Decidable fragments of simultaneous rigid reachability

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

MPI-I-1999-2-004 March 1999

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

We use $s, t, l, r, s_1, \ldots$ 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

$$|f(t_1, \ldots, t_n)| = |t_1| + \ldots + |t_n| + 1.$$
Positions in terms are sequences of integers. We use \( p, p_1 \) etc. for positions, \( \epsilon \) 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 \( p_1, \ldots, p_n \) are called parallel if they are pairwise incomparable 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 \( s \not\approx R t \), 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 \( s \not\rightarrow R t \), 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: \( \| R \| = \sum_{t \rightarrow r \in R} (\| t \| + \| r \|) \).

**Rigid Reachability.** A reachability constraint, or simply a constraint, in a signature \( \Sigma \) is a triple \((R, s, t)\) where \( R \) is a set of rules in \( \Sigma \), and \( s \) and \( t \) are \( \Sigma \)-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 \( \theta \) in \( \Sigma \), solves \((R, s, t)\) if \( \theta \) is grounding for \( R \), \( s \) and \( t \), and \( s \theta \not\rightarrow R\theta t \theta \). 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 & Vágyvölgyi 1994, Dauchet 1993], a tree automaton (or TA) \( A \) is a quadruple \((Q, \Sigma, R, F)\), where (i) \( Q \) is a finite set
of constants called states, (ii) $\Sigma$ is a finite signature that is disjoint from $Q$, (iii) $R$ is a system of rules of the form $f(q_1, \ldots, q_n) \rightarrow q$, where $f \in \Sigma$ has arity $n \geq 0$ and $q_1, \ldots, q_n \in Q$, 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 $\{t \in T_\Sigma \mid t \stackrel{\sigma}{\rightarrow}^* q\}$ 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 $\Sigma$ with the set of all rules $a(q) \rightarrow p$, also written as $q \xrightarrow{a} p$, 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 $a_1(a_2(\cdots a_n(q)))$ is written, using the reversed Polish notation, as the string $q a_n \cdots a_1$.

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 $A$, $q_n(\cdots a_2(a_1(q_0)) \cdots) \stackrel{\sigma}{\rightarrow}^* q$, i.e.,

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

The set of all strings accepted by $A$ is denoted by $L(A)$.

**Product automata.** Let $\Sigma$ be a signature, $m$ a positive integer, and $\bot$ a new constant. We write $\Sigma_\bot$ for $\Sigma \cup \{\bot\}$ and $\Sigma^m_\bot$ denotes the signature consisting of, for all $f_1, f_2, \ldots, f_m \in \Sigma_\bot$, 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 $t_i \in T_{\Sigma_\bot} \cup \bot$, $t_i = f_i(t_{i1}, \ldots, t_{ik})$, where $k_i \geq 0$, for $1 \leq i \leq m$. Let $k$ be the maximum of all the $k_i$ and let $t_{ij} = \bot$ 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:

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

For example:

$$f(c, g(c)) \otimes f(g(d), f(c, g(c))) = \langle ff \rangle(c \otimes g(d), g(c) \otimes f(c, g(c))) = \langle ff \rangle(\langle cg \rangle(\bot \otimes d), \langle gf \rangle(c \otimes c, \bot \otimes g(c))) = \langle ff \rangle(\langle cg \rangle(\langle \bot d \rangle, \langle gf \rangle(\langle cc \rangle, \langle \bot g \rangle(\bot \otimes c)))) = \langle ff \rangle(\langle cg \rangle(\langle \bot d \rangle, \langle gf \rangle(\langle cc \rangle, \langle \bot g \rangle(\langle \bot c \rangle))))$$
We write $T^{n}_\Sigma$ for the set of all $t$ in $T^n_\Sigma$ such that $t = t_1 \otimes \cdots \otimes t_m$ for some $t_1, \ldots, t_m \in T^n_\Sigma$. If $s \in T^n_\Sigma$ and $t \in T^n_\Sigma$, 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 $T^{n+m}_\Sigma$. Given a sequence $\vec{t} = t_1, \ldots, t_m$ of terms in $T^n_\Sigma \cup \bot$, 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 $\Sigma^n_1$ and $\Sigma^n_2$, respectively, the product of $A_1$ and $A_2$ is an automaton $A_1 \otimes A_2$ over $\Sigma^{n+n}_1$ 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 $q_{(i,j)}$ for all states $q_1$ in $A_1$ and $q_2$ in $A_2$, [see e.g. Conon et al. 1998]. In general, $\bigotimes_{i=1}^n A_i$ is defined accordingly.

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

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

## 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 $\Sigma$ and a positive integer $m$, we want to select only a certain subset from $\Sigma^m$ 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 $\Sigma$ and a set $I$ of selection constraints, we write $\Sigma^m|I$ for the following subset of $\Sigma^m$:

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

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

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 $\Sigma$. Let $(A_i)_{1 \leq i \leq n}$, $n \geq 1$, be a sequence of (string product) NFAs over the alphabets $\Sigma_i^{m_i}$ 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 $(\bigotimes_{i=1}^n L(A_i))|_I$ 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 $(A_i)_{1 \leq i \leq n}$, $m_i$, $m$, $\Sigma$, 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 $\Sigma_i^2$. (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 $\langle \bot, \bot \rangle v$ is accepted by $A_i$, e.g., we can assume that each automaton has a transition with label $\langle \bot, \bot \rangle$ 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 $\Sigma_i^{m_i}|_I$) Select $(a_1, \ldots, a_m) \in \Sigma_i^{m_i}|_I$ and store $a_i$ in **Letter**$_i$.

III (Guess the next transition of $(\bigotimes_{i=1}^n A_i)|_I$) 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 $\langle \text{Letter}_{2i-1} \text{Letter}_{2i}\rangle$-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 $(\bigotimes_{i=1}^n A_i)|_I$) 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 \( (\bigotimes_{i=1}^{n} A_i) | I \), 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 1) in \( L((\bigotimes_{i=1}^{n} A_i)) | I \) from the initial state to a final state. It is evident that the procedure always terminates, and that it accepts if and only if \( L((\bigotimes_{i=1}^{n} A_i)) | I \) 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.

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 \( L((\bigotimes_{i=1}^{n} A_i)) \{ i \approx i + 1 : 1 \leq i < n \} \) 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

- \((R, x, t)_i\), \(R\) 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 $S$. \[ \Box \]

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 $\Sigma$ be the signature of $S$. We may assume, by using Lemma 3, that $S$ is flat. Enumerate all the constraints in $S$ as $\rho_1, \ldots, \rho_m, \rho_{m+1}, \ldots, \rho_n$, where all the constraints of the form $(\emptyset, x, f(y))$ are enumerated 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 $1 \leq i \leq m$, 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 \( \Sigma \) consists of the unary function symbols \( f, g, \) and \( h \), and the constant \( c \). For example \( A \) recognizes the string \( \langle c \rangle \langle g \rangle \langle h \rangle \langle f \rangle \), i.e., the term \( \langle f \rangle \langle h \rangle \langle (g) \langle (f) \langle \langle c \rangle \rangle \rangle \rangle \) that is the same as \( f(h(g(c)))) \otimes h(g(c)) \).

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 \( \rho_i \) is a variable that occurs as the source of a \( \rho_j \), then \( 2i - 1 \approx 2j - 1 \in I \).

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

3. If the target of a \( \rho_i \) is a variable that occurs in the target of a \( \rho_j \), then \( 2i \approx 2j \in I \).

It remains to be proved that \( L(\bigotimes_{i=1}^n A_i) | I \) 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.

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{align*}
L(A_1) &= \{ s \otimes t : s \xrightarrow{R} t, s, t \in T_\Sigma \}, \\
L(A_2) &= \{ f(s) \otimes s : s \in T_\Sigma \}, \\
L(A_3) &= \{ g(s) \otimes s : s \in T_\Sigma \},
\end{align*}
and a set $I = \{ 1 \approx 3, 5 \approx 4, 6 \approx 2 \}$ of selection constraints. So $L(\bigotimes_{i=1}^{3} A_i) | I$ is as follows.
\begin{align*}
L(A_1 \otimes A_2 \otimes A_3) | I &= \{ s \otimes t \otimes f(u) \otimes u \otimes g(v) \otimes v : \\
&\quad s, t, u, v \in T_\Sigma, \ s \xrightarrow{R} t \} \{ 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 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 T_\Sigma, \ f(g(t)) \xrightarrow{R} t \}
\end{align*}
So, solvability of $S$ is equivalent to nonemptiness of $L(A_1 \otimes A_2 \otimes A_3) | I$.

### 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 $\Sigma$ 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) \(\Sigma\)-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 \(\Sigma\)-terms. Then the subset \(\{s_1\theta \otimes s_2\theta \otimes \cdots \otimes s_k\theta : \theta \text{ is a grounding } \Sigma\text{-substitution}\} \subseteq \mathcal{T}_{\Sigma}^s\) is recognized by a TA the size of which is in \(O((\|s_1\| + \|\Sigma\|) \ldots (\|s_k\| + \|\Sigma\|))\).

**Proof.** Let \(\Sigma\) and \(s = s_1, s_2, \ldots, s_k\) 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 \((\bigotimes A_i)|I\), where \(I\) is the set of all selection constraints \(i \approx j\) such that \(s_i\) and \(s_j\) are identical variables. \(\Box\)

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 \(\{s_1 \otimes \cdots \otimes s_k : s_1, \ldots, s_k \in \mathcal{T}_{\Sigma}, s[p_1 \leftarrow s_1, \ldots, p_k \leftarrow s_k] \in L(A)\}\)

**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 \(\vec{q}_i = q_{i1}, \ldots, q_{ik} \in Q\) such that, for some \(q_{i} \in F\), \(s[p_1 \leftarrow q_{i1}, \ldots, p_k \leftarrow q_{ik}] \xrightarrow{\vec{q}_i} q_{i}\). 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 \(m \leq |Q|^k\). The size of \(A'\) is therefore \(\|A'\| \leq \sum_{i=1}^m \|A_i\| \leq \sum_{i=1}^m \|A\|^k \leq |Q|^k \times \|A\|^k\) \(\Box\)

11
4.2 Parallel decomposition of sequences of terms

For technical reasons, we generalize the notion of a product of \( m \) terms by allowing non-ground 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 \( \vec{s} = s_1, \ldots, s_m \) of terms and let \( (\otimes(t_i))_{1 \leq i \leq k} \) be the sequence of all the subterms of the product term \( \bigotimes \vec{s} \) which have head symbol \( \otimes \). The parallel decomposition of \( \vec{s} = s_1, \ldots, s_m \) 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: \( pd(s, t) \) is the sequence \( (p_i)_{1 \leq i \leq k} \), where \( p_i \) is the position of \( \otimes(t_i) \) 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 \( \Sigma \)-terms, and let \( R \) be a ground rewrite system over \( \Sigma \). We will show how to capture all the solutions of the reachability constraint \( (R, s, t) \) as a certain regular set of \( \Sigma^2 \)-terms. First, construct the product \( s \otimes t \).

\[
s \otimes t \ = \ f(g(z), g(x)) \otimes f(y, f(x, y))
\]
\[
= \ (ff)(g(z) \otimes y, g(x) \otimes f(x, y))
\]
\[
= \ (ff)(\otimes(g(z), y), \langle gf \rangle(x \otimes x, \bot \otimes y))
\]
\[
= \ (ff)(\otimes(g(z), y), \langle gf \rangle(\otimes(x, x), \otimes(\bot, y)))
\]

The preorder traversal of \( s \otimes t \) yields the sequence \( \otimes(g(z), y), \otimes(x, x), \otimes(\bot, y) \).

Finally, \( pd(s, t) \) is the semi-linear sequence \( g(z), y, x, x, \bot, y \). (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 \( L(A_1) = \{ g(z\theta) \otimes y\theta \otimes x\theta \otimes x\theta \otimes \bot \otimes y\theta : \theta \text{ is a grounding } \Sigma \text{-substitution} \} \).

Now, consider a TA \( A_R \) that recognizes the product of \( \frac{1}{R} \), see Lemma 1, i.e., \( L(A_R) = \{ u \otimes v : u \overset{\downarrow}{R} v, u, v \in T_2 \} \). From \( A_R \) we can, by using Lemma 5, construct a TA \( A_2 \) such that

\[
L(A_2) = \{ s_1 \otimes s_2 : s_1, s_2 \in T_2^2, \langle ff \rangle(s_1, \langle gf \rangle(s_21, s_22)) \in L(A_R) \}
\]
Let $A$ recognize $L(A_1) \cap L(A_2)$. We get that

$$L(A) = L(A_1) \cap L(A_2)$$

$$= \left\{ s_1 \otimes s_{21} \otimes s_{22} : \begin{array}{l}
(\exists x\theta, y\theta, z\theta \in T_{\Sigma}) \\
\quad s_1 = g(z\theta) \otimes y\theta, \ s_{21} = x\theta \otimes x\theta, \ s_{22} = \bot \otimes y\theta,
\quad \langle ff\rangle(s_1, \langle gf\rangle(s_{21}, s_{22})) \in L(A_R)
\end{array} \right\}$$

$$= \{g(z\theta) \otimes \cdots \otimes y\theta : \langle ff\rangle(g(z\theta) \otimes y\theta, \langle gf\rangle(x\theta \otimes x\theta, \bot \otimes y\theta)) \in L(A_R)\}$$

$$= \{g(z\theta) \otimes \cdots \otimes y\theta : \theta \text{ solves } (R, s, t)\}$$

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 $\text{pd}(s_1, t_1, s_2, t_2, \ldots, s_n, t_n)$ 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.

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

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

$$L(A_i) = \{u \otimes v : u \xrightarrow{R_i} v, \ u, v \in T_{\Sigma}\}$$

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

Let $U = s_1^\bot \otimes t_1^\bot \otimes \cdots \otimes s_n^\bot \otimes t_n^\bot$ and $(p_1, \ldots, p_k) = \text{pd}(s_1, t_1, \ldots, s_n, t_n)$. We can use Lemma 5 to construct a TA $A'$ such that

$$L(A') = \left\{ v_1 \otimes \cdots \otimes v_k : v_1, \ldots, v_k \in T_{\Sigma}^n, \right\}$$

$$U[p_1 \leftarrow v_1, \ldots, p_k \leftarrow v_k] \in L(\bigotimes_{i=1}^n A_i)$$

13
By hypothesis, the sequence \( pd(s_1, t_2, \ldots, s_n, t_n) \), denoted \((u_1, \ldots, u_{kn})\), is semi-linear. Therefore, it follows from Lemma 4 that there is a TA \( A'' \) such that

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

Note that both \( L(A') \) and \( L(A'') \) are subsets of \( T_{\Sigma}^{kn} \). 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 \( t \in T_{\Sigma}^{kn} \). Then \( t \in L(A) \)

\[
\text{iff } t = u_1 \theta \otimes \ldots \otimes u_{kn} \theta \text{ for some grounding } \Sigma\text{-substitution } \theta \text{ (} t \in L(A'') \), and } \]

\[
U[p_1 \leftarrow w_1, \ldots, p_k \leftarrow w_k] \in L(\bigotimes_{i=1}^{n} A_i), \text{ where } w_i = \bigotimes_{j=(i-1)n+1}^{in} u_j \theta \text{ (} t \in L(A'') \),
\]

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

\[
\text{iff } s_1 \theta \rightarrow_{R_1} t_1 \theta, \ldots, s_n \theta \rightarrow_{R_n} t_n \theta.
\]

Let 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,

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

where \( N = \max \{ ||s_i||, ||t_i|| : i \leq n \} \). Hence,

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

**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 \((R, f(x, \ldots, x), f(q_1, \ldots, q_n))\), where \( R \) is ground and \( q_1, \ldots, q_n \) 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


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 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. Ganziinger, F. Jacquesmard, 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. Ramasubdy 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-R-E-Hard
MPI-i-98-2-007 S. Vorobyov The Most Non-elementary 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 E-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. Jacquesmard, C. Meyer, C. Weidenbacher Unification in Extensions of Substitutions of Equational Theories
MPI-i-98-1-031 G.W. Klaau, P. Mutzel Optimal Compaction of Orthogonal Grid Drawings
MPI-i-98-1-030 H. Druennman, L. Kettnser, S. Schirra, R. Védtamp 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
<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI-1-98-1-027</td>
<td>C. Bernié</td>
<td>Delaunay Graphs by Divide and Conquer</td>
</tr>
<tr>
<td>MPI-1-98-1-026</td>
<td>K. Jansen, L. Porkolab</td>
<td>Improved Approximation Schemes for Scheduling Unrelated Parallel Machines</td>
</tr>
<tr>
<td>MPI-1-98-1-025</td>
<td>K. Jansen, L. Porkolab</td>
<td>Linear-Time Approximation Schemes for Scheduling Malleable Parallel Tasks</td>
</tr>
<tr>
<td>MPI-1-98-1-024</td>
<td>S. Burkhard, A. Crauser, P. Ferragina, H. Lenhof, E. Rivals, M. Vingron</td>
<td>q-gram Based Database Searching Using a Suffix Array (QUASAR)</td>
</tr>
<tr>
<td>MPI-1-98-1-023</td>
<td>C. Burnikel</td>
<td>Rational Points on Circles</td>
</tr>
<tr>
<td>MPI-1-98-1-022</td>
<td>C. Burnikel, J. Ziegler</td>
<td>Fast Recursive Division</td>
</tr>
<tr>
<td>MPI-1-98-1-021</td>
<td>S. Albers, G. Schmidt</td>
<td>Scheduling with Unexpected Machine Breakdowns</td>
</tr>
<tr>
<td>MPI-1-98-1-020</td>
<td>C. Rübb</td>
<td>On Wallace's Method for the Generation of Normal Variates</td>
</tr>
<tr>
<td>MPI-1-98-1-019</td>
<td></td>
<td>2nd Workshop on Algorithm Engineering WAE '98 - Proceedings</td>
</tr>
<tr>
<td>MPI-1-98-1-018</td>
<td>D. Dubhashi, D. Ranjan</td>
<td>On Positive Influence and Negative Dependence</td>
</tr>
<tr>
<td>MPI-1-98-1-016</td>
<td>P. Krysta, K. Loryș</td>
<td>New Approximation Algorithms for the Achromatic Number</td>
</tr>
<tr>
<td>MPI-1-98-1-015</td>
<td>M.R. Henzinger, S. Leonardi</td>
<td>Scheduling Multicasts on Unit-Capacity Trees and Meshes</td>
</tr>
<tr>
<td>MPI-1-98-1-014</td>
<td>U. Meyer, J.F. Sibeyn</td>
<td>Time-Independent Gossiping on Full-Port Tori</td>
</tr>
<tr>
<td>MPI-1-98-1-013</td>
<td>G.W. Klau, P. Mutzel</td>
<td>Quasi-Orthogonal Drawing of Planar Graphs</td>
</tr>
<tr>
<td>MPI-1-98-1-012</td>
<td>S. Mahajan, E.A. Ramos, K.V. Subrahmanyam</td>
<td>Solving some discrepancy problems in NC*</td>
</tr>
<tr>
<td>MPI-1-98-1-011</td>
<td>G.N. Frederickson, R. Solís-Oba</td>
<td>Robustness analysis in combinatorial optimization</td>
</tr>
<tr>
<td>MPI-1-98-1-010</td>
<td>R. Solís-Oba</td>
<td>2-Approximation algorithm for finding a spanning tree with maximum number of leaves</td>
</tr>
<tr>
<td>MPI-1-98-1-009</td>
<td>D. Frigioni, A. Marchetti-Spaccamela, U. Nanni</td>
<td>Fully dynamic shortest paths and negative cycle detection on digraphs with Arbitrary Arc Weights</td>
</tr>
<tr>
<td>MPI-1-98-1-008</td>
<td>M. Jünger, S. Léperti, P. Mutzel</td>
<td>A Note on Computing a Maximal Planar Subgraph using PQ-Trees</td>
</tr>
<tr>
<td>MPI-1-98-1-007</td>
<td>A. Fabri, G. GüeXman, L. Kettner, S. Schirra, S. Schönherr</td>
<td>On the Design of CGAL, the Computational Geometry Algorithms Library</td>
</tr>
<tr>
<td>MPI-1-98-1-006</td>
<td>K. Jansen</td>
<td>A new characterization for parity graphs and a coloring problem with costs</td>
</tr>
<tr>
<td>MPI-1-98-1-005</td>
<td>K. Jansen</td>
<td>The mutual exclusion scheduling problem for permutation and comparability graphs</td>
</tr>
<tr>
<td>MPI-1-98-1-004</td>
<td>S. Schirra</td>
<td>Robustness and Precision Issues in Geometric Computation</td>
</tr>
<tr>
<td>MPI-1-98-1-003</td>
<td>S. Schirra</td>
<td>Parameterized Implementations of Classical Planar Convex Hull Algorithms and Extreme Point Computations</td>
</tr>
<tr>
<td>MPI-1-98-1-002</td>
<td>G.S. Brodal, M.C. Pinotti</td>
<td>Comparator Networks for Binary Heap Construction</td>
</tr>
<tr>
<td>MPI-1-98-1-001</td>
<td>T. Hagerup</td>
<td>Simpler and Faster Static AC$^0$ Dictionaries</td>
</tr>
<tr>
<td>MPI-1-97-2-012</td>
<td>L. Bachmaier, H. Ganzinger, A. Voroskov</td>
<td>Elimination of Equality via Transformation with Ordering Constraints</td>
</tr>
<tr>
<td>MPI-1-97-2-011</td>
<td>L. Bachmaier, H. Ganzinger</td>
<td>Strict Basic Superposition and Chaining</td>
</tr>
<tr>
<td>MPI-1-97-2-010</td>
<td>S. Vorobyov, A. Voroskov</td>
<td>Complexity of Nonrecursive Logic Programs with Complex Values</td>
</tr>
<tr>
<td>MPI-1-97-2-009</td>
<td>A. Bockmayr, F. Eisenbrand</td>
<td>On the Chvátal Rank of Polytopes in the 0/1 Cube</td>
</tr>
<tr>
<td>MPI-1-97-2-008</td>
<td>A. Bockmayr, T. Kasper</td>
<td>A Unifying Framework for Integer and Finite Domain Constraint Programming</td>
</tr>
<tr>
<td>MPI-1-97-2-007</td>
<td>P. Blackburn, M. Tzakova</td>
<td>Two Hybrid Logics</td>
</tr>
</tbody>
</table>