Symbolic Reachability Analysis Using Narrowing and its Application to Verification of Cryptographic Protocols [*]

José Meseguer, and Prasanna Thati

Department of Computer Science, UIUC, Urbana-Champaign, USA

Abstract

Narrowing was introduced, and has traditionally been used, to solve equations in initial and free algebras *modulo* a set of equations E. This paper proposes a generalization of narrowing which can be used to solve *reachability goals* in initial and free models of a rewrite theory $\mathcal{R}$. We show that narrowing is sound and weakly complete (i.e., complete for normalized solutions) under reasonable executability assumptions about $\mathcal{R}$. We also show that in general narrowing is not strongly complete, that is, not complete when some solutions can be further rewritten by $\mathcal{R}$. We then identify several large classes of rewrite theories, covering many practical applications, for which narrowing is strongly complete. Finally, we illustrate an application of narrowing to analysis of cryptographic protocols.

Key words: Rewriting logic, narrowing, reachability, security protocols.

1 Introduction

This paper addresses the following technical question. Given a rewrite theory $\mathcal{R}$ satisfying reasonable assumptions, is there a general deductive procedure to solve *reachability problems* for $\mathcal{R}$? By a "reachability problem" we mean the obvious, that is, an existential formula

$$(\exists \overrightarrow{x})\ t \rightarrow^* t'$$

or, more generally, an existentially quantified conjunction of such reachability goals. Since $\mathcal{R}$ typically specifies either a concurrent system or an inference system, the meaning and interest of solving such goals is quite obvious. The terms t and t' denote sets of states in the initial model of $\mathcal{R}$, and we want to know for what subset of the states denoted by t we can reach the set denoted

[*] Research supported by ONR Grant N00014-02-1-0715 and NSF Grant CCR-0234524

by t'. Under finite state assumptions, questions of this kind can be answered by model checking techniques [9]. Our interest, however, is in general methods that do not require finiteness assumptions and can complement such model checking techniques. In this paper, we generalize *narrowing* from a technique for solving equality goals [16,21,23] to one for solving reachability goals; indeed equational narrowing goals can be viewed as a special case of reachability goals.

That narrowing in this more general sense should be developed as a method for analyzing concurrent systems and should fit within a wider spectrum of analysis capabilities, was first proposed in [12]. One can view narrowing as a new form of "symbolic model checking", available also for infinite state systems, where the word "symbolic", instead of having the more restricted sense of representing finite sets of states by Boolean propositions, is widened to mean the representation of possibly infinite sets of states by terms with logical variables. These methods could even have useful applications in the case of finite-state systems that are too large to analyze by standard model checking techniques.

There are indeed a number of techniques actively investigated to analyze infinite state systems, including model checking for suitable subclasses, e.g. [4,5,15,17], abstraction techniques, e.g. [10,26,19,25,40], tree-automata based reachability analyses, e.g. [18,35], and theorem proving, e.g. [37,36]. We think that narrowing is a promising additional technique to be further explored. Indeed, narrowing like techniques have already been shown useful in the analysis of cryptographic protocols [2,22,29].

We formally define narrowing for *order-sorted unconditional* rewrite theories of the form $\mathcal{R} = (\Sigma, E, R)$ where $E = \Delta \cup B$, with Δ confluent and terminating modulo B. We prove soundness of solutions found for reachability problems using narrowing, and also show that the narrowing procedure is *weakly* complete in the following sense: if ρ is a solution of a given reachability problem and ρ is normalized with respect to rewriting with the rules R modulo E, then the narrowing procedure finds a solution η that subsumes ρ modulo E. This weak completeness result holds under reasonable executability assumptions about the given rewrite theory.

We also show that in general, narrowing is *not* complete in the following stronger sense: if ρ is a (not necessarily normalized) solution of a reachability goal, then the narrowing procedure finds a solution η that subsumes ρ modulo E. Hence the "weakness" in completeness of narrowing. This does not hold in general, as we show by several examples. The point is that in equational narrowing [23], confluence and termination are reasonable assumptions; by contrast, the rewrite rules R specifying a concurrent system are typically nonconfluent and nonterminating; indeed, termination may often have the meaning of an undesirable deadlock. All this implies that in general rewriting may also happen in the substitutions themselves, making narrowing incomplete in the strong sense.

A key question to investigate is identifying interesting classes of rewrite theories for which narrowing *is* complete in the strong sense. We prove that several important classes covering many practical applications have strongly complete solutions to reachability goals by narrowing. The first important such class is that of *topmost* rewrite theories, that is, theories in which terms can only be rewritten at the top. We then show how other large classes of rewrite theories, including, for example, most object-oriented distributed systems, a wide range of Petri net models, grammars, and many reflective distributed systems structured with a "Russian dolls" architecture [32] can be transformed into equivalent topmost rewrite theories with the same set of solutions for a given reachability problem. We furthermore establish a strong completeness result for the class of rewrite theories $\mathcal{R} = (\Sigma, \Delta \cup B, R)$ such that equations in B are regular (LHS and RHS have the same set of variables) and linear (LHS and RHS are linear terms), Δ is confluent and terminating modulo B, and R is right linear (RHS is linear).

As an example application, we show how narrowing can be used for analysis of security protocols. Many security protocol properties, such as the *secrecy* and *authenticity*, can be characterized as reachability problems. We show how the strong completeness results for topmost theories can be exploited to verify the secrecy property of a protocol when the number of protocol sessions is bounded. This technique can also be adapted to verify other security properties, including authenticity. A noteworthy feature of our analysis technique is that narrowing modulo equations provides a general procedure that can uniformly handle analysis of security protocols that employ cryptographic primitives with visible algebraic properties that can be exploited by an intruder (such as in the case of `xor` encryption and Diffie-Hellman exponentiation) [7,8,11,34,39].

2 Background

An *order-sorted signature* Σ is defined by a set of sorts S, a partial order relation of subsort inclusion $\leq$ on S, and an $S^* \times S$-indexed family of $\{\Sigma_{w,s}\}_{(w,s) \in S^* \times S}$ of operations. We denote $f \in \Sigma_{w,s}$ by $f : w \to s$. We define a relation $\equiv$ on S as the smallest equivalence relation that such that $s \leq s'$ implies $s \equiv s'$. We assume that each equivalence class of sorts contains a *top* [1] sort that is a supersort of every other sort in the class. Formally, for every sort s we assume that there is a sort $[s]$ such that $s \equiv s'$ implies $s' \leq [s]$. Furthermore, for each $f : s_1 \times \ldots \times s_n \to s$ we assume there is also an $f : [s_1] \times \ldots \times [s_n] \to [s]$. We require the signature Σ to be sensible, i.e., whenever we have $f : w \to s$ and $f : w' \to s'$ with w, w' of equal length then $w \equiv w'$ implies $s \equiv s'$.

[1] Note that this top sort plays the role of an "error supersort", or a *kind* in membership equational logic [31], although in some cases there may not be any real "error expressions" in this top sort because all terms in it happen to be well-defined.

A Σ-*algebra* is defined by an S-indexed family of sets $A = \{A_s\}_{s \in S}$ such that $s \leq s'$ implies $A_s \subseteq A_{s'}$, and for each function $f : w \to s$ with $w = s_1 \times \ldots \times s_n$ a function $f_A^{w,s} : A_{s_1} \times \ldots \times A_{s_n} \to A_s$. Further, we require that subsort overloaded operations agree, i.e., for each $f : w \to s$ and $(a_1, \ldots, a_n) \in A^w$ we require $f_A^{w,s}(a_1, \ldots, a_n) = f_A^{[w],[s]}(a_1, \ldots, a_n)$, where if $w = s_1 \times \ldots \times s_n$, then $[w] = [s_1] \times \ldots \times [s_n]$. We assume a family $X = \{X_s\}_{s \in S}$ of infinite sets of variables such that $s \neq s'$ implies $X_s \cap X_{s'} = \emptyset$, and the variables in X are different from constant symbols in Σ. We denote the set of ground Σ-terms and Σ-terms of sort s by $T_{\Sigma,s}$ and $T_\Sigma(X)_s$ respectively. We write T_Σ for the Σ-algebra of ground terms over Σ, and $T_\Sigma(X)$ for the Σ-algebra of terms with variables from the set X.

We use a finite sequence of positive integers, called a *position*, to denote an access path in a term. We let ω range over positions. For $t \in T_\Sigma(X)$ let $\mathit{Var}(t), \mathit{Pos}(t), \mathit{FuPos}(t)$ denote the set of variables, positions, and non-variable positions in t, respectively. The root of a term is at position ϵ. We denote the subterm of t at position ω by $t|_\omega$.

A *substitution* is a mapping $\sigma : X \to T_\Sigma(X)$ which maps variables to terms of the same sort, and which is different from the identity for a finite subset $\mathit{Dom}(\sigma)$ of X. We denote the homomorphic extension of σ to $T_\Sigma(X)$ also by σ. The set of variables introduced by σ is $\mathit{Ran}(\sigma) = \cup_{x \in \mathit{Dom}(\sigma)} \mathit{Var}(\sigma(x))$. The restriction of a substitution σ to a set of variables V is defined as

$$\sigma|_V(x) = \begin{cases} \sigma(x) & \text{if } x \in V \\ x & \text{otherwise} \end{cases}$$

We say that a substitution σ is *away* from a set of variables V if $\mathit{Ran}(\sigma) \cap V = \emptyset$. For substitutions σ, ρ such that $\mathit{Dom}(\sigma) \cap \mathit{Dom}(\rho) = \emptyset$ we define their union as

$$(\sigma \cup \rho)(x) = \begin{cases} \sigma(x) & \text{if } x \in \mathit{Dom}(\sigma) \\ \rho(x) & \text{if } x \in \mathit{Dom}(\rho) \\ x & \text{otherwise} \end{cases}$$

A Σ-*equation* is an expression of the form $t = t'$ where $t, t' \in T_\Sigma(X)_{[s]}$ for an appropriate $[s]$. Order-sorted equational logic has a sound and complete inference system $E \vdash_\Sigma$ (see [31]) inducing for any set of variables Y a congruence relation $=_E^Y$ on terms $t, t' \in T_\Sigma(Y)$: $t =_E^Y t'$ if and only if $E \vdash_\Sigma (\forall Y) t = t'$. For the sake of simplicity, we will assume that all sorts in Σ are non-empty, i.e., that for each sort there is a ground term of that sort. In that case, if $t, t' \in T_\Sigma(X) \cap T_\Sigma(Y)$ then $t =_E^X t'$ if and only if $t =_E^Y t'$. Therefore, the superscript notation $=_E^Y$ becomes unnecessary and we can just write $=_E$. Because of our assumptions about the signature Σ it is the case that $t =_E t'$, $t \in T_\Sigma(X)_s$, and $t' \in T_\Sigma(X)_{s'}$ implies $s \equiv s'$.

An equation $t = t'$ is said to be (i) *regular* if $\mathit{Var}(t) = \mathit{Var}(t')$, (ii) *sort* *preserving* if for each substitution σ we have $\sigma(t) \in T_{\Sigma}(X)_s$ if and only if $\sigma(t') \in T_{\Sigma}(X)_s$, (iii) *sort-decreasing* if for each substitution σ we have $\sigma(t) \in T_{\Sigma}(X)_s$ implies $\sigma(t') \in T_{\Sigma}(X)_s$, (iv) *left (or right) linear* if t (resp. t') is *linear* (i.e., each variable occurs at a single position), and (v) *linear* if it is both left and right linear. A set of equations E is said to be regular, or sort decreasing, or sort preserving, or (left or right) linear, if each equation in it is so.

The E*-subsumption* preorder $\ll_E$ on $T_{\Sigma}(X)$ is defined by $t \ll_E t'$ if there is a substitution σ such that $\sigma(t) =_E t'$; such a substitution σ is said to be an E*-match* from t to t'. For substitutions σ, ρ and a set of variables V we define $\sigma|_V =_E \rho|_V$ if $\sigma(x) =_E \rho(x)$ for all $x \in V$, and $\sigma|_V \ll_E \rho|_V$ if there is a substitution η such that $\rho|_V =_E (\eta \circ \sigma)|_V$. The following is a useful lemma.

Lemma 2.1 **([3])** *For substitutions σ, ρ and sets of variables $V \subseteq W$ let $Dom(\sigma) \cap W \subseteq V$ and $Ran(\sigma) \cap W = \emptyset$. Then $\sigma|_V \ll_E \rho|_V$ implies $\sigma|_W \ll_E \rho|_W$.* □

A *system of equations* F is an expression of the form $t_1 = t'_1 \wedge \ldots \wedge t_n = t'_n$, where each $t_i = t'_i$ is a Σ-equation. We define $Var(F) = \bigcup_i Var(t_i) \cup Var(t'_i)$. An E*-unifier* for F is a substitution σ such that $\sigma(t_i) =_E \sigma(t'_i)$ for $1 \leq i \leq n$. For $V = Var(F) \subseteq W$, a set of substitutions $CSU_E(F, W)$ is said to be a *complete* set of unifiers of F away from W if

- Each $\sigma \in CSU_E(F, W)$ is an E-unifier of F.
- For any E-unifier ρ of F there is a $\sigma \in CSU_E(F, W)$ such that $\sigma|_V \ll_E \rho|_V$.
- For all $\sigma \in CSU_E(F, W)$, $Dom(\sigma) \subseteq V$ and $Ran(\sigma) \cap W = \emptyset$.

An E-unification algorithm is *complete* if for any given system of equations it generates a complete set of E-unifiers. Note that this set need not be finite. A unification algorithm is said to be *finite* and complete if it terminates after generating a finite and complete set of solutions.

A *rewrite rule* is an expression of the form $l \to r$ where $l, r \in T_{\Sigma}(X)_{[s]}$ for an appropriate $[s]$. An *(unconditional) order-sorted rewrite theory* is a triple $\mathcal{R} = (\Sigma, E, R)$ with Σ an order-sorted signature, E a set of Σ-equations, and R a set of rewrite rules. We only consider rewrite theories $\mathcal{R}$ where for each rule $l \to r$ in R we have $Var(r) \subseteq Var(l)$. We define the *one-step rewrite relation* on $T_{\Sigma}(X)$ as follows: $t \to_R t'$ if there is an $\omega \in Pos(t)$, a rule $l \to r$ in R, and a substitution σ such that $t|_{\omega} = \sigma(l)$ and $t' = t[\omega \leftarrow \sigma(r)]$. The reader may check that because of our assumption about the signature Σ, it is the case that t' is always well-sorted, and $t \in T_{\Sigma}(X)_{[s]}$ implies $t' \in T_{\Sigma}(X)_{[s]}$. Let $\to_{R/E}$ be the relation $=_E \circ \to_R \circ =_E$. A term $t \in T_{\Sigma}(X)$ is called R/E*-irreducible* if there is no $t' \in T_{\Sigma}(X)$ such that $t \to_{R/E} t'$. Note that the reflexive transitive closure relation $\to^*_{R/E}$ defines the inferences of the rewrite theory $\mathcal{R}$ in the order-sorted version of the usual sequent-style presentation [30]. That is, for any $t, t' \in T_{\Sigma}(X)_{[s]}$ we have $t \to^*_{R/E} t'$ if and only if $\mathcal{R} \vdash [t]_E \to [t']_E$, where $[t]_E$ denotes the equivalence class of t modulo E.

For substitutions σ, ρ and a set of variables V we define $\sigma|_V \to_R \rho|_V$

if there is $x \in V$ such that $\sigma(x) \to_R \rho(x)$ and for all other $y \in V$ we have $\sigma(y) = \rho(y)$. The relation $\to_{R/E}$ on substitutions is defined as $=_E \circ \to_R \circ =_E$. A substitution σ is called *R/E-normalized* if $\sigma(x)$ is R/E-irreducible for all x; note that this is a stronger condition than saying there is no substitution ρ such that $\sigma|_X \to_{R/E} \rho|_X$ (because rules in R need not be sort-decreasing).

3 Reachability Goals

Given an order-sorted rewrite theory $\mathcal{R} = (\Sigma, E, R)$, a *reachability goal* G is a conjunction of the form $t_1 \to^* t'_1 \wedge \ldots \wedge t_n \to^* t'_n$, where for $1 \leq i \leq n$, $t_i, t'_i \in T_\Sigma(X)_{[s_i]}$ for appropriate $[s_i]$. We say that t_i are the *sources* of the goal G, while t'_i are the *targets*. We define $\mathit{Var}(G) = \bigcup_i \mathit{Var}(t_i) \cup \mathit{Var}(t'_i)$. A substitution σ is an *$\mathcal{R}$-solution* of G (or just a solution for short, when $\mathcal{R}$ is clear from the context) if $\sigma(t_i) \to^*_{R/E} \sigma(t'_i)$ for $1 \leq i \leq n$. We define $\mathcal{E}(G)$ to be the system of equations $t_1 = t'_1 \wedge \ldots \wedge t_n = t'_n$. We say σ is a *trivial solution* of G if it is an E-unifier for $\mathcal{E}(G)$. We say G is trivial if the identity substitution *id* is a trivial solution of G. Thus, σ is a trivial solution of G if and only if $\sigma(G)$ is trivial.

For goals $G : t_1 \to^* t_2 \wedge \ldots \wedge t_{2n-1} \to^* t_{2n}$ and $G' : t'_1 \to^* t'_2 \wedge \ldots \wedge t'_{2n-1} \to^* t'_{2n}$ we say $G =_E G'$ if $t_i =_E t'_i$ for all $1 \leq i \leq 2n$. We say $G \to_R G'$ if there is an odd i such that $t_i \to_R t'_i$ and for all $j \neq i$ we have $t_j = t'_j$. The relation $\to_{R/E}$ over goals is defined as $=_E \circ \to_R \circ =_E$.

Lemma 3.1 *σ is a solution of a reachability goal G if and only if $\sigma(G) \to^*_{R/E} G'$ for some trivial goal G'.* □

A set of substitutions Γ is said to be a *complete set of $\mathcal{R}$-solutions* of G if (i) every $\sigma \in \Gamma$ is an $\mathcal{R}$-solution of G, and (ii) for any $\mathcal{R}$-solution ρ of G there is a $\sigma \in \Gamma$ such that $\sigma|_{\mathit{Var}(G)} \ll_E \rho|_{\mathit{Var}(G)}$. We are interested in finding a complete set of $\mathcal{R}$-solutions for a given goal G and an order-sorted (unconditional) rewrite theory $\mathcal{R}$.

Since E-congruence classes can be infinite, $\to_{R/E}$-reducibility is undecidable in general. One way to get around this problem is to "implement" R/E-rewriting by a combination of rewriting using oriented equations and rules. Such an approach was proposed, for instance, by Patrick Viry [42] (for the unsorted case). We adopt this approach in this paper.

We assume that $E = \Delta \cup B$ such that (i) B is regular and sort preserving, (ii) B has a finite and complete unification algorithm (note that this implies that B-matching is decidable) and $\Delta \cup B$ has a complete (and not necessarily finite) unification algorithm[2], (iii) for each $t = t'$ in Δ we have $\mathit{Var}(t') \subseteq \mathit{Var}(t)$, and (iv) Δ is *sort-decreasing*, and is *confluent and terminating modulo B*.

[2] With certain additional assumptions such as B-coherence of Δ, B rewriting [23], it is the case that $\Delta \cup B$ has a complete unification algorithm by equational narrowing. But we also allow the possibility of special-purpose unification algorithms for $\Delta \cup B$.

We define the relation $\rightarrow_{\Delta,B}$ on $T_\Sigma(X)$ as follows: $t \rightarrow_{\Delta,B} t'$ if there is an $\omega \in Pos(t)$, $l = r$ in Δ, and a substitution σ such that $t|_\omega =_B \sigma(l)$ and $t' = t[\omega \leftarrow \sigma(r)]$. Note that, since B is sort-preserving and Δ is sort-decreasing, it is the case that t' is well-sorted, and $t \in T_\Sigma(X)_s$ implies $t' \in T_\Sigma(X)_s$. The relation $\rightarrow_{R,B}$ is similarly defined, and because of our assumption about the signature Σ, it is the case that $t \rightarrow_{R,B} t'$ implies t' is well-sorted, and $t \in T_\Sigma(X)_{[s]}$ implies $t' \in T_\Sigma(X)_{[s]}$. We define $\rightarrow_{R\cup\Delta,B}$ as $\rightarrow_{R,B} \cup \rightarrow_{\Delta,B}$. Note that, since B-matching is decidable, $\rightarrow_{\Delta,B}$, $\rightarrow_{R,B}$, and $\rightarrow_{R\cup\Delta,B}$ are decidable. These three relations are lifted to goals and substitutions as expected. $R \cup \Delta, B$-normalized (and similarly R, B or Δ, B-normalized) substitutions are defined as expected.

The idea is to implement $\rightarrow_{R/E}$ on (terms and goals) using $\rightarrow_{R\cup\Delta,B}$. For this to work, we need the following additional assumptions.

- We assume that $\rightarrow_{\Delta,B}$ is *coherent with* B, i.e., $\forall t_1, t_2, t_3$ we have $t_1 \rightarrow^+_{\Delta,B} t_2$ and $t_1 =_B t_3$ implies $\exists t_4, t_5$ such that $t_2 \rightarrow^*_{\Delta,B} t_4$, $t_3 \rightarrow^+_{\Delta,B} t_5$ and $t_4 =_E t_5$ [23].

$$\begin{tikzcd}
t_1 \arrow[r, "\rightarrow^+_{\Delta,B}"] \arrow[d, dash, "||_B"] & t_2 \arrow[r, "\rightarrow^*_{\Delta,B}"] & t_4 \arrow[d, dash, "||_E"] \\
t_3 \arrow[rr, "\rightarrow^+_{\Delta,B}"] & & t_5
\end{tikzcd}$$

- We assume $\rightarrow_{R,B}$ is *E-consistent with* B, i.e. $\forall t_1, t_2, t_3$ we have $t_1 \rightarrow_{R,B} t_2$ and $t_1 =_B t_3$ implies $\exists t_4$ such that $t_3 \rightarrow_{R,B} t_4$ and $t_2 =_E t_4$. We also assume $\rightarrow_{R,B}$ is *E-consistent with* $\rightarrow_{\Delta,B}$, i.e. $\forall t_1, t_2, t_3$ we have $t_1 \rightarrow_{R,B} t_2$ and $t_1 \rightarrow^*_{\Delta,B} t_3$ implies $\exists t_4, t_5$ such that $t_3 \rightarrow^*_{\Delta,B} t_4$ and $t_4 \rightarrow_{R,B} t_5$ and $t_5 =_E t_2$.

$$\begin{tikzcd}
t_1 \arrow[r, "\rightarrow_{R,B}"] \arrow[d, dash, "||_B"] & t_2 \arrow[d, dash, "||_E"] \\
t_3 \arrow[r, "\rightarrow_{R,B}"] & t_4
\end{tikzcd}$$

(a) E-consistency of $\rightarrow_{R,B}$ with B

$$\begin{tikzcd}
t_1 \arrow[r, "{R,B}"] \arrow[d, "{*}\,{\Delta,B}"] & t_2 \arrow[d, dash, "||_E"] \\
t_3 \arrow[r, "\rightarrow^*_{\Delta,B}\rightarrow_{R,B}"] & t_4
\end{tikzcd}$$

(b) E-consistency of $\rightarrow_{R,B}$ with $\rightarrow_{\Delta,B}$

The following lemma links $\rightarrow_{R/E}$ with $\rightarrow_{\Delta,B}$ and $\rightarrow_{R,B}$. It was originally established by Patrick Viry for unsorted unconditional rewrite theories [42], but lifts to our order-sorted setting in a straightforward way.

Lemma 3.2 *Let $\mathcal{R} = (\Sigma, \Delta \cup B, R)$ be an order-sorted rewrite theory with all the properties assumed above. Then $t_1 \rightarrow_{R/E} t_2$ if and only if $t_1 \rightarrow^*_{\Delta,B}\rightarrow_{R,B} t_3$ for some $t_3 =_E t_2$.* □

Thus $t_1 \rightarrow^*_{R/E} t_2$ if and only if $t_1 \rightarrow^*_{R\cup\Delta,B} t_3$ for some $t_3 =_E t_2$. The reader can check that this can be lifted to goals as $G_1 \rightarrow^*_{R/E} G_2$ if and only if $G_1 \rightarrow^*_{R\cup\Delta,B} G_3$ for some $G_3 =_E G_2$. *All the assumptions about $\mathcal{R}$ listed in this section, will apply to the rest of the paper, unless explicitly mentioned* *otherwise.*

4 Narrowing: Soundness and Weak Completeness

The R, B-*narrowing* relation on $T_\Sigma(X)$ is defined as follows: $t \overset{\sigma}{\rightsquigarrow}_{R,B} t'$ if there is $\omega \in \mathit{FuPos}(t)$, a rule $l \to r$ in R, where we assume $\mathit{Var}(t) \cap \mathit{Var}(l) = \emptyset$, and $\sigma \in \mathit{CSU}_E(t|_\omega = l, V)$ for a set of variables V containing $\mathit{Var}(t)$ and $\mathit{Var}(l)$, such that $t' = \sigma(t[\omega \leftarrow r])$. This is lifted to reachability goals as follows. Let $G : t_1 \to^* t_2 \wedge \ldots \wedge t_{2n-1} \to^* t_{2n}$ and $G' : t'_1 \to^* t'_2 \wedge \ldots \wedge t'_{2n-1} \to^* t'_{2n}$, and suppose that $\mathit{Var}(G) \subseteq V$. We define $G \overset{\sigma}{\rightsquigarrow}_{R,B} G'$, if there is an *odd* i such that $t_i \overset{\sigma}{\rightsquigarrow}_{R,B} t'_i$ for some σ that is away from $\mathit{Var}(G)$, and for all $j \neq i$ we have $t'_j = \sigma(t_j)$. We write $G \overset{\sigma}{\rightsquigarrow}{}^*_{R,B} G'$ if either $G = G'$ and $\sigma = id$, or there is a sequence of derivations $G \overset{\sigma_1}{\rightsquigarrow}_{R,B} \ldots \overset{\sigma_n}{\rightsquigarrow}_{R,B} G'$ such that $\sigma = \sigma_n \circ \sigma_{n-1} \circ \ldots \circ \sigma_1$. Similarly, Δ, B-narrowing and $R \cup \Delta, B$-narrowing relations are defined on terms and goals, as expected.

Δ, B-narrowing is known to give a sound and complete procedure for $\Delta \cup B$-unification [23]. We show that $R \cup \Delta, B$-narrowing gives a sound but only *weakly* complete (in the sense made precise below) procedure for computing the solutions of reachability goals.

4.1 Soundness

We first consider the soundness problem. Following the idea in [23], we associate with each $R \cup \Delta, B$-narrowing derivation a $R \cup \Delta, B$-rewriting derivation, and then appeal to Lemma 3.2 to complete the argument. First we consider one-step narrowing derivation on terms. The proof of the following lemma is the same as that for the correspondence between Δ, B-narrowing and Δ, B-rewriting, which can be found in [23].

Lemma 4.1 *$t \overset{\sigma}{\rightsquigarrow}_{R\cup\Delta,B} t'$ implies $\sigma(t) \to_{R\cup\Delta,B} t'$.* □

This can be lifted to narrowing derivations on goals as follows.

Lemma 4.2 *$G \overset{\sigma}{\rightsquigarrow}{}^*_{R\cup\Delta,B} G'$ implies $\sigma(G) \to^*_{R\cup\Delta,B} G'$.*

This gives us the following soundness theorem.

Theorem 4.3 (soundness) *Let $G \overset{\sigma}{\rightsquigarrow}{}^*_{R\cup\Delta,B} G'$, and let η be a trivial solution of G', then $\eta \circ \sigma$ is a solution of G.*

4.2 Weak Completeness

The idea behind proving weak completeness is to associate with each $R \cup \Delta, B$-rewriting derivation a $R \cup \Delta, B$-narrowing derivation. It is possible to establish such a correspondence only under certain assumptions, and hence the weakness in completeness. First we consider one-step rewriting on terms.

Lemma 4.4 *Let ρ be an $R \cup \Delta, B$-normalized substitution, and let V be a finite set of variables containing $Var(t)$. Let $\rho(t) \rightarrow_{R\cup\Delta,B} t'$ using the rule $l \rightarrow r$ in R or the equation $l = r$ in Δ. Then there are σ, t'', η such that:*

(i) *$t \overset{\sigma}{\leadsto}_{R\cup\Delta,B} t''$ using the same rule or equation.*

(ii) *η is $R \cup \Delta, B$-normalized*

(iii) *$\eta(t'') =_B t'$, and*

(iv) *$\rho|_V =_B (\eta \circ \sigma)|_V$*

Next, we associate to a one-step R/E-rewrite an $R \cup \Delta, B$-narrowing derivation.

Lemma 4.5 *Let ρ be an $R \cup \Delta, B$-normalized substitution, and let V be a finite set of variables containing $Var(t)$. Then $\rho(t) \rightarrow_{R/E} t'$ implies that there are $\sigma_1, \sigma_2, t'', \eta$ such that:*

(i) *$t \overset{\sigma_1}{\leadsto}^{*}_{\Delta,B} \overset{\sigma_2}{\leadsto}_{R,B} t''$*

(ii) *η is $R \cup \Delta, B$-normalized*

(iii) *$\eta(t'') =_E t'$, and*

(iv) *$\rho|_V =_E (\eta \circ \sigma_2 \circ \sigma_1)|_V$*

The above lemma can be lifted to narrowing derivations on goals as follows.

Lemma 4.6 *Let ρ be an $R \cup \Delta, B$-normalized substitution, V be a finite set of variables containing $Var(G)$, and let $\rho(G) \rightarrow^{*}_{R/E} G'$. Then, there are σ, G'', η such that:*

(i) *$G \overset{\sigma}{\leadsto}^{*}_{R\cup\Delta,B} G''$*

(ii) *η is $R \cup \Delta, B$-normalized*

(iii) *$\eta(G'') =_E G'$.*

(iv) *$\rho|_V =_E (\eta \circ \sigma)|_V$*

We are now ready to prove the weak completeness result.

Theorem 4.7 (weak completeness) *Let ρ be an R/E-normalized solution of a reachability goal G, and let V be a finite set of variables containing $Var(G)$. Then there are σ, G' such that:*

(i) *$G \overset{\sigma}{\leadsto}^{*}_{R\cup\Delta,B} G'$ and G' has a trivial solution.*

(ii) *There is $\eta \in CSU_E(\mathcal{E}(G'), V \cup Ran(\sigma))$ such that $(\eta \circ \sigma)|_V \ll_E \rho|_V$*

We shall see later that Theorem 4.7 need not hold for substitutions ρ that are not R/E-normalized, and hence narrowing is only weakly complete.

4.3 A Weakly Complete Algorithm for Reachability Goals

Theorem 4.8 *For a reachability goal G, let V be a finite set of variables containing $Var(G)$, and let Γ be the set of all substitutions of the form $\eta \circ \sigma$,* *where* $G \overset{\sigma}{\leadsto}^{*}_{R\cup\Delta,B} G'$ *and* $\eta \in CSU_E(\mathcal{E}(G'), V \cup Ran(\sigma))$. *Then* Γ *is a complete set of solutions of* G *with respect to* R/E*-normalized solutions.*

Proof. From Theorems 4.3 and 4.7. □

This theorem provides a general algorithm which builds a narrowing tree starting from G, to find all R/E-normalized solutions. Nodes in this tree correspond to goals, while edges correspond to one-step $R \cup \Delta, B$-narrowing derivations. Since there can be infinitely long narrowing derivations, the algorithm has to expand the tree in a *fair* manner to cover each possible derivation. Further, note that for each node in the tree, the algorithm invokes a $\Delta \cup B$-unification algorithm, which is not required to be finitary, i.e., the unification algorithm can return an infinite set of unifiers. Therefore, the execution of this unification algorithm is to be interleaved in a fair manner with the expansion of the narrowing tree. Finally, we note that it is important to study appropriate *strategies* [3] that, while preserving completeness, make this narrowing procedure as efficient as possible.

4.4 Incompleteness of Narrowing

Narrowing is complete only with respect to R/E-normalized solutions. It is incomplete in general, as shown by the following examples.

Example 4.9 Let $\mathcal{R} = (\Sigma, \emptyset, R)$, where the signature Σ has a single sort, and unary function symbols s, f, g, and R has the following two rules:

$$s(x) \to s^2(x) \qquad\qquad f(s^2(x)) \to g(s(x))$$

The reachability goal $G : f(x) \to^* g(x)$ has solutions $\sigma_k = \{s^k(y)/x\}$ for $k \geq 1$ (none of which is R/E-normalized). But narrowing returns only σ_2 as a solution, and it is not the case that $\sigma_2|_{\{x\}} \ll_{\emptyset} \sigma_1|_{\{x\}}$.

Example 4.10 Consider $\mathcal{R} = (\Sigma, \emptyset, R)$, where Σ has a single sort, and constants a, b, c, d, and a binary function symbol f, and R has the following three rules:

$$a \to b \qquad\quad a \to c \qquad\quad f(b, c) \to d$$

The reachability goal $G : f(x, x) \to^* d$ has $\sigma = \{a/x\}$ as a solution. But G has neither a trivial solution nor a narrowing derivation starting from it.

5 Some Strong Completeness Results

It is possible to obtain strong completeness results for useful classes of rewrite theories. We consider several such classes, including topmost rewrite theories, classes semantically equivalent to topmost rewrite theories, and linear rewrite theories.

5.1 Topmost Rewrite Theories

We say $\mathcal{R} = (\Sigma, E, R)$ is a *topmost* rewrite theory if in one of the equivalences classes of $S/\equiv$, there is a top sort *State* such that:

- Each rule in R rewrites terms of sort *State*, i.e., for each $l \to r$ in R it is the case that $l \in T_\Sigma(X)_{State}$ and $r \in T_\Sigma(X)_{State}$.
- For each $f : [s_1] \times \ldots \times [s_n] \to s$ in Σ, it is the case that $[s_i] \neq State$ for $1 \leq i \leq n$.

These two conditions force every rewrite to happen at the top of a term. More precisely, the relations $\to_{R/E}$ and $\to_{R,E}$ coincide, and if $t \to_{R,E} t'$ then this rewrite happens at the position ϵ in t. Thus, R/E-reducibility is decidable if we have an E-matching algorithm, and therefore the assumptions about the rewrite theory $\mathcal{R}$ in Section 3 can be simplified as follows. We assume (in this subsection only) that $\mathcal{R} = (\Sigma, E, R)$ has the following properties: (i) $\mathcal{R}$ is topmost, (ii) the equations in E do not have variables of sort *State*, and (iii) E has a complete unification algorithm. (In particular, the other assumptions in Section 3 are not necessary.)

We show that R, E-narrowing provides a sound and strongly complete procedure for solving reachability goals in rewrite theories with the properties (i)–(iii) listed above. The argument for soundness is the same as in Section 4. For completeness, we first establish a stronger version of Lemma 4.5, in which the substitution ρ is no longer required to be normalized.

Lemma 5.1 *Let t be a term that is not a variable, and let V be a set of variables containing $Var(t)$. For some substitution ρ, let $\rho(t) \to_{R/E} t'$ using the rule $l \to r$ in R. Then there are σ, η, t'' such that $t \overset{\sigma}{\leadsto}_{R,E} t''$ using the same rule, t'' is not a variable, $\eta(t'') =_E t'$, and $\rho|_V =_E (\eta \circ \sigma)|_V$.*

Using the above Lemma, by an argument similar to that in Section 4, we get the following theorem.

Theorem 5.2 (topmost strong completeness) *Let $G : t_1 \to^* t'_1 \wedge \ldots \wedge t_n \to^* t'_n$ be a reachability goal such that for $1 \leq i \leq n$, t_i is not a variable, and let ρ be a solution of G. Then there are σ, G' such that $G \overset{\sigma}{\leadsto}{}^*_{R,E} G'$ and there is $\eta \in CSU_E(\mathcal{E}(G'), V \cup Ran(\sigma))$ such that $(\eta \circ \sigma)|_V \ll_E \rho|_V$.* □

Thus for a goal G, none of whose sources is a variable, the set of all substitutions $\eta \circ \sigma$ such that $G \overset{\sigma}{\leadsto}_{R,E} G'$ and $\eta \in CSU_E(\mathcal{E}(G'), V \cup Ran(\sigma))$, where V is a finite set of variables containing $Var(G)$, is a *complete* set of solutions of G. As in Section 4, this gives us a general algorithm for computing a complete set of solutions, by building a narrowing tree starting from G. Note that since the E-unification algorithm can return an infinite set of unifiers, the narrowing tree can be infinitely branching. Thus, to ensure completeness, it is essential to expand the narrowing tree in a fair manner.

In practice, we can often transform a given rewrite theory into a topmost rewrite theory which is in some sense equivalent to it, and then exploit the completeness result above. In the following, we consider several classes of theories for which this can be done.

Topmost modulo associativity, commutativity, and identity (ACU):

An order-sorted rewrite theory $\mathcal{R} = (\Sigma, E, R)$ is said to be topmost modulo ACU if in one of the equivalence classes of $S/\equiv$, there is a top sort *Config* such that:

- Each $l \to r$ in R is such that $l, r \in T_\Sigma(X)_{\mathit{Config}}$.
- There is only one operator whose arity includes a sort s such that $[s] = \mathit{Config}$, namely, $\_ \otimes \_ : \mathit{Config} \times \mathit{Config} \to \mathit{Config}$. The operator $\otimes$ is associative and commutative, and has identity *null*.

Many order-sorted rewrite theories specifying object-oriented systems are topmost modulo ACU, in particular, object-oriented systems involving *flat* configurations in which the distributed state is a multiset of objects and messages, are typically topmost modulo ACU. Another large class of examples is provided by different styles of Petri nets [41].

A theory $\mathcal{R}$ that is topmost modulo ACU can be transformed into a corresponding topmost theory $\hat{\mathcal{R}} = (\hat{\Sigma}, E, \hat{R})$ as follows. The signature $\hat{\Sigma}$ extends Σ by adding a new top sort *State*, and a single new operator $\{\_\} : \mathit{Config} \to \mathit{State}$. The set $\hat{R}$ contains for each rewrite rule $l \to r$ in R the rewrite rule $\{l \otimes C\} \to \{r \otimes C\}$, where C is a fresh variable of sort *Config*. This transformation satisfies the following equivalence.

Lemma 5.3 *Let $\mathcal{R}$ be a rewrite theory that is topmost modulo ACU. Then, for any terms t, t' of sort* Config *we have $t \to_{R/E} t'$ if and only if $\{t\} \to_{\hat{R}/E} \{t'\}$.*$\square$

The above lemma implies that the set of $\mathcal{R}$-solutions of $G : t_1 \to^* t'_1 \wedge \ldots \wedge t_n \to^* t'_n$ is the same as the set of $\hat{\mathcal{R}}$-solutions of $\hat{G} : \{t_1\} \to^* \{t'_1\} \wedge \ldots \wedge \{t_n\} \to^* \{t'_n\}$. Thus, to find a complete set of $\mathcal{R}$-solutions of G, we can just find a complete set of $\hat{\mathcal{R}}$-solutions for the goal $\hat{G}$.

Note that the above transformation $\mathcal{R} \mapsto \hat{\mathcal{R}}$ can easily be generalized to operators $\_ \otimes \_$ satisfying the same assumptions, except that $\_ \otimes \_$ satisfies only axioms of associativity and commutativity (AC), or associativity and identity (AU), or associativity alone (A). This makes these results available also for many string-processing rewrite theories, such as grammars. In each of these cases, the transformation $\mathcal{R} \mapsto \hat{\mathcal{R}}$ has to add the appropriate "extension rules". For example, for AC we have to also add the rule $\{l\} \to \{r\}$; for AU we just add $\{C \otimes l \otimes C'\} \to \{C \otimes r \otimes C'\}$; and for A we must also add $\{C \otimes l\} \to \{C \otimes r\}$, $\{l \otimes C\} \to \{r \otimes C'\}$, and $\{l\} \to \{r\}$. With these modifications, the results above also hold for the AC, AU, and A cases as well.

Russian Dolls of Non-increasing Depth:

Many distributed object-based systems are not flat configurations; they are instead *structured* configurations in which multisets of objects and messages can themselves contain nested submultisets encapsulated by appropriate boundary operators. Meseguer and Talcott [32] call such structured configurations *Russian dolls*, to emphasize their nested and recursive character. Since in a system of this kind rewrites can happen at any level of nesting, the results just developed for theories that are topmost modulo ACU do not directly apply. However, under the reasonable assumptions that the equations do *not change* the depth of nesting, and the rewrite rules do *not increase* the depth, it is possible to extend the same idea to Russian dolls, so that narrowing remains a strongly complete analysis method for appropriate goals.

A theory $\mathcal{R} = (\Sigma, E, R)$ of Russian dolls has the following form.

- The signature Σ includes sorts *FlatConfig* and *Config*, with $\mathit{FlatConfig} < \mathit{Config}$, *Config* a top sort in $S/\equiv$, and $s < \mathit{Config}$ implies $s \leq \mathit{FlatConfig}$.
- The only function symbols[3] in Σ whose arity includes a sort s such that $[s] = \mathit{Config}$ are:

  $$\begin{aligned} \_\otimes\_ &: \mathit{FlatConfig} \times \mathit{FlatConfig} \to \mathit{FlatConfig} \\ \_\otimes\_ &: \mathit{Config} \times \mathit{Config} \to \mathit{Config} \\ [\_] &: \mathit{Config} \to \mathit{Config} \end{aligned}$$

  where $\otimes$ is associative and commutative, and has identity *null*. Further, if $f : w \to \mathit{Config}$, then f is either $\_\otimes\_$ or $[\_]$. We say a term t is of *bounded nesting* if for all $x \in \mathit{Var}(t)$, $x \in T_\Sigma(X)_{\mathit{Config}}$ implies $x \in T_\Sigma(X)_{\mathit{FlatConfig}}$.

  For terms of bounded nesting, we define the (nesting) depth of t as follows:

  (i) $\mathit{depth}(t) = 0$ if $t \notin T_\Sigma(X)_{\mathit{Config}}$ or $t \in T_\Sigma(X)_{\mathit{FlatConfig}}$,

  (ii) $\mathit{depth}(t_1 \otimes t_2) = \mathit{max}\{\mathit{depth}(t_1), \mathit{depth}(t_2)\}$,

  (iii) $\mathit{depth}([t]) = \mathit{depth}(t) + 1$.
- For each equation $t = t'$ in E and substitution σ, it is the case that $\sigma(t)$ is of bounded nesting if and only if $\sigma(t')$ is, and further if $\sigma(t)$ and $\sigma(t')$ are of bounded nesting, then $\mathit{depth}(\sigma(t)) = \mathit{depth}(\sigma(t'))$. In short, equations do not change the depth of terms.
- For each rule $l \to r$ in R we have $l, r \in T_\Sigma(X)_{\mathit{Config}}$, and for each substitution σ such that $\sigma(l)$ and $\sigma(r)$ are of bounded nesting, it is the case that $\mathit{depth}(\sigma(l)) \geq \mathit{depth}(\sigma(r))$, i.e. rewrites do not increase the depth of terms.

Note that for a term t of bounded nesting, $\sigma(t)$ is of bounded nesting for any substitution σ, $t =_E t'$ implies t' is of bounded nesting and $\mathit{depth}(t) = \mathit{depth}(t')$, and $t \to_{R/E} t'$ implies t' is of bounded nesting and $\mathit{depth}(t) \geq \mathit{depth}(t')$. The reader is refered to [32] for examples of Russian doll theories.

[3] To simplify the exposition, we assume a very simple operator $[\_] : \mathit{Config} \to \mathit{Config}$ to structure configurations in a nested way as Russian dolls. In general, however, such a structuring operator may have additional sorts as arguments. Our method can be easily extended to those more general structuring operators.

Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ such as above, and a natural number n, we can transform $\mathcal{R}$ into a topmost rewrite theory $\hat{\mathcal{R}}_n = (\hat{\Sigma}_n, E, \hat{R}_n)$ as follows. The signature $\hat{\Sigma}_n$ extends Σ with a new top sort *State* and a new operator $\{\_\} : \mathit{Config} \to \mathit{State}$. The set $\hat{R}_n$ contains for each rule $l \to r$ in R and $0 \leq k \leq n$, the rule

$$\{C_0 \otimes [C_1 \otimes [C_2 \otimes [\ldots [C_k \otimes l] \ldots]]]\} \to \{C_0 \otimes [C_1 \otimes [C_2 \otimes [\ldots [C_k \otimes r] \ldots]]]\}$$

where $C_1, \ldots, C_k$ are fresh variables of sort *Config*.

Lemma 5.4 *Let $\mathcal{R} = (\Sigma, E, R)$ be a Russian doll rewrite theory. Let t be a term of bounded nesting and of sort* Config, *and let $depth(t) = n$, then $t \to_{R/E} t'$ if and only if $\{t\} \to_{\hat{R}_n/E} \{t'\}$.* □

We say a goal $G : t_1 \to^* t_2 \wedge \ldots \wedge t_{2n-1} \to^* t_{2n}$ is of bounded nesting if the t_i are of bounded nesting for all $1 \leq i \leq 2n$. For a goal G of bounded nesting, we define $\mathit{depth}(G) = \mathit{max}\{\mathit{depth}(t_1), \ldots, \mathit{depth}(t_{2n})\}$. The above lemma implies that the set of $\mathcal{R}$-solutions of a goal $G : t_1 \to^* t_2 \wedge \ldots \wedge t_{2n-1} \to^* t_{2n}$ of depth k, is the same as the set of $\hat{\mathcal{R}}_k$-solutions of the goal $\hat{G} : \{t_1\} \to^* \{t_2\} \wedge \ldots \wedge \{t_{2n-1}\} \to^* \{t_{2n}\}$. Thus, to find a complete set of $\mathcal{R}$-solutions of G, we can just find a complete set of $\hat{\mathcal{R}}_k$-solutions of $\hat{G}$.

5.2 Linear Rewrite Theories

In this section we consider linear rewrite theories $\mathcal{R} = (\Sigma, \Delta \cup B, R)$ which, in addition to the assumptions in Section 3, also satisfy the property that B is linear, and each rule in R is sort-decreasing and right linear. We say that a goal $G : t_1 \to^* t'_1 \wedge \ldots \wedge t_n \to^* t'_n$ is *linear* if for all $1 \leq i, j \leq n$ (i) t_i is linear, (ii) $\mathit{Var}(t_i) \cap \mathit{Var}(t_j) = \emptyset$ for $i \neq j$, and (iii) $\mathit{Var}(t_i) \cap \mathit{Var}(t'_j) = \emptyset$. Note that t'_i need not be linear, and it may happen that $\mathit{Var}(t'_i) \cap \mathit{Var}(t'_j) \neq \emptyset$ for some $i \neq j$. We say that a substitution σ is *linear* on a set of variables V if (i) $\sigma(x)$ is linear for all $x \in V$, and (ii) for all $x, y \in V$, we have $\mathit{Var}(\sigma(x)) \cap \mathit{Var}(\sigma(y)) = \emptyset$ for $x \neq y$.

The main reason for incompleteness of narrowing in Section 4 was that, if the rewrite in $\rho(t) \to_{R/E} t'$ happens "within" the substitution ρ, then it is not possible to associate with it a narrowing derivation; this is the reason why we required ρ to be R/E-normalized. But for the case of a linear rewrite theory $\mathcal{R} = (\Sigma, E, R)$ and a linear reachability goal G, we can overcome this limitation to some extent, so that if ρ is an $\mathcal{R}$-solution of G, then narrowing is guaranteed to find another $\mathcal{R}$-solution η such that for some θ we have $\rho|_{\mathit{Var}(G)} \to^*_{R/E} \theta|_{\mathit{Var}(G)}$ and $\eta|_{\mathit{Var}(G)} \ll_E \theta|_{\mathit{Var}(G)}$.

Lemma 5.5 *Let t, t' be terms such that t' is linear and $\mathit{Var}(t) \cap \mathit{Var}(t') = \emptyset$. Let V be a finite set of variables containing $\mathit{Var}(t)$ and $\mathit{Var}(t')$. Let B be a linear and regular set of equations. Then, there is a complete set of B-unifiers of $t = t'$ away from V, namely Γ, such that every $\sigma \in \Gamma$ is linear on $\mathit{Var}(t)$.*

Following are the analogues of Lemmas 4.4, 4.5 and 4.6.

Lemma 5.6 *Given $\mathcal{R} = (\Sigma, \Delta \cup B, R)$, let t be a linear term, and let V be a finite set of variables containing $Var(t)$. Further, for some substitution ρ, let $\rho(t) \rightarrow_{R \cup \Delta, B} t'$ using the rule $l \rightarrow r$ in R or the equation $l = r$ in Δ. Then one of the following is true:*

(i) *$t' = \eta(t)$ for some η such that $\rho|_V \rightarrow_{R\cup\Delta,B} \eta|_V$ using the same rule or equation.*

(ii) *There are σ, η, t'' such that $t \overset{\sigma}{\leadsto}_{R\cup\Delta,B} t''$ using the same rule or equation, t'' is linear, $\eta(t'') =_B t'$, and $\rho|_V =_B (\eta \circ \sigma)|_V$.*

Lemma 5.7 *Let t be a linear term, V be a finite set of variables containing $Var(t)$, and for some substitution ρ, let $\rho(t) \rightarrow_{R/E} t'$, then there is a linear t'', and a substitution η such that $\eta(t'') =_E t'$, and one of the following is true*

(i) *There is σ such that $t \overset{\sigma}{\leadsto}^*_{\Delta,B} t''$ and $\rho|_V \rightarrow^*_{R/E} (\eta \circ \sigma)|_V$*

(ii) *There are σ_1, σ_2 such that $t \overset{\sigma_1}{\leadsto}^*_{\Delta,B} \overset{\sigma_2}{\leadsto}_{R,B} t''$ and $\rho|_V =_E (\eta \circ \sigma_2 \circ \sigma_1)|_V$.*

Lemma 5.8 *Let G be a linear goal, V be a finite set of variables containing $Var(G)$, and for some substitution ρ let $\rho(G) \rightarrow^*_{R/E} G'$, then there are σ, G'', η such that*

(i) *$G \overset{\sigma}{\leadsto}^*_{R\cup\Delta,B} G''$ for some G'' that is linear*

(ii) *$\eta(G'') =_E G'$, and*

(iii) *Either $\rho|_V \rightarrow^*_{R/E} (\eta \circ \sigma)|_V$ or $\rho|_V =_E (\eta \circ \sigma)|_V$*

We are now ready to state the strong completeness result for linear rewrite theories and goals.

Theorem 5.9 (linear strong completeness) *Let G be a linear goal, V be a finite set of variables containing $Var(G)$, and ρ be a solution of G, then there are σ, G' such that:*

- *$G \overset{\sigma}{\leadsto}^*_{R\cup\Delta,B} G'$ and G' has a trivial solution.*
- *There is $\eta \in CSU_E(\mathcal{E}(G'), V \cup Ran(\sigma))$ such that, for some substitution θ, we have $\rho|_V \rightarrow^*_{R/E} \theta|_V$ and $(\eta \circ \sigma)|_V \ll_E \theta|_V$.*

6 Example: Bounded-Process Security Protocol Analysis

Verification of many security protocol properties can be formulated as reachability problems. For instance, verifying the *secrecy* property of a protocol amounts to checking if the protocol can reach a state where an intruder has discovered a data item that was meant to be a secret. In this section, we will exploit the strong completeness result in Section 5.1 to show how narrowing provides a generic and complete procedure for the analysis of such security properties.

In the general case, the reachability problem for security protocols is known to be undecidable [14]. An important decidable subcase is where the number of protocol sessions, i.e., where the number of principals instantiating the protocol roles, is bounded. Even this restricted scenario has an infinite state space, since the intruder can interfere with the protocol execution by forging arbitrary messages. Several authors have proposed decision procedures for the reachability problem in this subcase [20,1,33,38]. An important limitation of all these works is that their analyses do not account for the algebraic properties of the underlying cryptographic primitives. This simplification is not valid for a variety of cryptographic primitives used in practice, such as `xor`, products, and Diffie-Hellman exponentiation. The attacker can exploit algebraic properties of these primitives, such as commutativity, associativity, and cancellation, to find attacks that are otherwise not possible [39].

Recently, extensions to the original decision procedures for the reachability problem, that also account for the algebraic properties of cryptographic primitives, have been proposed [11,34,8,7]. However, these extensions are adhoc and not generic. Specifically, each cryptographic primitive with a different set of algebraic properties has been dealt with by an essentially different extension. We show that narrowing modulo equations provides a generic procedure that can account for a wide class of primitives with algebraic properties. Although narrowing is complete in that it will discover an attack if one exists, it is only a semidecision procedure in that it need not terminate. However, it may be possible to identify several cases where the narrowing procedure for reachability goals is guaranteed to terminate. This is beyond the scope of this paper, and is an important problem for future research.

We now briefly describe how narrowing can be used for security analysis, and illustrate it with a few examples. A protocol can be described as a list of actions, called a *role*, for each honest principal [14]. An action is a pair of terms u, v with variables, which is interpreted as: upon receiving a message matching u, send the corresponding message v. For the sake of concreteness let us consider the case where terms have the following grammar

$$M ::= \mathit{Var} \mid \mathit{Atoms} \mid (M_1, M_2) \mid \{M\}_k$$

where *Atoms* contains the set *Names* of principal names, the set *Keys* of public and private keys of principals, and the set *Nonce* of nonces, (M_1, M_2) is a pair containing M_1 and M_2, and $\{M\}_k$ is the public key encryption of M with key k. We assume functions $pb(\cdot), pv(\cdot) : \mathit{Names} \rightarrow \mathit{Keys}$ which map principal names to the corresponding public and private keys respectively. For a public key k, we denote its private key by k^{-1}. We can consider richer signatures such as those including symmetric key encryption with possibly non-atomic keys and hashing functions, and the discussion below applies to them as well. But we restrict ourselves to this limited signature in the interest of simplicity. Later in this section, to illustrate the fact that narrowing is a general analysis technique that can handle cryptographic primitives with algebraic properties,

$$\text{(Axiom)}\quad K, M \vdash M \qquad\qquad \text{(Pair)}\quad \frac{K \vdash M_1 \qquad K \vdash M_2}{K \vdash (M_1, M_2)}$$

$$\text{(Project)}\quad \frac{K \vdash (M_1, M_2)}{K \vdash M_i}\quad i = 1, 2 \qquad\qquad \text{(Encrypt)}\quad \frac{K \vdash M \qquad K \vdash k}{K \vdash \{M\}_k}$$

$$\text{(Decrypt)}\quad \frac{K \vdash \{M\}_k \qquad K \vdash k^{-1}}{K \vdash M}$$

Table 1
The Dolev-Yao inference rules for intruder capabilities

we will also consider `xor`-encryption.

A protocol instance is a collection of principals, each instantiating a role; we are interested only in finite collections. An intruder can try to compromise the execution of a protocol by replacing an instance of u that was sent by an honest principal with another message that it can build. Typically one assumes that every message exchanged between the honest principals is mediated by the intruder, and the intruder can use the messages that it has observed so far to build fake messages. The most widely used model for the intruder's capability to build messages from the ones it knows, is the *Dolev-Yao* model [13], which is shown in Table 1. The judgment $K \vdash M$ is read as: an attacker that knows all the messages in the set K can construct the message M.

Verifying if the secrecy property is violated amounts to checking if there is a total ordering of actions $(u_1, v_1), \ldots, (u_n, v_n)$ of all the principals, that is consistent with the ordering at each principal, and there is a substitution σ such that

$$K_0, \sigma(v_1), \ldots, \sigma(v_i) \vdash \sigma(u_{i+1}) \qquad \text{and} \qquad K_0, \sigma(v_1), \ldots, \sigma(v_n) \vdash s$$

where K_0 is the initial knowledge of the intruder, and s is the data item that is to be kept secret. K_0, for instance, may contain the name of all the principals, and their public keys. Thus, the protocol is insecure if and only if there is an ordering such that the corresponding set of *constraints* that it generates, have a solution; a solution, if it exists, essentially describes an attack. Note that, since the number of principals is finite, there are only a finite number of total orderings of actions. Such a formalization of the secrecy problem can be found, for instance, in [1,33], to which the reader is referred for further details.

We can represent the constraint system above as a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ that is topmost modulo ACU, and use narrowing to find a complete set of solutions for a given finite set of constraints. The signature Σ has sorts $\mathit{Keys} < \mathit{Atoms} < \mathit{Msg} < \mathit{MsgSet}$, and *Constraint*. The following are constructors for the sort *Msg*.

$$(\_,\_) \ : \ \mathit{Msg} \times \mathit{Msg} \to \mathit{Msg} \qquad\qquad \{\_\}_{\_} \ : \ \mathit{Msg} \times \mathit{Keys} \to \mathit{Msg}$$

The sort *MsgSet* has a single operator $\_,\_ : \mathit{MsgSet} \times \mathit{MsgSet} \to \mathit{MsgSet}$, which is associative and commutative, and has identity *null*. The sort *Constraint* has operators

$$\begin{array}{rcl} \mathit{true} & : & \mathit{Constraint} \\ \_ \vdash \_ & : & \mathit{MsgSet} \times \mathit{Msg} \to \mathit{Constraint} \\ \_ \wedge \_ & : & \mathit{Constraint} \times \mathit{Constraint} \to \mathit{Constraint} \end{array}$$

The operator $\wedge$ is associative, commutative, and has *true* as identity. The rules in R model the inference system of Table 1. An inference rule

$$\frac{K \vdash M_1 \qquad K \vdash M_2}{K \vdash M_3}$$

is modeled as the rewrite rule $K \vdash M_3 \to K \vdash M_1 \wedge K \vdash M_2$ that rewrites multisets of judgments. The idea is that rewriting with these rules, starting from the conclusion, corresponds to searching for a proof of the conclusion in the inference system. To satisfy the condition that $\mathit{Var}(r) \subseteq \mathit{Var}(l)$ for each rule $l \to r$ in R, we consider the following alternate version of the rules (Project) and (Decrypt).

$$\text{(Project')} \ \frac{K, M_1, M_2 \vdash M}{K, (M_1, M_2) \vdash M}$$

$$\text{(Decrypt')} \ \frac{K, \{M_1\}_k \vdash k^{-1} \qquad K, \{M_1\}_k, M_1 \vdash M_2}{K, \{M_1\}_k \vdash M_2}$$

Replacing the rules (Project) and (Decrypt) in Table 1 with the rules above gives us an equivalent inference system, which can be modeled by the following rules.

$$\begin{array}{lrcl}
\text{(Axiom)} & K, M \vdash M & \to & \mathit{true} \\
\text{(Pair)} & K \vdash (M_1, M_2) & \to & K \vdash M_1 \ \wedge \ K \vdash M_2 \\
\text{(Project')} & K, (M_1, M_2) \vdash M & \to & K, M_1, M_2 \vdash M \\
\text{(Encrypt)} & K \vdash \{M\}_k & \to & K \vdash M \ \wedge \ K \vdash k \\
\text{(Decrypt')} & K, \{M_1\}_k \vdash M_2 & \to & K, \{M_1\}_k \vdash k^{-1} \ \wedge \ K, \{M_1\}_k, M_1 \vdash M_2
\end{array}$$

Lemma 6.1 *$K \vdash M$ according to the Dolev-Yao inference rules if and only if $K \vdash M \to^*_{R/E}$ true.* □

From this lemma, it follows that σ is a solution of $K_1 \vdash M_1 \wedge \ldots \wedge K_n \vdash M_n$ if and only if $\sigma(K_i) \vdash \sigma(M_i) \to^*_{R/E}$ *true* for $1 \leq i \leq n$. Now, note that $\mathcal{R}$ is topmost modulo ACU, and hence it can be transformed into a topmost theory, as described in Section 5.1. The resulting topmost theory also satisfies the additional assumptions in Section 5.1, namely, E has a complete unification algorithm, and none of the equations in E have a variable of (the newly introduced) sort *State*. Thus, we can use narrowing to find a complete set of solutions of the goal $K_1 \vdash M_1 \to^*_{R/E} \mathit{true} \wedge \ldots \wedge K_n \vdash M_n \to^*_{R/E} \mathit{true}$.

Example 6.2 Consider the following simplified variant of the Needham-Schroeder public key protocol.

$$\begin{array}{ll} 1.\ A \to B\ :\ \{(N_A, A)\}_{pb(B)} & 3.\ A \to B\ :\ \{N_B\}_{pb(B)} \\ 2.\ B \to A\ :\ \{(N_A, N_B)\}_{pb(A)} & \end{array}$$

A, B denote names of the principals and N_A, N_B denote nonces. In our protocol model, this is represented by two roles *Initiator*(A, B, N_A) and *Responder*(A, B, N_B) as follows.

$$\begin{array}{ll} \mathit{Initiator}(A, B, N_A): & \mathit{Responder}(A, B, N_B): \\ (I_1)\quad \Rightarrow \{(N_A, A)\}_{pb(B)} & (R_1)\ \{(X_1, A)\}_{pb(B)} \Rightarrow \{(X_1, N_B)\}_{pb(A)} \\ (I_2)\ \{(N_A, X_2)\}_{pb(A)} \Rightarrow \{X_2\}_{pb(B)} & (R_2)\ \{N_B\}_{pb(B)} \Rightarrow \end{array}$$

Now, consider an instance with three principals a, b, c, where a plays the role *Initiator*(a, c, n_a) (i.e. intends to initiate the protocol with c), b plays the role *Responder*(a, b, n_b) (i.e., b expects an initiation from a), and c is a dishonest principal (i.e. the intruder). The data item n_b is to be kept secret from the intruder c.

The initial knowledge K_0 of the intruder c includes $a, pb(a), b, pb(b), c, pb(c)$, and $pv(c)$. Consider the following ordering of actions of the honest principals a and b: I_1, R_1, I_2, R_2. This generates the following constraints.

$$\begin{array}{l} K_0, \{(n_a, a)\}_{pb(c)} \vdash \{(X_1, a)\}_{pb(b)} \\ K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1, n_b)\}_{pb(a)} \vdash \{(n_a, X_2)\}_{pb(a)} \\ K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1, n_b)\}_{pb(a)}, \{X_2\}_{pb(c)} \vdash \{n_b\}_{pb(b)} \\ K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1, n_b)\}_{pb(a)}, \{X_2\}_{pb(c)} \vdash n_b \end{array}$$

The narrowing procedure finds the solution $\sigma = \{n_a/X_1, n_b/X_2\}$, which corresponds to the following well-known attack discovered by Lowe [27]:

1. $a \rightarrow c \quad : \{(n_a, a)\}_{pb(c)}$
2. $c(a) \rightarrow b : \{(n_a, a)\}_{pb(b)}$
3. $b \rightarrow c(a) : \{(n_a, n_b)\}_{pb(a)}$
4. $c \rightarrow a : \{(n_a, n_b)\}_{pb(a)}$
5. $a \rightarrow c : \{n_b\}_{pb(c)}$

As mentioned earlier, narrowing modulo equations provides a generic analysis technique that can also handle cases where the underlying cryptographic primitives have algebraic properties that can be exploited by the intruder. We illustrate this with the `xor` encryption primitive. The signature Σ is extended with the following operators

$$0 \ : \ \mathit{Msg} \qquad\qquad \oplus \ : \ \mathit{Msg} \times \mathit{Msg} \rightarrow \mathit{Msg}$$

The constant 0 is the identity for the $\oplus$ operator. Note that in `xor`-encryption, it is possible to use non-atomic keys, i.e., a term of sort *Msg* rather than just a term of sort *Keys*. The set of equations E now also includes the following set of equations XOR for the $\oplus$ operator:

$$\begin{array}{lrcl}
\text{(Assoc)} & (M_1 \oplus M_2) \oplus M_3 & = & M_1 \oplus (M_2 \oplus M_3) \\
\text{(Comm)} & M_1 \oplus M_2 & = & M_2 \oplus M_1 \\
\text{(Ident)} & 0 \oplus M & = & M \\
\text{(Inv)} & M \oplus M & = & 0
\end{array}$$

This equational theory is known to have a complete unification algorithm. The inference system of Table 1 is extended with the following inference rules.

$$\text{(Equality)}\ \frac{K \vdash M_1}{K \vdash M_2} \quad \mathit{if}\ M_1 =_{XOR} M_2 \qquad\qquad \text{(Xor)}\ \frac{K \vdash M_1 \qquad K \vdash M_2}{K \vdash M_1 \oplus M_2}$$

Note that the (Equality) rule captures the intruder's ability to exploit the algebraic properties of `xor`. The set of rules R is extended with the following rule.

$$\text{(Xor)} \quad K \vdash M_1 \oplus M_2 \rightarrow K \vdash M_1 \ \wedge \ K \vdash M_2$$

Since rewrites happen modulo the equations E, the rule (Equality) is implicit. The resulting rewrite theory is again topmost modulo ACU. As before, we can transform it into a topmost theory, and use narrowing to find a complete set of solutions.

Now, consider the following variant of the Needham-Schroeder public key protocol with Lowe's fix [27]. This variant was presented in [8].

1. $A \to B \;:\; \{(N_A, A)\}_{pb(B)}$
2. $B \to A \;:\; \{(N_A \oplus B, N_B)\}_{pb(A)}$
3. $A \to B \;:\; \{N_B\}_{pb(B)}$

In this variant, $\oplus$ is used in step 2, instead of pairing as in Lowe's fix. This is represented in our protocol model as follows.

$FixedInitiator(A, B, N_A)$:

$(I_1) \quad \Rightarrow \{(N_A, A)\}_{pb(B)}$

$(I_2) \; \{(N_A \oplus B, X_2)\}_{pb(A)} \Rightarrow \{X_2\}_{pb(B)}$

$FixedResponder(A, B, N_B)$:

$(R_1) \; \{(X_1, A)\}_{pb(B)} \Rightarrow \{(X_1 \oplus B, N_B)\}_{pb(A)}$

$(R_2) \; \{N_B\}_{pb(B)} \Rightarrow$

Consider the instance with three participants a, b, c as before, with a playing the role $FixedInitiator(a, c, n_a)$, b playing the role $FixedResponder(a, b, n_b)$, and c a dishonest principal. As usual, n_b is to be kept secret from c. The sequence of actions I_1, R_1, I_2, R_2 generates the constraints

$$K_0, \{(n_a, a)\}_{pb(c)} \vdash \{(X_1, a)\}_{pb(b)}$$
$$K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1 \oplus b, n_b)\}_{pb(a)} \vdash \{(n_a \oplus c, X_2)\}_{pb(a)}$$
$$K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1 \oplus b, n_b)\}_{pb(a)}, \{X_2\}_{pb(c)} \vdash \{n_b\}_{pb(b)}$$
$$K_0, \{(n_a, a)\}_{pb(c)}, \{(X_1 \oplus b, n_b)\}_{pb(a)}, \{X_2\}_{pb(c)} \vdash n_b$$

The narrowing procedure finds the solution $\sigma = \{n_a \oplus b \oplus c / X_1, n_b / X_2\}$, which corresponds to the following attack that critically makes use of the equality $n_a \oplus b \oplus c \oplus b = n_a \oplus c$.

1. $a \to c \quad : \{(n_a, a)\}_{pb(c)}$
2. $c(a) \to b : \{(n_a \oplus b \oplus c, a)\}_{pb(b)}$
3. $b \to c(a) : \{(n_a \oplus b \oplus c \oplus b, n_b)\}_{pb(a)}$
4. $c \to a : \{(n_a \oplus b \oplus c \oplus b, n_b)\}_{pb(a)}$
5. $a \to c : \{n_b\}_{pb(c)}$

Finally, we note that other security properties such as authenticity can be analyzed using similar techniques.

7 Concluding Remarks

We have proposed narrowing as a general deductive method to solve reachability problems for a system axiomatized as a rewrite theory. We have proved its soundness and a weak completeness result, have shown that in full generality is incomplete in the strong sense, and have identified important classes of rewrite theories, covering many applications, for which narrowing is indeed strongly complete.

Much more work remains ahead in several directions, including the following:

- Extending the present results to broader classes of rewrite theories.
- Developing narrowing strategies, to be as efficient as possible and to avoid combinatorial explosions; in particular, the use of *constraints* and of how to best combine narrowing with equations (to solve equalities) and with rules should be investigated; also "smart" strategies that can detect looping situations would be very useful [24].
- Building a prototype implementation based on such strategies, that would allow experimentation and supporting unification modulo different equational axioms.
- Investigating termination conditions for the narrowing procedure.
- Studying relationship with other methods that can be used to approximate reachability problems, such as procedures based on tree-automata techniques [18,35].
- Developing applications and case studies, particularly to analyze distributed systems and security protocols.
- Integrating narrowing with other theorem proving methods, for example deductive methods for temporal logic properties [28,6] of rewrite theories.

8 Acknowledgements

The authors would like to thank Jean-Pierre Jouannaud for insightful discussions on narrowing.

References

[1] R. Amadio and D. Lugiez. On the reachability problem in cryptographic primitives. In 11^{th} *International conference on concurrency theory (CONCUR '00)*, volume 1877 of *Lecture Notes in Computer Science*, pages 380–394. Springer, 2000.

[2] David Basin, Sebastian Modersheim, and Luca Vigano. Constraint differentiation: A new reduction technique for constraint-based analysis of security protocols. Technical Report TR-405, Swiss Federal Insititute of Technology, Zurich, May 2003.

[3] Alexander Bockmayr, Stefan Krischer, and Andreas Werner. An optimal narrowing strategy for general canonical systems. In M. Rusinowitch and J.L. Rémy, editors, 3^{rd} *International Workshop on Conditional Term Rewrite systems*, volume 656 of *Lecture Notes in Computer Science*, pages 483–497. Springer, 1992.

[4] Ahmed Bouajjani and Richard Mayr. Model checking lossy vector addition systems. In *STACS*, pages 323–333, 1999.

[5] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification over Infinite States. In *Handbook of Process Algebra*, pages 545–623. Elsevier Publishing, 2001.

[6] K.M. Chandy and J. Misra. *Parallel programming design - A foundation.* Addison Wesley Publishing Company, 1988.

[7] Yannick Chevalier, Ralf Kusters, Michael Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In 23^{rd} *Conference on Foundations Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, 2003. to appear.

[8] Yannick Chevalier, Ralf Kusters, Michael Rusinowitch, and Mathieu Turuani. An NP decisoin procedure for protocol insecurity with XOR. In 18^{th} *Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, 2003.

[9] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* MIT Press, 1999.

[10] Edmund M. Clarke, Orna Grumberg, and David E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, September 1994.

[11] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In 18^{th} *Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 271–280, 2003.

[12] Grit Denker, José Meseguer, and Carolyn L. Talcott. Protocol specification and analysis in Maude. In N. Heintze and J. Wing, editors, *Proceedings of Workshop on Formal Methods and Security Protocols, June 25, 1998, Indianapolis, Indiana*, 1998. http://www.cs.bell-labs.com/who/nch/fmsp/index.html.

[13] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transaction on Information Theory*, 29(2):198–208, 1983.

[14] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Workshop on formal methods and security protocols*, 1999. FLOC.

[15] A. Emerson and K. Namjoshi. On model checking for nondeterministic infinite state systems. In *IEEE Symposium on Logic in Computer Science*, 1998.

[16] M. Fay. First order unification in equational theories. In W. Bibel and R. Kowalski, editors, 4^{th} *Conference on Automated Deduction*, volume 87 of *Lecture Notes in Computer Science*, pages 161–167. Springer, 1979.

[17] Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1):63–92, 2001.

[18] Thomas Genet and Valérie Viet Triem Tong. Reachability analysis of term rewriting systems with Timbuk. In 8^{th} *International Conference on Logic for Programming*, volume 2250 of *Lecture Notes in Computer Science*, 2001.

[19] Susanne Graf and Hassen Saidi. Construction of abstract state graphs with PVS. In Orna Grumberg, editor, *Computer Aided Verification. 9th International Conference, CAV'97, Haifa, Israel, June 22-25, 1997, Proceedings*, volume 1254 of *Lecture Notes in Computer Science*, pages 72–83. Springer-Verlag, 1997.

[20] A. Huima. Efficient infinite state analysis of security protocols. In *Workshop on formal methods and security protocols*, 1999. FLOC.

[21] J.M. Hullot. Canonical forms and unification. In W. Bibel and R. Kowalski, editors, 5^{th} *Conference on Automated Deduction*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334. Springer, 1980.

[22] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Compiling and verifying security protocols. In *Logic Programming and Automated Reasoning*, pages 131–160, 2000.

[23] Jean-Pierre Jouannaud, Claude Kirchner, and Helene Kirchner. Incremental construction of unification algorithms in equational theories. In 10^{th} *International Colloquium on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 361–373. Springer, 1983.

[24] Y. Kaji, T. Fujiwara, and T. Kasami. Solving a unification problem under constrained substitutions using tree automata. *Journal of Symbolic Computation*, 23(1):79–118, 1997.

[25] Yonit Kesten and Amir Pnueli. Control and data abstraction: The cornerstones of practical formal verification. *International Journal on Software Tools for Technology Transfer*, 4(2):328–342, 2000.

[26] C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6:1–36, 1995.

[27] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using fdr. In *Tools and algorithms for construction and analysis of systems (TACAS '96)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer, 1996.

[28] Zohar Manna and Amir Pnueli. Completing the temporal picture. *Theoretical Computer Science*, 83:97–130, 1991.

[29] Catherine Meadows. The NRL protocol analyzer: An overview. *Journal of logic programming*, 26(2):113–131, 1996.

[30] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[31] José Meseguer. Membership algebra as a logical framework for equational specification. In F. Parisi-Presicce, editor, *Proc. WADT'97*, pages 18–61. Springer LNCS 1376, 1998.

[32] José Meseguer and Carolyn Talcott. Semantic models for distributed object reflection. In 16^{th} *European Conference on Object-Oriented Programming*, volume 2374 of *Lecture Notes in Computer Science*, pages 1–36. Springer, 2002.

[33] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In 8^{th} *ACM Conference on Computer and Communications Security (CCS '01)*, pages 166–175, 2001.

[34] J. Millen and V. Shmatikov. Symbolic protocol analysis with products and Diffie-Hellman exponentiation. In 16^{th} *IEEE Computer Security Foundations Workshop (CSFW-16)*, pages 47–61, 2003.

[35] Hitoshi Ohsaki, Hiroyuki Seki, and Toshinori Takai. Recognizing boolean closed A-tree languages with membership conditional mechanism. In 14^{th} *International Conference on Rewriting Techniques and Applications*, volume 2706 of *Lecture notes in computer science*, pages 483–498. Springer Verlag, 2003.

[36] S. Owre, N. Shankar, J. Rushby, and D. Stringer-Calvert. *PVS system guide, PVS language reference, and PVS prover guide version 2.4.* Computer Science Laboratory, SRI International, 2001.

[37] Larry Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.

[38] Michael Rusinowitch and Mathieu Turuani. Protocol insecurity with a finite number of sessions and composed keys is NP-complete. In 14^{th} *IEEE Computer Security Foundations Workshop*, pages 174–190, 2001.

[39] P. Ryan and S. Schneider. An attack on a recursive authentication protocol. *Information Processing Letters*, 65, 1998.

[40] Hassen Saïdi and Natarajan Shankar. Abstract and model check while you prove. In Nicolas Halbwachs and Doron Peled, editors, *Computer Aided Verification. 11th International Conference, CAV'99, Trento, Italy, July 6-10, 1999, Proceedings*, volume 1633 of *Lecture Notes in Computer Science*, pages 443–454. Springer-Verlag, 1999.

[41] Mark-Oliver Stehr, José Meseguer, and Peter Csaba Ölveczky. Rewriting logic as a unifying framework for Petri nets. In H. Ehrig, G. Juhas, J. Padberg, and G. Rozenberg, editors, *Unifying Petri Nets*, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[42] P. Viry. Rewriting: An effective model of concurrency. In C. Halatsis, D. Maritsas, G. Philokyprou, and S. Theodoridis, editors, *PARLE'94 Parallel Architectures and Languages Europe, 6th International PARLE Conference, Athens, Greece, July 4–8, 1994, Proceedings*, volume 817 of *Lecture Notes in Computer Science*, pages 648–660. Springer-Verlag, 1994.

A Appendix

Proof of Lemma 4.2: Lemma 4.1 can be lifted to goals as $G \overset{\eta}{\rightsquigarrow}_{R\cup\Delta,B} G'$ implies $\eta(G) \to_{R\cup\Delta,B} G'$. Then the result follows by a simple induction on the number of narrowing steps in $G \overset{\sigma}{\rightsquigarrow}^*_{R\cup\Delta,B} G'$, using the fact that rewrites are stable under substitution. □

Proof of Theorem 4.3: By Lemma 4.2, we have $\sigma(G) \to^*_{R\cup\Delta,B} G'$, and using Lemma 3.2, we have $\sigma(G) \to^*_{R/E} G'$. Then, since rewrites are stable under substitutions, we have $\eta \circ \sigma(G) \to^*_{R/E} \eta(G')$. Now, since $\eta(G')$ is trivial, from Lemma 3.1 we conclude that $\eta \circ \sigma$ is a solution of G. □

Proof of Lemma 4.4: Without loss of generality we may assume that $\mathit{Dom}(\rho) \subseteq V$, otherwise we can consider $V \cup \mathit{Dom}(\rho)$ instead of V. We may also assume $V \cap \mathit{Var}(l) = \emptyset$. Now, since ρ is $R \cup \Delta, B$-normalized, the rewrite $\rho(t) \to_{R\cup\Delta,B} t'$ occurs at some position $\omega \in \mathit{FuPos}(t)$. Then there is ρ' such that $\mathit{Dom}(\rho') \subseteq \mathit{Var}(l)$, $\rho(t)|_\omega = \rho(t|_\omega) =_B \rho'(l)$, and $t' = \rho(t)[\omega \leftarrow \rho'(r)]$. Let $W = \mathit{Var}(t|_\omega) \cup \mathit{Var}(l)$. Then there is some $\sigma \in \mathit{CSU}_B(t|_\omega = l, V \cup \mathit{Var}(l))$ such that $\sigma|_W \ll_B (\rho \cup \rho')|_W$. Since $\sigma(t|_\omega) =_B \sigma(l)$, and B is regular, we have $\mathit{Var}(\sigma(t|_\omega)) = \mathit{Var}(\sigma(l))$. But since $V \cap \mathit{Var}(l) = \emptyset$, σ is away from $V \cup \mathit{Var}(l)$, and $\mathit{Dom}(\sigma) \subseteq W$, we deduce $\mathit{Dom}(\sigma) = W$. Let η' be such that $(\rho \cup \rho')|_W =_B (\eta' \circ \sigma)|_W$, and $\eta = \eta'|_{\mathit{Ran}(\sigma)} \cup \rho|_V$. Then we have $\rho|_V =_B (\eta \circ \sigma)|_V$, and $\rho'|_{\mathit{Var}(l)} =_B (\eta \circ \sigma)|_{\mathit{Var}(l)}$ (note that $\mathit{Dom}(\sigma) = W \supseteq \mathit{Var}(l)$). Then for $t'' = \sigma(t[\omega \leftarrow r])$, we have $t \overset{\sigma}{\rightsquigarrow}_{R\cup\Delta,B} t''$, and further, since $\mathit{Var}(r) \subseteq \mathit{Var}(l)$, we have $\eta(t'') =_B t'$. Now, we prove by contradiction that η is $R \cup \Delta, B$-normalized. Suppose it is not. Then since $\mathit{Dom}(\eta) \subseteq \mathit{Ran}(\sigma) \cup V$, $\eta|_V = \rho|_V$, and ρ is $R \cup \Delta, B$-normalized it follows that there is $x \in \mathit{Ran}(\sigma)$ such that $\eta(x)$ is not $R \cup \Delta, B$-normalized. Since $\mathit{Var}(\sigma(t|_\omega)) = \mathit{Var}(\sigma(l))$, and $\mathit{Dom}(\sigma) = W$, we have $\mathit{Ran}(\sigma) = \mathit{Ran}(\sigma|_{\mathit{Var}(t|_\omega)})$. Then it follows that there is $x \in V$ such that $\eta \circ \sigma(x)$ is not $R \cup \Delta, B$-normalized. But since $\rho(x) =_B \eta \circ \sigma(x)$, $\to_{\Delta,B}$ is coherent with B, and $\to_{R,B}$ is E-consistent with B, it follows that $\rho(x)$ is not $R \cup \Delta, B$-normalized, a contradiction. □

Proof of Lemma 4.5: By Lemma 3.2, since $\rho(t) \to_{R/E} t'$ we have $\rho(t) \to^*_{\Delta,B}\to_{R,B} s$ for some $s =_E t'$. Now, we exploit the fact that Δ is terminating modulo B and prove the lemma by noetheranian induction on the relation $\to_{\Delta,B} \circ =_B$. For the base case, we have $\rho(t) \to_{R,B} s$, and the result follows by a direct application of Lemma 4.4. For the induction step we have the following diagram.

$$\begin{tikzcd}
\rho(t) & \to_{\Delta,B} & s' & \to^*_{\Delta,B}\to_{R,B} & s \\
\| & & \|_B & & \|_E \\
t \arrow[u, "\rho"'] & \overset{\sigma}{\rightsquigarrow}_{\Delta,B} & s'' \arrow[u, "\eta'"'] & \overset{\sigma'}{\rightsquigarrow}^*_{\Delta,B}\overset{\sigma_2}{\rightsquigarrow}_{R,B} & t'' \arrow[u, "\eta"']
\end{tikzcd}$$

We have $\rho(t) \to_{\Delta,B} s' \to^*_{\Delta,B}\to_{R,B} s$ for some s'. By Lemma 4.4, there are

σ, s'', η' such that $t \overset{\sigma}{\rightsquigarrow}_{\Delta,B} s''$, η' is $R \cup \Delta, B$-normalized, $\eta'(s'') =_B s'$, and $\rho|_V =_B (\eta' \circ \sigma)|_V$. Now, let W be a finite set of variables containing V and $\mathit{Ran}(\sigma)$. Note that since B is regular and the rules in R do not introduce new variables, W contains $\mathit{Var}(s'')$. Now, we have $\eta'(s'') \rightarrow_{R/E} s$. Then by the induction hypothesis, there are $\sigma', \sigma_2, t'', \eta$ such that $s'' \overset{\sigma'}{\rightsquigarrow}^{*}_{\Delta,B} \overset{\sigma_2}{\rightsquigarrow}_{R,B} t''$, η is $R \cup \Delta, B$-normalized, $\eta(t'') =_E s$, and $\eta'|_W =_E (\eta \circ \sigma_2 \circ \sigma')|_W$. Let $\sigma_1 = \sigma' \circ \sigma$. Then we have $\rho|_V =_E (\eta \circ \sigma_2 \circ \sigma_1)|_V$. We have thus proved the result. □

Proof of Lemma 4.6: By induction on the number of derivation steps in $\rho(G) \rightarrow^{*}_{R/E} G'$, using the fact that Lemma 4.5 can be lifted to goals. □

Proof of Theorem 4.7: Since ρ is a solution of G, by Lemma 3.1 we have $\rho(G) \rightarrow^{*}_{R/E} G''$ for some trivial G''. Recall that, since B is sort-preserving and Δ is sort-decreasing, it is the case that $t \rightarrow_{\Delta,B} t'$ and $t \in T_{\Sigma}(X)_s$ implies $t' \in T_{\Sigma}(X)_s$. Therefore, $\rho =_E \rho'$ for some Δ, B-normalized substitution ρ'. Then $\rho'(G) \rightarrow^{*}_{R/E} G''$. Further, since ρ is R/E-normalized it follows that ρ' is $R \cup \Delta, B$-normalized. By Lemma 4.6, there are σ, G', η' such that $G \overset{\sigma}{\rightsquigarrow}^{*}_{R\cup\Delta,B} G'$, $\eta'(G') = G''$, and $\rho'|_V =_E (\eta' \circ \sigma)|_V$. Since G'' is trivial, η' is an E-unifier of $\mathcal{E}(G')$, and hence there is $\eta \in \mathit{CSU}_E(\mathcal{E}(G'), V \cup \mathit{Ran}(\sigma))$ such that $\eta|_{\mathit{Var}(G')} \ll_E \eta'|_{\mathit{Var}(G')}$. Note that since B is regular and the rules in R do not introduce new variables, we have $\mathit{Var}(G') \subseteq V \cup \mathit{Ran}(\sigma)$. Then, by Lemma 2.1, we have $\eta|_{V \cup \mathit{Ran}(\sigma)} \ll_E \eta'|_{V \cup \mathit{Ran}(\sigma)}$. From this and the fact that $\rho'|_V =_E (\eta' \circ \sigma)|_V$ we conclude that $(\eta \circ \sigma)|_V \ll_E \rho'|_V =_E \rho|_V$. □

Proof of Lemma 5.1: Without loss of generality we may assume that $\mathit{Dom}(\rho) \subseteq V$, otherwise we can consider $V \cup \mathit{Dom}(\rho)$ instead of V. We may also assume $V \cap \mathit{Var}(l) = \emptyset$. Now, since $\mathcal{R}$ is topmost and t is not a variable, the rewrite occurs at position $\epsilon \in \mathit{FuPos}(t)$. Then there is ρ' such that $\mathit{Dom}(\rho') \subseteq \mathit{Var}(l)$, $\rho(t) =_E \rho'(l)$, and $t' = \rho'(r)$. Let $W = \mathit{Var}(t) \cup \mathit{Var}(l)$. Then there is some $\sigma \in \mathit{CSU}_E(t = l, V \cup \mathit{Var}(l))$ such that $\sigma|_W \ll_E (\rho \cup \rho')|_W$. Let η' be such that $(\rho \cup \rho')|_W =_E (\eta' \circ \sigma)|_W$, and $\eta = \eta'|_{\mathit{Ran}(\sigma) \cup \mathit{Var}(l)} \cup \rho|_V$. Then we have $\rho|_V =_E (\eta \circ \sigma)|_V$, and $\rho'|_{\mathit{Var}(l)} =_E (\eta \circ \sigma)|_{\mathit{Var}(l)}$. Then for $t'' = \sigma(r)$, we have $t \overset{\sigma}{\rightsquigarrow}_{R,E} t''$, and further, since $\mathit{Var}(r) \subseteq \mathit{Var}(l)$, we have $\eta(t'') =_E t'$. Now, we prove by contradiction that t'' is not a variable. Suppose $t'' = x$ for some variable x. Since $t'' = \sigma(r)$ and r is of sort State, we have that x is of sort State, r is a variable, and σ maps r to x. Since $\mathit{Var}(r) \subseteq \mathit{Var}(l)$ and l does not contain a variable of sort State unless it is itself a variable, it follows that $l = r$. Then, from $\sigma(t) =_E \sigma(l)$, we have that $\sigma(t) =_E x$. But this is impossible, because neither t (and hence $\sigma(t)$) nor any of the equations in E contains a variable of sort State. □

Proof of Lemma 5.5: Consider some Γ' that is a complete set of B-unifiers of $t = t'$ away from V, and let $W = \mathit{Var}(t) \cup \mathit{Var}(t')$. We are done if we show that for each $\sigma' \in \Gamma'$ there is a σ such that σ is a B-unifier of $t = t'$, $\sigma|_W \ll_B \sigma'|_W$, and σ is away from V and is linear on $\mathit{Var}(t)$. Now, let $\sigma' \in \Gamma'$, and let $\rho' = \sigma'|_{\mathit{Var}(t)}$ and $\eta' = \sigma'|_{\mathit{Var}(t')}$. Then, $\rho'(t) =_B \eta'(t')$. Now, we can write $\rho' = (\theta \circ \rho)|_{\mathit{Var}(t)}$ for some θ, ρ such that ρ is linear on $\mathit{Var}(t)$ and away from V, $\mathit{Dom}(\rho) \subseteq \mathit{Var}(t)$, and θ maps variables to variables. Now, since B is linear and regular, t' is linear, and $\mathit{Var}(t) \cap \mathit{Var}(t') = \emptyset$, from $\rho'(t) =_B \eta'(t')$ it follows that $\rho(t) =_B \eta(t')$ for some η such that $\eta' = (\theta \circ \eta)|_{\mathit{Var}(t')}$, η is away from V, and $\mathit{Dom}(\eta) \subseteq \mathit{Var}(t')$. Since $\mathit{Var}(t) \cap \mathit{Var}(t') = \emptyset$, we can take $\sigma = \rho \cup \eta$. Note that σ is a B-unifier of $t = t'$. We have $\sigma'|_W = \rho' \cup \eta' = (\theta \circ \rho)|_{\mathit{Var}(t)} \cup (\theta \circ \eta)|_{\mathit{Var}(t')} = (\theta \circ (\rho \cup \eta))|_W = (\theta \circ \sigma)|_W$, and hence $\sigma|_W \ll_B \sigma'|_W$. Further, since ρ is linear on $\mathit{Var}(t)$ so is σ. Finally, since ρ and η are away from V, so is σ. □

Proof of Lemma 5.6: There are two cases, depending on the position $\omega \in \mathit{Pos}(\rho(t))$ at which the rewrite $\rho(t) \rightarrow_{R \cup \Delta, B} t'$ happens. The first case is when $\omega \notin \mathit{FuPos}(t)$. Then the rewrite happens within the substitution ρ, and since B is sort-preserving, and Δ and R are sort-decreasing, there is a substitution η such that $\rho|_V \rightarrow_{R \cup \Delta, B} \eta|_V$. Further, since t is linear, we have $t' = \eta(t)$. The second case is when $\omega \in \mathit{FuPos}(t)$. Then the proof is the same as that of Lemma 4.4, with the following additional argument for linearity of $t'' = \sigma(t[\omega \leftarrow r])$. Since t is linear, $\mathit{Dom}(\sigma) \subseteq \mathit{Var}(t|_\omega) \cup \mathit{Var}(l)$, and $\mathit{Var}(l) \cap V = \emptyset$, we have $t'' = t[\omega \leftarrow \sigma(r)]$. Now, by Lemma 5.5, we can choose $\mathit{CSU}_B(t|_\omega = l, V \cup \mathit{Var}(l))$ so that σ is linear on $\mathit{Var}(l)$. Since $\mathit{Var}(r) \subseteq \mathit{Var}(l)$, we have that σ is also linear on $\mathit{Var}(r)$. Furthermore, since σ is away from $V \cup \mathit{Var}(l)$, and r is linear, we conclude that t'' is linear. □

Proof of Lemma 5.7: The proof uses Lemma 5.6 and is similar to the proof of Lemma 4.5. The following observations are useful in the proof. For any substitutions $\theta_1, \theta_2, \theta_3, \theta_4$ and sets of variables W, W', we have: (a) $\theta_1|_W \rightarrow_{\Delta, B} \theta_2|_W$ implies $\theta_1|_W =_E \theta_2|_W$, and (b) $\theta_1|_W =_E (\theta_3 \circ \theta_2)|_W$ and $\theta_3|_{W'} \rightarrow^*_{R/E} \theta_4|_{W'}$ for some W' containing $W \cup \mathit{Ran}(\theta_2)$ implies $\theta_1|_W \rightarrow^*_{R/E} (\theta_4 \circ \theta_2)|_W$. □

Proof of Lemma 5.8: By induction on the number of derivation steps in $\rho(G) \rightarrow^*_{R/E} G'$, using the fact that Lemma 5.7 can be lifted to linear goals. □

Proof of Theorem 5.9: The proof uses Lemma 5.8, and is similar to the proof of Theorem 4.7. □