

Decidable fragments of
simultaneous rigid reachability

Veronique Cortier, Harald Ganzinger,
Florent Jacquemard, and Margus Veanes

MPI-I-1999-2-004

March 1999

FORSCHUNGSBERICHT RESEARCH REPORT

MAX - PLANCK - INSTITUT
FÜR
INFORMATIK

Im Stadtwald 66123 Saarbrücken Germany

Authors' Addresses

Veronique Cortier
École Normale Supérieure de Cachan, dpt Mathématiques
61 Avenue du Président Wilson, 94235 Cachan cedex, France
Veronique.Cortier@dptmaths.ens-cachan.fr

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

Florent Jacquemard
LORIA and INRIA, 615 rue du Jardin Botanique,
B.P. 101, 54602 Villers-les-Nancy Cedex, France
Florent.Jacquemard@loria.fr

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

Publication Notes

A short version of this paper appears in the proceedings of the *26-th International Colloquium on Automata, Languages, and Programming (ICALP'99)*, In *LNCS*.

Acknowledgements

We thank Jürgen Stuber for suggestions to improvements and for pointing out several inconsistencies in an earlier version on this report.

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 forces us to use different methods, from the ones that have been successful in the context of SREU in the past (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. 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.

Keywords

Rigid Unification, Term Rewriting, Reachability.

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 each rule only 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 reachability arises when the R_i are symmetric, containing for each rule $s \rightarrow t$ also its converse $t \rightarrow s$. 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 introduced in the context of second-order unification [Farmer 1991, Levy 1998].

Although 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 nontrivial 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 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 [Degtyarev, Gurevich, Narendran, Veanes & Voronkov 1998b].

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, our decidability result generalizes the decidability result by Degtyarev, Gurevich, Narendran, Veanes & Voronkov [1998a] of the largest known decidable fragment of SREU with ground rules and implies EXPTIME-completeness of the complexity of this fragment (which is left open in [Degtyarev et al. 1998a]). We use finite tree automata techniques over product languages, that have been used in decision procedures for “automata with constraints between brothers” [cf. Comon, Dauchet, Gilleron, Lugiez, Tison & Tommasi 1998].

2 Preliminaries

A *signature* Σ is a collection of *function symbols* with fixed arities ≥ 0 and, unless otherwise stated, Σ is assumed to contain at least one *constant*, that is, one function symbol with arity 0. We use a, b, c, d, a_1, \dots for constants and f, g, f_1, \dots for function symbols in general. A signature is called *monadic* if all function symbols in it have arity ≤ 1 . A *ground* term is one that contains no variables. The set of all ground terms over a signature Σ is denoted by \mathcal{T}_Σ .

We use s, t, l, r, s_1, \dots for terms. The size $\|t\|$ of a term t is defined recursively by: $\|t\| = 1$ if t is either a variable or a constant and

$$\|f(t_1, \dots, t_n)\| = \|t_1\| + \dots + \|t_n\| + 1.$$

Positions in terms are sequences of integers. We use p, p_1 etc for positions, ϵ for the empty sequence (root position) and pp' for the concatenation product of two positions p and p' . We will also use the prefix ordering \prec on positions. Some positions p_1, \dots, p_n are called *parallel* if they are pairwise uncomparable with respect to \prec .

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

Rigid Reachability. 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 & Vágvolgyi 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 $f(q_1, \dots, q_n) \rightarrow q$, where $f \in \Sigma$ has arity $n \geq 0$ and $q, q_1, \dots, q_n \in Q$, 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 $a(q) \rightarrow p$, 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 $a_1(a_2(\dots a_n(q)))$ is written, using the *reversed Polish notation*, as the string $qa_n \dots a_1$.

Then A *accepts a string* $a_1 a_2 \dots a_n$ if and only if, for some final state q and the initial state q_0 of A , $a_n(\dots a_2(a_1(q_0)) \dots) \xrightarrow{*}_A q$, i.e.,

$$q_0 \xrightarrow{a_1}_A q_1 \xrightarrow{a_2}_A \dots \xrightarrow{a_n}_A q.$$

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 Σ_\perp for $\Sigma \cup \{\perp\}$ and Σ_\perp^m denotes the signature consisting of, for all $f_1, f_2, \dots, f_m \in \Sigma_\perp$, a unique function symbol $\langle f_1 f_2 \dots f_m \rangle$ with arity equal to the maximum of the arities of the f_i 's.

Let $t_i \in \mathcal{T}_\Sigma \cup \perp$, $t_i = f_i(t_{i1}, \dots, t_{ik_i})$, 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* $t_1 \otimes \dots \otimes t_m$ of t_1, \dots, t_m is defined by recursion on the subterms:

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

For example:

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

We write \mathcal{T}_Σ^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 $t_1, \dots, t_m \in \mathcal{T}_\Sigma \cup \perp$. 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}_Σ^{m+n} . Given a sequence $\vec{t} = t_1, \dots, t_m$ of terms in $\mathcal{T}_\Sigma \cup \perp$, 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 Σ_\perp^m and Σ_\perp^n , respectively, the *product* of A_1 and A_2 is an automaton $A_1 \otimes A_2$ over Σ_\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 $\Sigma^m \downarrow I$ for the following subset of Σ^m :

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

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

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 $(\bigotimes_{i=1}^n L(A_i)) \downarrow I$ is nonempty. Our key lemma is the following one. Its proof is a straightforward extension of the inclusion part of Kozen's [1977] PSPACE-completeness 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.*

Proof. Let $(A_i)_{1 \leq i \leq n}$, m_i , m , Σ , and I be given as above. Assume also that $m_i = 2$ for $1 \leq i \leq n$, i.e., $m = 2n$ and each automaton has alphabet Σ_{\perp}^2 . (Proof of the general case is analogous.)

Furthermore, we can assume, without loss of generality, that none of the automata accepts the empty string and that, for each string v that is accepted by A_i also $\langle \perp \perp \rangle v$ is accepted by A_i , e.g., we can assume that each automaton has a transition with label $\langle \perp \perp \rangle$ from the initial state to the initial state. Consider the following nondeterministic decision procedure.

- I (**Initialize**) Calculate the number of states in $\bigotimes_{i=1}^n A_i$ and save it in **IterationLimit**.
(This calculation is easy, because each state of $\bigotimes_{i=1}^n A_i$ corresponds to a sequence of states $(q_i)_{1 \leq i \leq n}$, where q_i is a state of A_i .)
Save in **State_i** the initial state of A_i for $1 \leq i \leq n$.
- II (**Guess the next letter from $\Sigma_{\perp}^m \downarrow I$**) Select $(a_1, \dots, a_m) \in \Sigma_{\perp}^m \downarrow I$ and store a_i in **Letter_i**.
- III (**Guess the next transition of $(\bigotimes_{i=1}^n A_i) \downarrow I$**) For $1 \leq i \leq n$, guess nondeterministically a state q_i from A_i .
Check that, for $1 \leq i \leq n$, there is a $\langle \mathbf{Letter}_{2i-1} \mathbf{Letter}_{2i} \rangle$ -transition in A_i from **State_i** to q_i , and if so, save q_i in **State_i**. If there is no such transition then terminate and **reject**.
- IV (**Check acceptance of $(\bigotimes_{i=1}^n A_i) \downarrow I$**) If, for $1 \leq i \leq n$, **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 corresponds to walking through the graph of $(\bigotimes_{i=1}^n A_i) \downarrow I$, 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 $L(\bigotimes_{i=1}^n A_i) \downarrow I$ from the initial state to a final state. It is evident that the procedure always terminates, and that it accepts if and only if $L(\bigotimes_{i=1}^n A_i) \downarrow I$ is nonempty.

It is obvious, through straightforward binary encodings, that no more than polynomial space is required, in order to meet the space requirements of the procedure. Hence, the procedure runs in nondeterministic polynomial space and thus in PSPACE, by using the result of Savitch [1970].

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

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 $L(\bigotimes_{i=1}^n A_i) \downarrow \{i \approx i + 1 : 1 \leq i < n\}$ is nonempty.

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.*

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

1. Replace each constraint (R, s, t) , where s is not a variable, or when $s = t$, by the constraints (R, x, t) and (\emptyset, x, s) , where x is a new variable.
2. 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.
3. 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.
4. 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 . \square

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.

Proof. The PSPACE-hardness has been proved already in the case when the rule sets are symmetric [Goubault 1994] and there is only one variable [Gurevich & Voronkov 1997]. We prove inclusion in PSPACE by giving a polynomial time reduction to the constrained product nonemptiness problem of monadic TAs.

Let S be a system of reachability constraints with ground rules. Let Σ be the signature of S . We may assume, by using Lemma 3, that S is flat. Enumerate all the constraints in S as $\rho_1, \dots, \rho_m, \rho_{m+1}, \dots, \rho_n$, where all the constraints of the form $(\emptyset, x, f(y))$ are enumerated as $\rho_{m+1}, \dots, \rho_n$. 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 $1 \leq i \leq m$, construct a TA A_i such that,

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

For $m < i \leq n$, construct a TA A_i such that,

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

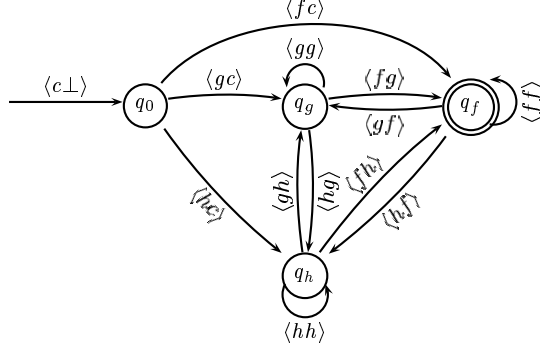


Figure 1: A DFA (or monadic DTA) A that recognizes $\{f(s) \otimes s : s \in \mathcal{T}_\Sigma\}$, where Σ consists of the unary function symbols f , g , and h , and the constant c . For example A recognizes the string $\langle c\perp \rangle \langle gc \rangle \langle gg \rangle \langle hg \rangle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

(Such an automaton is illustrated in Figure 1.) It follows from Lemma 1 that all these TAs can be constructed in polynomial time.

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

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

It remains to be proved that $L(\bigotimes_{i=1}^n A_i) \upharpoonright I$ is nonempty if and only if S is solvable. (This proof is straightforward, and is illustrated in Example 1.) The theorem follows then from Lemma 2. \square

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 $L(\bigotimes_{i=1}^3 A_i) \downarrow I$ is as follows.

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

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

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. 1998b].

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

Proof. Let Σ and $\vec{s} = s_1, s_2, \dots, s_k$ 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 $(\bigotimes A_i)|I$, where I is the set of all selection constraints $i \approx j$ such that s_i and s_j are identical variables. \square

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, \dots, p_k parallel positions in s . Then there is a TA A' , with $\|A'\| \in O(\|A\|^{2k})$, that recognizes the set $\{s_1 \otimes \dots \otimes s_k : s_1, \dots, s_k \in \mathcal{T}_\Sigma, s[p_1 \leftarrow s_1, \dots, p_k \leftarrow s_k] \in L(A)\}$*

Proof. For all states $q \in Q$, let A_q be the automaton $(\Sigma, Q, R, \{q\})$. Let $\{\vec{q}_i\}_{1 \leq i \leq m}$ be the collection of all sequences $\vec{q}_i = q_{i1}, \dots, q_{ik} \in Q$ such that, for some $q_f \in F$, $s[p_1 \leftarrow q_{i1}, \dots, p_k \leftarrow q_{ik}] \xrightarrow{*}_R q_f$. For all such sequences \vec{q}_i , $1 \leq i \leq m$, construct a TA A_i that recognizes

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

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 $m \leq |Q|^k$. The size of A' is therefore $\|A'\| \leq \sum_{i=1}^m \|A_i\| \leq \sum_{i=1}^m \|A\|^k \leq |Q|^k \times \|A\|^k$ \square

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

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

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

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 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 Σ_{\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 $pdp(s, t)$ is the sequence 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

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

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 $((R_1, s_1, t_1), \dots, (R_n, s_n, t_n))$ of reachability constraints is called *balanced* if the parallel decomposition $pd(s_1, t_1, s_2, t_2, \dots, s_n, t_n)$ 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.*

Proof. Let $S = ((R_1, s_1, t_1), \dots, (R_n, s_n, t_n))$ be a given a balanced system of reachability constraints such that R_1, \dots, R_n are ground.

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

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

The ground terms s_i^\perp and t_i^\perp are obtained from the source and target terms by replacement of every variable by the constant \perp .

Let $U = s_1^\perp \otimes t_1^\perp \otimes \dots \otimes s_n^\perp \otimes t_n^\perp$ and $(p_1, \dots, p_k) = pdp(s_1, t_1, \dots, s_n, t_n)$. We can use Lemma 5 to construct a TA A' such that

$$L(A') = \left\{ v_1 \otimes \dots \otimes v_k : v_1, \dots, v_k \in \mathcal{T}_\Sigma^n, \right. \\
\left. U[p_1 \leftarrow v_1, \dots, p_k \leftarrow v_k] \in L\left(\bigotimes_{i=1}^n A_i\right) \right\}$$

By hypothesis, the sequence $pd(s_1, t_2, \dots, s_n, t_n)$, denoted (u_1, \dots, u_{kn}) , is semi-linear. Therefore, it follows from Lemma 4 that there is a TA A'' such that

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

Note that both $L(A')$ and $L(A'')$ are subsets of \mathcal{T}_Σ^{kn} . Let A be a TA recognizing $L(A') \cap L(A'')$. We have that $L(A) \neq \emptyset$ if and only if S is satisfiable.

Let $t \in \mathcal{T}_\Sigma^{kn}$. Then $t \in L(A)$

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

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

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

iff $s_1\theta \xrightarrow{R_1} t_1\theta, \dots, s_n\theta \xrightarrow{R_n} t_n\theta$.

Lets now finally evaluate the size of A , the complexity of its construction being linearly proportional to 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 one constant independent from the problem size Therefore, $\|A'\| \leq M^{2cnk}$, see Lemma 5. According to Lemma 4,

$$\|A''\| \leq \|u_1\| \times \dots \times \|u_{kn}\| \leq \prod_{i=1}^n \|s_i\| \times \prod_{i=1}^n \|t_i\| \leq N^{2n},$$

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 2 SRR 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 $(R, f(x, \dots, x), f(q_1, \dots, q_n))$, where R is ground and q_1, \dots, q_n are constants. □

The balanced case can easily be used to show the decidability of the following case: for each variable x there exists an integer d_x such that x occurs only at positions of length d_x . For example with $s_1 = f(x, g(y))$, $t_1 = f(f(y, y), x)$, $s_2 = g(x)$, and $t_2 = g(f(a, y))$, first “guess” a term a , $g(x_1)$, or $f(x_1, x_2)$ for x to obtain a system where all variables occur at the same depth.

References

- Baader, F. & Nipkow, T. (1998), *Term Rewriting and All That*, Cambridge University Press.
- 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 manuscript.
- Coquidé, J., Dauchet, M., Gilleron, R. & Vágvolgyi, 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. Jouanaud, 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.
- Dershowitz, N. & Jouannaud, J.-P. (1990), Rewrite systems, *in* J. Van Leeuwen, ed., ‘Handbook of Theoretical Computer Science’, Vol. B: Formal Methods and Semantics, North Holland, Amsterdam, chapter 6, pp. 243–309.
- Doner, J. (1970), ‘Tree acceptors and some of their applications’, *Journal of Computer and System Sciences* **4**, 406–451.
- Farmer, W. (1991), ‘Simple second-order languages for which unification is undecidable’, *Theoretical Computer Science* **87**, 25–41.
- 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 \vec{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 semi-groups’, *Mat. Sbornik (in Russian)* **103**(2), 147–236. English Translation in American Mathematical Soc. Translations (2), vol. 117, 1981.
- Savitch, W. (1970), ‘Relationships between nondeterministic and deterministic tape complexities’, *Journal of Computer and System Sciences* **4**(2), 177–192.
- 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*.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from [ftp.mpi-sb.mpg.de](ftp://ftp.mpi-sb.mpg.de) under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL <http://www.mpi-sb.mpg.de>. If you have any questions concerning ftp or WWW access, please contact reports@mpi-sb.mpg.de. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
 Library
 attn. Birgit Hofmann
 Im Stadtwald
 D-66123 Saarbrücken
 GERMANY
 e-mail: library@mpi-sb.mpg.de

MPI-I-1999-2-003	U. Waldmann	Cancellative Superposition Decides the Theory of Divisible Torsion-Free Abelian Groups
MPI-I-1999-2-001	W. Charatonik	Automata on DAG Representations of Finite Trees
MPI-I-1999-1-001	A. Crauser, P. Ferragina	A Theoretical and Experimental Study on the Construction of Suffix Arrays in External Memory
MPI-I-98-2-018	F. Eisenbrand	A Note on the Membership Problem for the First Elementary Closure of a Polyhedron
MPI-I-98-2-017	M. Tzakova, P. Blackburn	Hybridizing Concept Languages
MPI-I-98-2-014	Y. Gurevich, M. Veanes	Partisan Corroboration, and Shifted Pairing
MPI-I-98-2-013	H. Ganzinger, F. Jacquemard, M. Veanes	Rigid Reachability
MPI-I-98-2-012	G. Delzanno, A. Podelski	Model Checking Infinite-state Systems in CLP
MPI-I-98-2-011	A. Degtyarev, A. Voronkov	Equality Reasoning in Sequent-Based Calculi
MPI-I-98-2-010	S. Ramangalahy	Strategies for Conformance Testing
MPI-I-98-2-009	S. Vorobyov	The Undecidability of the First-Order Theories of One Step Rewriting in Linear Canonical Systems
MPI-I-98-2-008	S. Vorobyov	AE-Equational theory of context unification is Co-RE-Hard
MPI-I-98-2-007	S. Vorobyov	The Most Nonelementary Theory (A Direct Lower Bound Proof)
MPI-I-98-2-006	P. Blackburn, M. Tzakova	Hybrid Languages and Temporal Logic
MPI-I-98-2-005	M. Veanes	The Relation Between Second-Order Unification and Simultaneous Rigid <i>E</i> -Unification
MPI-I-98-2-004	S. Vorobyov	Satisfiability of Functional+Record Subtype Constraints is NP-Hard
MPI-I-98-2-003	R.A. Schmidt	E-Unification for Subsystems of S4
MPI-I-98-2-002	F. Jacquemard, C. Meyer, C. Weidenbach	Unification in Extensions of Shallow Equational Theories
MPI-I-98-1-031	G.W. Klau, P. Mutzel	Optimal Compaction of Orthogonal Grid Drawings
MPI-I-98-1-030	H. Brönniman, L. Kettner, S. Schirra, R. Veltkamp	Applications of the Generic Programming Paradigm in the Design of CGAL
MPI-I-98-1-029	P. Mutzel, R. Weiskircher	Optimizing Over All Combinatorial Embeddings of a Planar Graph
MPI-I-98-1-028	A. Crauser, K. Mehlhorn, E. Althaus, K. Brengel, T. Buchheit, J. Keller, H. Krone, O. Lambert, R. Schulte, S. Thiel, M. Westphal, R. Wirth	On the performance of LEDA-SM

MPI-I-98-1-027	C. Burnikel	Delaunay Graphs by Divide and Conquer
MPI-I-98-1-026	K. Jansen, L. Porkolab	Improved Approximation Schemes for Scheduling Unrelated Parallel Machines
MPI-I-98-1-025	K. Jansen, L. Porkolab	Linear-time Approximation Schemes for Scheduling Malleable Parallel Tasks
MPI-I-98-1-024	S. Burkhardt, A. Crauser, P. Ferragina, H. Lenhof, E. Rivals, M. Vingron	q -gram Based Database Searching Using a Suffix Array (QUASAR)
MPI-I-98-1-023	C. Burnikel	Rational Points on Circles
MPI-I-98-1-022	C. Burnikel, J. Ziegler	Fast Recursive Division
MPI-I-98-1-021	S. Albers, G. Schmidt	Scheduling with Unexpected Machine Breakdowns
MPI-I-98-1-020	C. Rüb	On Wallace's Method for the Generation of Normal Variates
MPI-I-98-1-019		2nd Workshop on Algorithm Engineering WAE '98 - Proceedings
MPI-I-98-1-018	D. Dubhashi, D. Ranjan	On Positive Influence and Negative Dependence
MPI-I-98-1-017	A. Crauser, P. Ferragina, K. Mehlhorn, U. Meyer, E. Ramos	Randomized External-Memory Algorithms for Some Geometric Problems
MPI-I-98-1-016	P. Krysta, K. Lorys	New Approximation Algorithms for the Achromatic Number
MPI-I-98-1-015	M.R. Henzinger, S. Leonardi	Scheduling Multicasts on Unit-Capacity Trees and Meshes
MPI-I-98-1-014	U. Meyer, J.F. Sibeyn	Time-Independent Gossiping on Full-Port Tori
MPI-I-98-1-013	G.W. Klau, P. Mutzel	Quasi-Orthogonal Drawing of Planar Graphs
MPI-I-98-1-012	S. Mahajan, E.A. Ramos, K.V. Subrahmanyam	Solving some discrepancy problems in NC*
MPI-I-98-1-011	G.N. Frederickson, R. Solis-Oba	Robustness analysis in combinatorial optimization
MPI-I-98-1-010	R. Solis-Oba	2-Approximation algorithm for finding a spanning tree with maximum number of leaves
MPI-I-98-1-009	D. Frigioni, A. Marchetti-Spaccamela, U. Nanni	Fully dynamic shortest paths and negative cycle detection on digraphs with Arbitrary Arc Weights
MPI-I-98-1-008	M. Jünger, S. Leipert, P. Mutzel	A Note on Computing a Maximal Planar Subgraph using PQ-Trees
MPI-I-98-1-007	A. Fabri, G. Giezeman, L. Kettner, S. Schirra, S. Schönherr	On the Design of CGAL, the Computational Geometry Algorithms Library
MPI-I-98-1-006	K. Jansen	A new characterization for parity graphs and a coloring problem with costs
MPI-I-98-1-005	K. Jansen	The mutual exclusion scheduling problem for permutation and comparability graphs
MPI-I-98-1-004	S. Schirra	Robustness and Precision Issues in Geometric Computation
MPI-I-98-1-003	S. Schirra	Parameterized Implementations of Classical Planar Convex Hull Algorithms and Extreme Point Computations
MPI-I-98-1-002	G.S. Brodal, M.C. Pinotti	Comparator Networks for Binary Heap Construction
MPI-I-98-1-001	T. Hagerup	Simpler and Faster Static AC ⁰ Dictionaries
MPI-I-97-2-012	L. Bachmair, H. Ganzinger, A. Voronkov	Elimination of Equality via Transformation with Ordering Constraints
MPI-I-97-2-011	L. Bachmair, H. Ganzinger	Strict Basic Superposition and Chaining
MPI-I-97-2-010	S. Vorobyov, A. Voronkov	Complexity of Nonrecursive Logic Programs with Complex Values
MPI-I-97-2-009	A. Bockmayr, F. Eisenbrand	On the Chvátal Rank of Polytopes in the 0/1 Cube
MPI-I-97-2-008	A. Bockmayr, T. Kasper	A Unifying Framework for Integer and Finite Domain Constraint Programming
MPI-I-97-2-007	P. Blackburn, M. Tzakova	Two Hybrid Logics