 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.

3.1 Constrained product nonemptiness of NFAs

Given a signature Σ and a positive integer m, we want to select only a certain subset from Σ^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 $\Sigma^m \lfloor I$ for the following subset of Σ^m:

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

For an automaton A, let $A \lfloor I$ denote the reduction of A to the alphabet $\Sigma^m \lfloor I$. 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 $\Sigma^m \lfloor I$.

We consider the following decision problem, that is closely related to the nonemptiness 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 nonemptiness 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. Our key lemma is the following one. Its proof is a straightforward extension of the inclusion part of Kozen's [1977] PSPACE-completess result of the *intersection nonemptiness problem of DFAs*: given a sequence $(A_i)_{1 \leq i \leq n}$ of DFAs, is $\bigcap_{i=1}^n L(A_i)$ nonempty?

Lemma 2. *Constrained product nonemptiness of NFAs (or monadic TAs) is in* PSPACE*.*

The proof of Lemma 2 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 & Stockmeyer [1981], to EXPTIME upper bound for the constrained product nonemptiness problem of TAs.

Although we will not use this fact, it is worth noting that the constrained product nonemptiness 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.

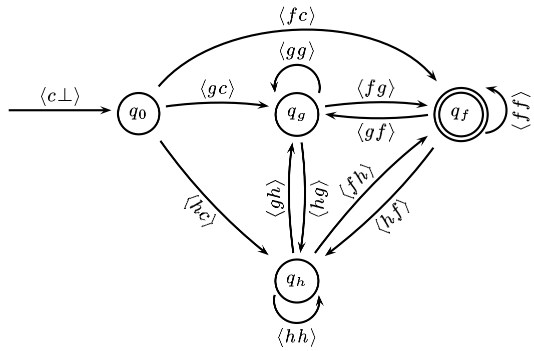

Figure 1: A DFA (or monadic DTA) A that recognizes $\{f(s) \otimes s \ : \ s \in \mathcal{T}_\Sigma\}$, where Σ consists of the unary function symbols f, g, and h, and the constant c. For example A recognizes the string $\langle c\bot\rangle\langle gc\rangle\langle gg\rangle\langle hg\rangle\langle fh\rangle$, 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 $f(h(g(g(c)))) \otimes h(g(g(c)))$.

3.2 Reduction of monadic SRR with ground rules to constrained product nonemptiness of NFAs

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 solvability of a reachability constraint with empty rule set is simply unifiability of the source and the target. The following simple lemma is useful.

Lemma 3. *Let S be a system of reachability constraints. There is a flat system that can be obtained in polynomial time from S, that is solvable if and only if S is solvable.*

By using Lemma 2 and Lemma 3 we can now show the following theorem, that is the main result of this section.

Theorem 1. *Monadic SRR with ground rules is* PSPACE*-complete.*

The crucial step in the proof of Theorem 1 is the construction of an automaton that recognizes the language $\{f(s)\otimes s \ : \ s \in \mathcal{T}_\Sigma\}$. (See Figure 1.) 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 nonmonadic signatures. The next example illustrates how the reduction in the proof of Theorem 1 works.

Example 1. Consider a 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. (This system is solvable if and only if the constraint $(R, f(g(x)), x)$ is solvable.)

The construction in the proof of Theorem 1 gives us the monadic TAs 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.

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

So, solvability of S is equivalent to nonemptiness of $L(A_1 \otimes A_2 \otimes A_3)\lfloor I$.

3.3 Some decidable extensions of the monadic case

Some restrictions imposed by only allowing monadic function symbols can be relaxed, without losing decidability of SRR for the resulting classes of constraints. One decidable fragment of SRR is obtained by requiring only the rules to be ground and monadic. It can be shown that SRR for this class is still in PSPACE. Furthermore, an easy argument using the intersection nonemptiness problem of DFAs shows 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 [Degtyarev et al. 1998*b*].

4 A decidable nonmonadic fragment

In this section, we consider general signatures and give a criteria on the source and target terms of a system of reachability constraints for the decidability of SRR when the rules are ground. Moreover, we prove that SRR is EXPTIME-complete in this case. Our decision algorithm involves essentially tree automata techniques. Let Σ be a signature fixed for the rest of the section.

4.1 Semi-linear sequences of terms

We say that a sequence of terms $(t_1, t_2, \ldots, t_m)$ of (possibly non ground) Σ-terms or $\perp$ is *semi-linear* if one of the following conditions holds for each t_i:

1. t_i is a variable, or
2. 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 4. *Let $(s_1, s_2, \ldots, s_k)$ be a semi-linear sequence of Σ-terms. Then the subset $\{s_1\theta \otimes s_2\theta \otimes \cdots \otimes s_k\theta : \theta$ is a grounding Σ-substitution$\} \subseteq \mathcal{T}_\Sigma^m$ is recognized by a TA the size of which is in $O((\|s_1\| + \|\Sigma\|) \times \ldots \times (\|s_k\| + \|\Sigma\|))$.*

Proof. Let Σ and $\boldsymbol{s} = s_1, s_2, \ldots, s_k$ 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 5. *Let $A = (\Sigma, Q, R, F)$ be a TA, $s \in \mathcal{T}_\Sigma$, and $p_1, \ldots, p_k$ parallel positions in s. Then there is a TA A', with $\|A'\| \in O(\|A\|^{2k})$, that recognizes the set $\{s_1 \otimes \cdots \otimes s_k : s_1, \ldots, s_k \in \mathcal{T}_\Sigma,\ s[p_1 \leftarrow s_1, \ldots, p_k \leftarrow s_k] \in L(A)\}$*

4.2 Parallel decomposition of sequences of terms

For technical reasons, we generalize the notion of a product of m terms by allowing *nonground* terms. The resulting term is in an extended signature with $\otimes$ as an additional variadic function symbol. The definition is the same as for ground terms (see (1)), with the additional condition that if one of the t_i's is a variable then

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

Consider a sequence $\boldsymbol{s} = s_1, \ldots, s_m$ of terms and let $(\otimes(\boldsymbol{t}_i))_{1 \leq i \leq k}$ be the sequence of all the subterms of the product term $\bigotimes \boldsymbol{s}$ which have head symbol $\otimes$. The *parallel decomposition* of $\boldsymbol{s} = s_1, \ldots, s_m$ or $pd(\boldsymbol{s})$ is the sequence $(\boldsymbol{t}_i)_{1 \leq i \leq k}$, i.e., we forget the symbol $\otimes$. We need the following technical notion in the proof of Lemma 6: $pdp(\boldsymbol{s})$ is the sequence $(p_i)_{1 \leq i \leq k}$, where p_i is the position of $\otimes(\boldsymbol{t}_i)$ in $\bigotimes \boldsymbol{s}$.

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 Σ-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 preorder traversal of $s \otimes t$ yields the sequence $\otimes(g(z), y)$, $\otimes(x, x)$, $\otimes(\perp, y)$.

Finally, $pd(s,t)$ is the semi-linear sequence $g(z), y, x, x, \perp, y$. (Note that the sequence $pdp(s,t)$ is $1, 21, 22$.) It follows from Lemma 4 that there is a TA A_1 such that $L(A_1) = \{g(z\theta) \otimes y\theta \otimes x\theta \otimes x\theta \otimes \perp \otimes y\theta : \theta \text{ is a grounding } \Sigma\text{-substitution}\}$.

Now, consider a TA A_R that recognizes the product 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 5, construct a TA A_2 such that

$$L(A_2) = \{s_1 \otimes s_{21} \otimes s_{22} : s_1, s_{21}, s_{22} \in \mathcal{T}_\Sigma^2,\ \langle ff\rangle(s_1, \langle gf\rangle(s_{21}, s_{22})) \in L(A_R)\}$$

Let A recognize $L(A_1) \cap L(A_2)$. We get that

$$\begin{aligned}
L(A) &= L(A_1) \cap L(A_2) \\
&= \left\{ \begin{array}{ll} s_1 \otimes s_{21} \otimes s_{22} : & (\exists x\theta, y\theta \in \mathcal{T}_\Sigma) \\ & s_1 = x\theta \otimes y\theta,\ s_{21} = x\theta \otimes x\theta,\ s_{22} = \perp \otimes y\theta, \\ & \langle ff\rangle(s_1, gf(s_{21}, s_{22})) \in L(A_R) \end{array} \right. \\
&= \{\theta : \langle ff\rangle(x\theta \otimes y\theta, gf(x\theta \otimes x\theta, \perp \otimes y\theta)) \in L(A_R)\} \\
&= \{\theta : \theta \text{ solves } (R, s, t)\}
\end{aligned}$$

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

The crucial property that is needed in the example to prove the decidability of 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.

4.3 Balanced systems with ground rules

A system $((R_1, s_1, t_1), \ldots, (R_n, s_n, t_n))$ of reachability constraints 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 6 is a generalization of the construction in Example 2.

Lemma 6. *From every balanced system S of reachability constraints with ground rules, we can construct in* EXPTIME *a TA A such $L(A) \neq \emptyset$ iff S is satisfiable.*

Theorem 2. *Simultaneous rigid reachability is* EXPTIME*-complete for balanced systems with ground rules.*

Proof. The EXPTIME-hardness follows from [Ganzinger et al. 1998], where we have proved that one can reduce the emptiness decision for intersection of n tree automata to the satisfiability of a rigid reachability constraint $(R, f(x, \ldots, x), f(q_1, \ldots, q_n))$, where R is ground and $q_1, \ldots, q_n$ are constants. ⊠

The balanced case embeds the case where for each variable x with multiple occurrences in source and target terms, there exists an integer d_x such that x occurs only at positions of length d_x, e.g. with $s_1 = f(x, g(y))$, $t_1 = f(f(y, y), x)$, $s_2 = g(x)$ and $t_2 = g(f(a, z))$. Note that this is a strict subcase of the balanced case, for instance, the system described in example 2, though balanced, does not fulfill this condition.

References

Chandra, A., Kozen, D. & Stockmeyer, L. (1981), 'Alternation', *Journal of the Association for Computing Machinery* **28**(1), 114–133.

Comon, H., Dauchet, M., Gilleron, R., Lugiez, D., Tison, S. & Tommasi, M. (1998), *Tree Automata Techniques and Applications*, unpublished.

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.

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. (1990), 'Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems', *Information and Computation* **88**, 187–201.

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

Degtyarev, A. & Voronkov, A. (1996), Decidability problems for the prenex fragment of intuitionistic logic, *in* 'Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)', IEEE Computer Society Press, New Brunswick, NJ, pp. 503–512.

Degtyarev, A., Gurevich, Y. & Voronkov, A. (1996), Herbrand's theorem and equational reasoning: Problems and solutions, *in* 'Bulletin of the European Association for Theoretical Computer Science', Vol. 60. The "Logic in Computer Science" column.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M. & Voronkov, A. (1998*a*), 'Decidability and complexity of simultaneous rigid E-unification with one variable and related results', *Theoretical Computer Science*. To appear.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M. & Voronkov, A. (1998*b*), 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.

Degtyarev, A., Matiyasevich, Y. & Voronkov, A. (1996), Simultaneous rigid E-unification and related algorithmic problems, *in* 'Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)', IEEE Computer Society Press, New Brunswick, NJ, pp. 494–502.

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.

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.

Ganzinger, H., Jacquemard, F. & Veanes, M. (1998), Rigid reachability, *in* J. Hsiang & A. Ohori, eds, 'Advances in Computing Science – ASIAN'98, 4th Asian Computing Science Conference, Manila, The Philippines, December 1998, Proceedings', Vol. 1538 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 4–21.

Goubault, J. (1994), Rigid ***E***-unifiability is DEXPTIME-complete, *in* 'Proc. IEEE Conference on Logic in Computer Science (LICS)', IEEE Computer Society Press.

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), Lower bounds for natural proof systems, *in* 'Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)', pp. 254–266.

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.

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

Schulz, K. (1990), Makanin's algorithm: Two improvements and a generalization, *in* K. Schulz, ed., 'Proceedings of the First International Workshop on Word Equations and Related Topics, Tübingen', number 572 *in* 'Lecture Notes in Computer Science'.

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. (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.

Voronkov, A. (1998), 'Simultaneous rigid E-unification and other decision problems related to Herbrand's theorem', *Theoretical Computer Science*. Article after invited talk at *LFCS'97*.