

Complete Symbolic Reachability Analysis Using Back-and-Forth Narrowing

Prasanna Thati¹ and José Meseguer²

¹ Carnegie Mellon University, USA. thati@cs.cmu.edu

² University of Illinois at Urbana-Champaign, USA. meseguer@cs.uiuc.edu

Abstract. We propose a method called *back-and-forth narrowing* for solving reachability goals of the form $(\exists \vec{x}).t_1 \rightarrow^* t'_1 \wedge \dots \wedge t_n \rightarrow^* t'_n$ in general term rewrite systems. The method is a complete semi-decision procedure in the sense that it is guaranteed to find a solution when one exists, but in general it may not terminate when there are no solutions. The completeness result is very general in that it makes no assumptions about the given term rewrite system. Specifically, the rewrite rules need *not* be linear, confluent, or terminating, and can even have extra-variables in the righthand side. Such generality is often essential while modeling concurrent systems or axiomatizing inference systems as rewrite rules, and in such applications back-and-forth narrowing can be used as a sound and complete technique for symbolic reachability analysis or as a deductive procedure for proving existential formulae.

1 Introduction

A concurrent or an inference system can be naturally expressed as a rewrite system $\mathcal{R} = (\Sigma, R)$, where Σ is a signature and R is a collection of rewrite rules. For a concurrent system terms represent states, and a rewrite rule $t \rightarrow t'$ is understood as a (parametric) local transition. For an inference system terms represent formulae, and rewrite rules specify basic inference steps. In this paper, we address the question of solving *reachability goals* in a rewrite system \mathcal{R} . By a reachability goal we mean an existentially quantified formula of the form

$$(\exists \vec{x}) t_1 \rightarrow^* t'_1 \wedge \dots \wedge t_n \rightarrow^* t'_n$$

where each *source* t_i is a term with variables $Var(t_i) \subseteq \vec{x}$ specifying a possibly infinite set of initial configurations (namely all its instances by *ground substitutions*), and each *target* t'_i is a term with variables $Var(t'_i) \subseteq \vec{x}$ that represents likewise a possibly infinite set of configurations that we want to reach by a sequence of transitions starting from the corresponding source t_i . *Solutions* to this reachability problem can then be described by *substitutions* σ for which indeed we have, $\mathcal{R} \vdash \sigma(t_i) \rightarrow^* \sigma(t'_i)$ for $1 \leq i \leq n$. The meaning and interest of solving reachability goals such as the above is clear; it would serve as both a *symbolic reachability analysis* technique for concurrent systems, and, alternatively, as a *deductive procedure* for proving existential formulae in inference systems.

We propose a semi-decision procedure called *back-and-forth-narrowing* for solving reachability goals. This procedure is *complete* in the *solvability* sense in that it is guaranteed to find a solution when there is one. The procedure is very general in the sense that there are absolutely no assumptions on the given rewrite system \mathcal{R} . In particular, the rewrite rules in \mathcal{R} need *not* be left or right linear, or confluent, or terminating, and can also have *extra variables* in the righthand side. This is to be contrasted with other approaches such as model-checking results for special classes of systems [5, 8, 11, 19, 31], or

tree-automata based reachability analysis [12, 24, 29, 4] where typically the rules and the goal are assumed to be linear. In some tree automata approaches [12, 24, 29] non-linearity is dealt with using abstractions or conservative approximations of the reachability set; in contrast, back-and-forth narrowing is an *exact* procedure. A more detailed comparison with related work is presented in Section 7.

Back-and-forth-narrowing is a generalization of *narrowing*, a technique originally introduced as a complete method for generating all solutions of an equational unification problem. Specifically, narrowing was introduced for solving goals of the form $(\exists \vec{x}) t_1 = t'_1 \wedge \dots \wedge t_n = t'_n$ in free algebras modulo a set of confluent and terminating equations used as rewrite rules [16, 18, 23]. Of course, in our new reachability setting, the meaning of a rewrite rule is changed from the previous meaning as an *equality* to a new meaning as a *transition* or *inference*. Further, the completeness of narrowing for equational unification critically depends on the confluence property of equations; an assumption which is done away with in our reachability setting. As a result of these generalizations, a naive extension of narrowing to the reachability setting turns about to be *incomplete* as shown in [22]. It is also shown in [22] that the naive narrowing procedure, however, is complete for certain restricted classes of rewrite theories such as those that are top-most or right-linear. We show, in this paper, that completeness can be regained for arbitrary rewrite systems by using back-and-forth narrowing.

Several applications of solving reachability goals using (naive) narrowing have been reported, especially in the area of verification of computer security protocols [2, 17, 21, 22]. While the approach in [2, 17, 21] is to use narrowing to symbolically search the reachable state space of a protocol, the approach in [22] is to use narrowing to solve appropriate existential formulae in the Dolev-Yao inference system [7] in order to discover attacks if any. These applications exploit the fact mentioned above that naive narrowing is complete for a restricted class of rewrite systems that is sufficient to model the protocols being considered. Our back-and-forth narrowing procedure would substantially expand the scope of these applications to cases where one needs completeness for general rewrite systems.

In the following section, we describe the essential idea behind back-and-forth narrowing at an intuitive level. We follow it up with a more formal treatment in Sections 3 to 6. We discuss related work in Section 7, and conclude in Section 8.

2 The Basic Idea

The essential idea behind using narrowing for solving reachability goals is that a single narrowing sequence starting from a term t can be used to symbolically represent *many* rewrite sequences starting from instances of t . Specifically, for a term t and substitution σ , suppose $\sigma(t) \rightarrow t'$ by rewriting with the rule $l \rightarrow r$ at a non-variable position ω in t . Then clearly, l and $t|_\omega$ (the subterm of t at position ω) are unifiable. If ρ is the most general unifier, then we can show that $\rho(t) \rightarrow t''$ for some t'' by applying $l \rightarrow r$ at position ω , and there is a substitution η such that $t' = \eta(t'')$. This observation motivates the definition of the narrowing step $t \rightsquigarrow t''$; the intention is to use this narrowing step to symbolically represent several rewrite steps, one for each unifier σ of l and $t|_\omega$.

Building on the above idea, one can compose several narrowing steps to get a sequence that symbolically represents many underlying rewrite sequences. One can then hope to use narrowing sequences to search for solutions of reachability goals. Specifically, to solve a given goal $\exists \vec{x}. t_1 \rightarrow^* t_2$ we systematically explore the narrowing tree starting from t_1 ,

and look for a narrowing sequence $t_1 \xrightarrow{\rho_1} \dots \xrightarrow{\rho_n} t'_1$ such that t'_1 and t_2 are unifiable. If η is one such unifier then $\eta \circ \rho_n \circ \dots \circ \rho_1$ is a solution. Unfortunately, although sound, this procedure is *not always complete* even in the solvability sense, i.e., it may fail to find a solution even when one exists. The crucial reason is that, by definition, narrowing can be performed only at non-variable positions, and therefore cannot account for rewrites that occur within the solution (i.e. under variable positions)¹. Such “under-the-feet” rewrites can have non-trivial effects if the rewrite rules or the reachability goal are non-linear, and the rules are not confluent. Consider for example the rewrite rules: (i) $a \rightarrow b$, (ii) $a \rightarrow c$, (iii) $f(b, c) \rightarrow d$, and the reachability goal $\exists x. f(x, x) \rightarrow^* d$. The substitution $\{a/x\}$ is a solution, but the narrowing procedure returns no solutions since $f(x, x)$ can neither be narrowed further nor unified with d .

A natural question to ask is whether the simple narrowing procedure above is complete for specific classes of rewrite systems, or with respect to specific classes of solutions. Indeed, as shown in Section 5, the narrowing procedure above is *weakly* complete in that it can find all R -normalized solutions provided the rewrite rules have no extra variables in the righthand side (see Theorems 2 and 3). However, narrowing may not find solutions that are not normalized. More generally, in [22] we also identified several classes of rewrite systems for which the naive narrowing procedure can find *all* solutions, and applied these results to verify safety properties of cryptographic protocols.

In this paper, we establish a completeness result of a much broader scope by (i) generalizing the basic narrowing step through *linearization* of the term being narrowed, and (ii) using a *combination of forward and backward narrowing* with this generalized relation. Specifically, we account for under-the-feet rewrites by defining a narrowing step that is capable of “skipping” several such rewrites and capturing the first rewrite that occurs at a non-variable position. This is achieved by linearizing a term before narrowing it with a rule. The intermediate under-the-feet rewrites that have thus been skipped will be accounted for by extending the reachability goal with appropriate subgoals. For example, consider the goal $\exists x. f(x, x) \rightarrow^* d$ again. We (i) linearize the term $f(x, x)$ to, say, $f(x_1, x_2)$, (ii) narrow the linearized term with the rule $f(b, c) \rightarrow d$ and the unifier $\{b/x_1, c/x_2\}$, and (iii) extend the reachability goal with subgoals $x \rightarrow^* b$ and $x \rightarrow^* c$. This gives us the goal

$$\exists x. f(b, c) \rightarrow^* d \wedge x \rightarrow^* b \wedge x \rightarrow^* c$$

which can now be solved even using naive narrowing.

Linearization alone does not help us regain completeness in general. For example, consider a goal $\exists \vec{x}. t \rightarrow^* t'$ where the solution σ is such that any rewrite sequence $\sigma(t) \rightarrow^* \sigma(t')$ is such that none of the rewrites occur at non-variable positions of t . But observe that if atleast one of these rewrites occurs at a variable position in t' , then we can narrow the right side t' in the *backward* direction, i.e. using R^{-1} , to obtain a simpler goal. This might in turn enable *forward* narrowing steps using R on the lefthand side, and so on, until we reach a point where all the rewrites occur under variable positions of both the lefthand and righthand sides. In this case, however, the lefthand and righthand sides are unifiable, and we are therefore done.

To keep the presentation simple at this point, we postpone a detailed example illustrating all of these until Section 6 (see Example 3). For the simple example considered

¹ One could of course generalize the definition of narrowing to allow narrowing steps at variable positions. But that would make the narrowing procedure very inefficient since, in general, we will have to perform *arbitrary* instantiations of variables.

above, however, note that just backward narrowing with R^{-1} , even without any linearization, gives us the solution as follows: $d \xrightarrow{id} f(b, c) \xrightarrow{id} f(a, a)$. But as shown in Example 3, a combination of forward and backward narrowing is necessary, in that neither is complete by itself. In Theorems 5 and 6 we prove that with both the generalizations above we regain completeness in the solvability sense for arbitrary rewrite systems.

An important problem for back-and-forth narrowing to be effective in practice is to devise *strategies* that improve its efficiency. Otherwise, one would quickly face a combinatorial explosion in the number of possible narrowing sequences. When several back-and-forth narrowing derivations are possible for the same solution, the question is whether there is a preferred strategy and whether a standardization result is possible. Several *lazy* narrowing strategies that address these questions are known for special classes of rewrite systems [1, 9], but extending these to back-and-forth narrowing is an open question and is beyond the scope of this paper. However, a few comments on a promising approach [10] are made in Section 8.

3 Background

An *signature* Σ is a ranked alphabet $\Sigma = \{\Sigma_n \mid n \in \mathbb{N}\}$, where Σ_n is a set of function symbols of arity n . A Σ -algebra is a set A together with a function $f_A : A^n \rightarrow A$ for each $f \in \Sigma_n$. We assume an infinite set of variables X that are all different from constant symbols in Σ . 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 $Var(t)$, $Pos(t)$, $FuPos(t)$ denote the set of variables, positions, and non-variable (or functional) 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 $\sigma : X \rightarrow T_\Sigma(X)$ which maps variables to terms, and which is different from the identity for only a finite subset $Dom(\sigma)$ of X . We denote the homomorphic extension of σ to $T_\Sigma(X)$ also by σ . The set of variables introduced by σ is $Ran(\sigma) = \cup_{x \in Dom(\sigma)} Var(\sigma(x))$. The restriction of a substitution σ to a set of variables V , is defined as $\sigma|_V(x) = \sigma(x)$ if $x \in V$, and $\sigma|_V(x) = x$ otherwise. We say that a substitution σ is *away* from a set of variables V if $Ran(\sigma) \cap V = \emptyset$. For substitutions σ, ρ such that $Dom(\sigma) \cap Dom(\rho) = \emptyset$ we define their union as

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

For a substitution σ that maps x_i to t_i for $1 \leq i \leq n$, we write $\{t_1/x_1, \dots, t_n/x_n\}$ to denote σ . We denote the identity substitution by *id*.

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

A Σ -*equation* is an expression of the form $t = t'$. A *unifier* for the equation $t = t'$ is a substitution σ such that $\sigma(t) = \sigma(t')$. It is the case that, if t and t' are unifiable, then for any given finite set of variables V containing $W = Var(t) \cup Var(t')$, there is a most general unifier $\sigma = MGU(t = t', V)$ away from V such that (i) $Dom(\sigma) \subseteq W$, and (ii)

$\sigma|_V \ll \rho|_V$ for any other unifier ρ of $t = t'$. This most general unifier σ is unique upto renaming of variables and can be computed by a unification algorithm [28].

A *rewrite rule* is an expression of the form $l \rightarrow r$, where $l, r \in T_\Sigma(X)$. An (*unconditional and unsorted*) *rewrite system* is a tuple $\mathcal{R} = (\Sigma, R)$ with Σ a signature, and R a set of rewrite rules. We write R^{-1} for the set that contains $l \rightarrow r$ if and only if $r \rightarrow l$ is in R . We define the *one-step rewrite relation* on $T_\Sigma(X)$ as follows: $t \rightarrow_R t'$ if there is an $\omega \in Pos(t)$, a rule $l \rightarrow r$ in R , and a substitution σ such that $t|_\omega = \sigma(l)$ and $t' = t[\omega \leftarrow \sigma(r)]$. We also write $t \xrightarrow{[\omega]}_R t'$ to make explicit the position at which the rewrite occurs. Note that $t \rightarrow_R t'$ if and only if $t' \rightarrow_{R^{-1}} t$. A term $t \in T_\Sigma(X)$ is called *R-irreducible* (or just *irreducible* if R is clear from the context) if there is no $t' \in T_\Sigma(X)$ such that $t \rightarrow_R t'$. For substitutions σ, ρ and a set of variables V we define $\sigma|_V \rightarrow_R \rho|_V$ if there is $x \in V$ such that $\sigma(x) \rightarrow_R \rho(x)$ and for all other $y \in V$ we have $\sigma(y) = \rho(y)$. A substitution σ is called *R-normalized* if $\sigma(x)$ is irreducible for all x .

4 Reachability Goals

A *reachability goal* G is a conjunction of the form $t_1 \rightarrow^* t'_1 \wedge \dots \wedge t_n \rightarrow^* t'_n$. It is understood that the order of the subgoals $t_i \rightarrow^* t'_i$ in the expression is irrelevant, i.e., \wedge is associative and commutative. We define $|G| = n$, and $Var(G) = \bigcup_i Var(t_i) \cup Var(t'_i)$. We write G^{-1} to denote the goal $t'_1 \rightarrow^* t_1 \wedge \dots \wedge t'_n \rightarrow^* t_n$. A substitution σ is an *R-solution* of G (or just a solution of G when R is clear from the context) if $\sigma(t_i) \rightarrow_R^* \sigma(t'_i)$ for $1 \leq i \leq n$. Note that since $\sigma(t_i) \rightarrow_R^* \sigma(t'_i)$ if and only if $\sigma(t'_i) \rightarrow_{R^{-1}}^* \sigma(t_i)$, we have that ρ is an *R-solution* of G if and only if ρ is an R^{-1} -solution of G^{-1} . We denote the empty goal, i.e., for the case $n = 0$, by Λ , and define every substitution to be a solution of Λ . We call a goal G of the form $x_1 \rightarrow^* y_1 \wedge \dots \wedge x_n \rightarrow^* y_n$, where all the lefthand sides and the righthand sides are variables, as a *trivial goal*. Note that the substitution σ such that $\sigma(x_i) = \sigma(y_i) = z$ for some variable z , is a solution of this goal. We also define Λ to be a trivial goal.

Definition 1. *We define the rewrite relation on goals as follows.*

$$\begin{aligned} \text{(Reduce)} \quad & G \wedge t_1 \rightarrow^* t_2 \xrightarrow{[\omega]}_R G \wedge t'_1 \rightarrow^* t_2 \quad \text{if } t_1 \xrightarrow{[\omega]}_R t'_1 \\ \text{(Eliminate)} \quad & G \wedge t \rightarrow^* t \xrightarrow{[\epsilon]}_R G. \end{aligned}$$

Note that in $G \xrightarrow{[\omega]}_R G'$, the position ω is not sufficient to determine the exact subgoal at which the rewrite happens. But we adopt this notation because it is sufficient for our purposes and it simplifies the presentation. Further, instead of $G \xrightarrow{[\omega]}_R G'$ we may simply write $G \rightarrow_R G'$.

Lemma 1. *σ is an R-solution of G if and only if $\sigma(G) \rightarrow_R^* \Lambda$.* □

For a set of variables V containing $Var(G)$, we say that a set of substitutions $CSS(G, V)$ is a *complete* set of *R-solutions* of G away from V if: (i) every $\sigma \in CSS(G, V)$ is an *R-solution* of G , (ii) for each solution ρ of G there is a $\sigma \in CSS(G, V)$ such that $\sigma|_{Var(G)} \ll \rho|_{Var(G)}$, and (iii) for every $\sigma \in CSS(G, V)$, $Dom(\sigma) \subseteq Var(G)$ and $Ran(\sigma) \cap V = \emptyset$. We are interested in finding a complete set of *R-solutions* of a goal G in an (unconditional) rewrite system \mathcal{R} .

5 Narrowing: Soundness and Weak Completeness

In this section we show that narrowing provides a sound but only weakly complete procedure (in the sense made precise below) for computing the solutions of reachability goals. We introduced the main ideas in this Section in [22], but here we reformulate their technical presentation in a manner that allows a smooth extension to our more general back-and-forth narrowing procedure in the next section.

The essential idea behind narrowing is to *symbolically* represent the transition relation between terms as a narrowing relation between terms. Specifically, narrowing instantiates the variables in a term by the most general unifier that enables a rewrite with a given rule and a term position. This narrowing relation on terms is then extended to reachability goals by narrowing only the lefthand sides of the goals, while the righthand sides only accumulate substitutions. The idea is to repeatedly narrow the lefthand sides until each lefthand side unifies with the corresponding righthand side. The composition of the unifier with all the substitutions generated (in the reverse order) gives us a solution of the goal.

Definition 2 (narrowing of terms). We define $t \overset{\sigma}{\rightsquigarrow}_R t'$ if there is $\omega \in \text{FuPos}(t)$, a rule $l \rightarrow r$ in R (assume $\text{Var}(t) \cap \text{Var}(l, r) = \emptyset$), such that for a set of variables V containing $\text{Var}(t)$ and $\text{Var}(l, r)$ and $\sigma = \text{MGU}(t|_{\omega} = l, V)$, we have $t' = \sigma(t[\omega \leftarrow r])$.

Definition 3 (narrowing of goals). The narrowing relation on goals is defined by the following two inference rules.

$$\begin{aligned} \text{(Narrow)} \quad G \wedge t \rightarrow^* t' \overset{\sigma}{\rightsquigarrow}_R \sigma(G) \wedge t'' \rightarrow^* \sigma(t') & \quad \text{if } t \overset{\sigma}{\rightsquigarrow}_R t' \text{ and} \\ & \quad \sigma \text{ is away from } \text{Var}(G, t, t') \\ \text{(Unify)} \quad G \wedge t \rightarrow^* t' \overset{\sigma}{\rightsquigarrow}_R \sigma(G) & \quad \text{if } \sigma = \text{MGU}(t = t', \text{Var}(G, t, t')) \end{aligned}$$

We write $G \overset{\sigma}{\rightsquigarrow}_R^* G'$ if either $G = G'$ and $\sigma = \text{id}$, or there is a sequence of derivations $G \overset{\sigma_1}{\rightsquigarrow}_R \dots \overset{\sigma_n}{\rightsquigarrow}_R G'$ such that $\sigma = \sigma_n \circ \sigma_{n-1} \circ \dots \circ \sigma_1$.

Soundness: We first consider the soundness problem. Following the idea in [16] we associate with each narrowing step between terms, a corresponding rewrite step. The proofs of the propositions below are easy.

Lemma 2. $t \overset{\sigma}{\rightsquigarrow}_R t'$ implies $\sigma(t) \rightarrow_R t'$. □

Lemma 3. If $G \overset{\sigma}{\rightsquigarrow}_R G'$ and ρ is a solution of G' , then $\rho \circ \sigma$ is a solution of G . □

This gives us the following soundness theorem.

Theorem 1 (Soundness). If $G \overset{\sigma}{\rightsquigarrow}_R^* \Lambda$, then σ is solution of G . □

Weak Completeness: The idea behind proving weak completeness is to associate with each rewrite step a corresponding narrowing step. It is possible to establish such a correspondence only under certain assumptions, and hence the weakness in the completeness. In the following, note that we assume that each rule $l \rightarrow r$ in R has no extra variables in its righthand side, i.e., $\text{Var}(r) \subseteq \text{Var}(l)$. However, we will *drop* this assumption in the following section where we consider the more general back-and-forth narrowing.

Lemma 4. Let R be a set of rules with no extra variables in their righthand sides. Let ρ be an R -normalized substitution, and let V be a finite set of variables containing $\text{Var}(t)$. Let $\rho(t) \rightarrow_R t'$ using the rule $l \rightarrow r$. Then there are σ, t'', η such that: (i) $t \overset{\sigma}{\rightsquigarrow}_R t''$ using the same rule, σ away from V , (ii) η is R -normalized, (iii) $\eta(t'') = t'$, and (iv) $\rho|_V = (\eta \circ \sigma)|_V$. □

The above lemma can be easily lifted to goals as follows.

Lemma 5. *Let R be a set of rules with no extra variables in their righthand sides. Let ρ be an R -normalized substitution, V be a finite set of variables containing $\text{Var}(G)$, and let $\rho(G) \rightarrow_R G'$. Then, there are σ, G'', η such that: (i) $G \xrightarrow{\sigma}_R G''$, σ away from V , (ii) η is R -normalized, (iii) $\eta(G'') = G'$, and (iv) $\rho|_V = (\eta \circ \sigma)|_V$. \square*

This gives us the following weak completeness result.

Theorem 2 (Weak Completeness). *Let R be a set of rewrite rules with no extra-variables in the righthand side, let ρ be an R -normalized solution of a reachability goal G , and let V be a finite set of variables containing $\text{Var}(G)$. Then $G \xrightarrow{\sigma}^*_R \Lambda$ for some σ away from V such that $\sigma|_V \ll \rho|_V$.*

Proof. By Lemma 1, we have $\rho(G) \rightarrow^*_R \Lambda$. The proof is by induction on the length of this derivation. The base case is obvious. For the induction step, suppose $\rho(G) \rightarrow_R G' \rightarrow^*_R \Lambda$. By Lemma 5, there are σ_1, η, G'' such that $G \xrightarrow{\sigma_1}_R G''$, σ_1 is away from V , η is R -normalized, $\eta(G'') = G'$, and $\rho|_V = (\eta \circ \sigma_1)|_V$. Let $W = V \cup \text{Ran}(\sigma_1)$. Note that $\text{Var}(G'') \subseteq W$. Then by the induction hypothesis there is σ_2 such that $G'' \xrightarrow{\sigma_2}^*_R \Lambda$ for some σ_2 away from W and $\sigma_2|_W \ll \eta|_W$. Then, for $\sigma = \sigma_2 \circ \sigma_1$ we have $G \xrightarrow{\sigma}^*_R \Lambda$, σ is away from V , and $\sigma|_V \ll \rho|_V$. \square

We show below that Theorem 2 need not hold for substitutions ρ that are not R -normalized, and hence narrowing is only weakly complete.

A Weakly Complete Algorithm for Reachability Goals: A simple consequence of Theorems 1 and 2 is the following.

Theorem 3. *Let R be a set of rules with no extra-variables in the righthand side. Then for a finite set of variables V containing $\text{Var}(G)$, the set of all substitutions $\sigma|_{\text{Var}(G)}$ such that $G \xrightarrow{\sigma}^*_R \Lambda$ and σ is away from V , is a complete set of solutions of G away from V , with respect to R -normalized solutions.*

Proof. From Theorems 1 and 2. \square

This theorem provides a general algorithm which builds a narrowing tree starting from G , to find all normalized solutions. Nodes in this tree correspond to goals, while edges correspond to one-step 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.

Incompleteness of Narrowing: Narrowing is complete only with respect to normalized solutions. Specifically, it may not find solutions that are not normalized. We showed an example in the introduction where a reachability goal had a single non-normalized solution, but the narrowing procedure failed to find it. Here is another example.

Example 1. Let $\mathcal{R} = (\Sigma, R)$, where the signature Σ has unary function symbols s, f, g , and R has the following two rules: $s(x) \rightarrow s^2(x)$, and $f(s^2(x)) \rightarrow g(s(x))$. The reachability goal $G = f(x) \rightarrow^* g(x)$ has solutions $\sigma_k = \{s^k(y)/x\}$ for $k \geq 1$ (none of which is R -normalized). But narrowing returns only σ_2 as a solution, and it is not the case that $\sigma_2|_{\{x\}} \ll \sigma_1|_{\{x\}}$.

6 Back-and-Forth Narrowing

The main reason for the incompleteness of narrowing is that, since terms can be narrowed only at non-variable positions (see Definition 2), it is not possible to associate a narrowing step for the rewrite $\rho(t) \xrightarrow{[\omega]}_R t'$ where $\omega \notin FuPos(t)$. Such rewrites “under-the-feet” of t are possible if the substitution ρ is not normalized. This is precisely the reason for the assumption in Theorem 2 that the solution ρ of the goal G is normalized. Fortunately, it is possible to generalize the narrowing relation to one that, in some sense, also accounts for such under-the-feet rewrites.

Suppose ρ is a (not necessarily normalized) solution of the reachability goal $G = G_1 \wedge t_1 \rightarrow^* t_2$. Let

$$\rho(t_1) \xrightarrow{[\omega_1]}_R \dots u \xrightarrow{[\omega_k]}_R v \dots \xrightarrow{[\omega_n]}_R \rho(t_2) \quad (1)$$

and let k be such that $\omega_i \notin FuPos(t_1)$ for $1 \leq i < k$ and $\omega_k \in FuPos(t_1)$. Suppose we *linearize* the term t_1 by renaming each occurrence of a variable $x \in Var(t_1)$ to a distinct variable $x' \notin Var(G)$, and thereby obtain a term \bar{t}_1 . Then, since all the rewrites in $\rho(t_1) \rightarrow^*_R u$ occur under-the-feet of t_1 , i.e., at positions $\omega \notin FuPos(t_1)$, there is a substitution ρ' such that $\rho'(\bar{t}_1) = u$. Specifically, if a variable $x \in Var(t_1)$ is renamed to, say, x_1, \dots, x_n , in \bar{t}_1 , then $\rho(x) \rightarrow^*_R \rho'(x_i)$ for $1 \leq i \leq n$. Now, as in Lemma 4, we can associate to the rewrite step $\rho'(\bar{t}_1) \xrightarrow{[\omega_k]}_R v$ a narrowing step $\bar{t}_1 \xrightarrow{\sigma}_R w$ for some σ and w .

The observation above motivates the definition of an extended narrowing relation on goals that effectively “skips” several under-the-feet rewrites and captures the first rewrite that occurs at a non-variable position in one of the lefthand sides of the goal. Specifically, in the generalized narrowing relation, to solve the goal $G = G_1 \wedge t_1 \rightarrow^* t_2$ above, we (i) linearize the lefthand side t_1 to \bar{t}_1 , (ii) narrow the linearized term \bar{t}_1 as, say, $\bar{t}_1 \xrightarrow{\sigma}_R w$, and (iii) add to the resulting goal a subgoal H that accounts for the intermediate under-the-feet rewrites that have been skipped. Specifically, for each variable $x \in Var(t_1)$ whose occurrences are renamed to, say, $x_1 \dots x_n$, in \bar{t}_1 , the subgoal H contains $x \rightarrow^* \sigma(x_1) \wedge \dots \wedge x \rightarrow^* \sigma(x_n)$. According to this extended narrowing relation, the goal G above narrows to the goal $G' = G_1 \wedge w \rightarrow^* t_2 \wedge H^2$. Since G has a solution ρ as assumed above, it is the case that G' has a solution η such that $\eta|_{Var(G)} = \rho|_{Var(G)}$.

The above discussion applies in particular to the case where in the rewrite sequence (1) above, there is a k such that $\omega_k \in FuPos(t_1)$. Otherwise, there are two possibilities. First, if there is a k such that $\omega_k \in FuPos(t_2)$, then we can apply the above idea in the *backward* direction, i.e., we linearize the righthand side t_2 and narrow the resulting term using R^{-1} . This is justified by the observation that $t \rightarrow^*_R t'$ if and only if $t' \rightarrow^*_{R^{-1}} t$. Thus, we have a procedure that combines forward and backward reachability analysis. Of course, for this idea to work, unlike in Section 5, we have to *allow the rules in R to have extra variables in their righthand sides*. Finally, we are left with the case where $\omega_i \notin FuPos(t_1, t_2)$ for all $1 \leq i \leq n$. We note that in this case, $\rho(t_1)$ and $\rho(t_2)$ should be identical at all positions $\omega \in FuPos(t_1, t_2)$. This observation can be used to further instantiate variables in G , or to reduce G to a trivial goal.

Definition 4 (extended narrowing of terms). *For a term t , let \bar{t} be a linearized form of t , where each occurrence of a variable $x \in Var(t)$ is renamed to a distinct fresh variable*

² Note that the subgoal G_1 is unchanged in the narrowing step. This is because the variables x_1, \dots, x_n that are introduced during linearization of t_1 are fresh w.r.t G_1 , and therefore the substitution σ has no effect on G_1 (see Definitions 4 and 5.)

$y \notin \text{Var}(t)$. Further, suppose $\bar{t} \xrightarrow{\sigma}_R t'$ for σ away from $\text{Var}(t)$. Then we define $t \rightarrow_R t'; H$, where H is the reachability goal such that if the occurrences of a variable $x \in \text{Var}(t)$ are renamed to, say, x_1, \dots, x_n , then H includes the subgoal $x \rightarrow^* \sigma(x_1) \wedge \dots \wedge x \rightarrow^* \sigma(x_n)$.

For example, consider the rewrite system of the example in the Introduction. We have $f(x, x) \rightarrow_R d; (x \rightarrow^* b \wedge x \rightarrow^* c)$, using the rule $f(b, c) \rightarrow d$.

Definition 5 (back-and-forth narrowing of goals). We define a back-and-forth narrowing relation on goals as a decorated relation of the form $G \xrightarrow{\sigma}_R G'$ defined as follows.

- (Narrow-left) $G \wedge t \rightarrow^* t'$
 $\xrightarrow{id}_R G \wedge t'' \rightarrow^* t' \wedge H$ if $t \rightarrow_R t''; H$
- (Narrow-right) $G \wedge t \rightarrow^* t'$
 $\xrightarrow{id}_R G \wedge t \rightarrow^* t'' \wedge H^{-1}$ if $t' \rightarrow_{R^{-1}} t''; H$
- (Decompose) $G \wedge f(t_1, \dots, t_n) \rightarrow^* f(t'_1, \dots, t'_n)$
 $\xrightarrow{id}_R G \wedge t_1 \rightarrow^* t'_1 \wedge \dots \wedge t_n \rightarrow^* t'_n$
- (Match-left) $G \wedge x \rightarrow^* f(t_1, \dots, t_n)$
 $\xrightarrow{\sigma}_R \sigma(G) \wedge x_1 \rightarrow^* \sigma(t_1) \wedge \dots \wedge x_n \rightarrow^* \sigma(t_n)$
if $x_i \notin \text{Var}(G, x, t_1, \dots, t_n)$ for $1 \leq i \leq n$,
and $\sigma = \{f(x_1, \dots, x_n)/x\}$
- (Match-right) $G \wedge f(t_1, \dots, t_n) \rightarrow^* x$
 $\xrightarrow{\sigma}_R \sigma(G) \wedge \sigma(t_1) \rightarrow^* x_1 \wedge \dots \wedge \sigma(t_n) \rightarrow^* x_n$
if $x_i \notin \text{Var}(G, x, t_1, \dots, t_n)$ for $1 \leq i \leq n$,
and $\sigma = \{f(x_1, \dots, x_n)/x\}$
- (Unify) $G \wedge t \rightarrow^* t'$
 $\xrightarrow{\sigma}_R \sigma(G)$ if $\sigma = \text{MGU}(t = t', \text{Var}(G, t, t'))$

For $t \rightarrow_R t''; H$ in the case **Narrow-left** above, we impose the following additional condition. Suppose t is linearized to \bar{t} and $\bar{t} \xrightarrow{\sigma}_R t''$, then we require that the new variables introduced in linearizing t to \bar{t} are fresh with respect to $\text{Var}(G, t, t')$, and the substitution σ is away from $\text{Var}(G, t, t')$. Similar conditions apply to **Narrow-right**. The relation $G \xrightarrow{\sigma}_R^* G'$ is defined by composing the substitutions of each step as expected.

We cannot in general hope for a procedure that, given a goal G enumerates a complete set of solutions of G . For instance, consider the trivial goal $G = x_1 \rightarrow^* y_1 \wedge \dots \wedge x_n \rightarrow^* y_n$. Enumerating a complete set of solutions of G is equivalent to enumerating a set S of tuples $(u_1, v_1, \dots, u_n, v_n)$ such that (i) for each $(u_1, v_1, \dots, u_n, v_n) \in S$ we have $u_i \rightarrow_R^* v_i$, and (ii) for each $s_1, t_1, \dots, s_n, t_n$ such that $s_i \rightarrow_R^* t_i$ there is a $(u_1, v_1, \dots, u_n, v_n) \in S$ and a substitution σ such that $s_i = \sigma(u_i)$ and $t_i = \sigma(v_i)$. We can systematically enumerate one such set S , namely, the set of all tuples $(u_1, v_1, \dots, u_n, v_n)$ such that $u_i \rightarrow_R^* v_i$, but that would be extremely inefficient.

We will therefore give a procedure that is complete only as far as *solvability* of goals is concerned. Specifically, if a given goal G has a solution, then the procedure is guaranteed to find *some* solution of G . For example, for the trivial goal G above, the substitution σ such that $\sigma(x_i) = \sigma(y_i) = z$ will be returned as a solution. In addition, if we have a procedure that enumerates a complete set of solutions for trivial goals, we can combine it with the procedure for solvability to obtain a procedure that enumerates a complete set of solutions for any given goal G (see Theorem 6).

Examples: We now show a few examples where the narrowing procedure of Section 5 fails to find any solution, but back-and-forth narrowing succeeds.

Example 2. Consider the rewrite theory \mathcal{R} and the reachability goal $G = f(x, x) \rightarrow^* d$ of Example 2 in Section 5. We have

$$\begin{array}{ll}
f(x, x) \rightarrow^* d \xrightarrow{id} f(x, x) \rightarrow^* f(b, c) & \text{(Narrow-right)} \\
 \xrightarrow{id} f(x, x) \rightarrow^* f(a, c) & \text{(Narrow-right)} \\
 \xrightarrow{id} f(x, x) \rightarrow^* f(a, a) & \text{(Narrow-right)} \\
 \xrightarrow{\{a/x\}} \Lambda & \text{(Unify)}
\end{array}$$

Thus, back-and-forth narrowing finds the solution $\sigma = \{a/x\}$, whereas the narrowing procedure of Section 5 fails to find any solution. Here is another back-and-forth narrowing derivation that finds the same solution.

$$\begin{array}{ll}
f(x, x) \rightarrow^* d \xrightarrow{id} d \rightarrow^* d \wedge x \rightarrow^* b \wedge x \rightarrow^* c & \text{(Narrow-left)} \\
 \xrightarrow{id} x \rightarrow^* b \wedge x \rightarrow^* c & \text{(Unify)} \\
 \xrightarrow{id} \xrightarrow{id} x \rightarrow^* a \wedge x \rightarrow^* a & \text{(2} \times \text{Narrow-right)} \\
 \xrightarrow{\{a/x\}} \xrightarrow{id} \Lambda & \text{(2} \times \text{Unify)}
\end{array}$$

Example 3. Here is an example that illustrates the use of **Decompose**, and **Match-left**. Consider the rewrite theory $\mathcal{R} = (\Sigma, R)$, where the signature Σ contains the constants a, b, c , a unary function symbol g , and two binary function symbols f, h , and the set R contains the following three rules

$$a \rightarrow h(b, c) \quad b \rightarrow g(a) \quad c \rightarrow h(b, c)$$

Consider the goal $G = f(x, y) \rightarrow^* f(g(y), h(x, y))$. Clearly, there is no narrowing derivation (in the sense of Section 5) starting from $f(x, y)$. But G has the solution $\sigma = \{g(a)/x, h(b, c)/y\}$ because

$$\begin{array}{l}
f(g(a), h(b, c)) \xrightarrow{[1.1]} f(g(h(b, c)), h(b, c)) \xrightarrow{[2.1]} f(g(h(b, c)), h(g(a), c)) \\
 \xrightarrow{[2.2]} f(g(h(b, c)), h(g(a), h(b, c)))
\end{array}$$

Note that all the above rewrites occur under-the-feet of both the lefthand and righthand sides of G . The solution σ is found by back-and-forth narrowing as follows.

$$\begin{array}{ll}
f(x, y) \rightarrow^* f(g(y), h(x, y)) & \\
 \xrightarrow{id} x \rightarrow^* g(y) \wedge y \rightarrow^* h(x, y) & \text{(Decompose)} \\
 \xrightarrow{\sigma_1} x_1 \rightarrow^* y \wedge y \rightarrow^* h(g(x_1), y) & \text{(Match-left)} \\
 \xrightarrow{\sigma_2} x_1 \rightarrow^* h(y_1, y_2) \wedge y_1 \rightarrow^* g(x_1) \wedge y_2 \rightarrow^* h(y_1, y_2) & \text{(Match-left)} \\
 \xrightarrow{id} x_1 \rightarrow^* h(y_1, y_2) \wedge y_1 \rightarrow^* g(x_1) \wedge y_2 \rightarrow^* c \wedge b \rightarrow^* y_1 \wedge c \rightarrow^* y_2 & \text{(Narrow-right)} \\
 \xrightarrow{\sigma_3} x_1 \rightarrow^* h(b, c) \wedge b \rightarrow^* g(x_1) & \text{(3} \times \text{Unify)} \\
 \xrightarrow{id} x_1 \rightarrow^* h(b, c) \wedge g(a) \rightarrow^* g(x_1) & \text{(Narrow-left)} \\
 \xrightarrow{\sigma_4} a \rightarrow^* h(b, c) & \text{(Unify)} \\
 \xrightarrow{id} \Lambda & \text{(Narrow-left, Unify)}
\end{array}$$

where $\sigma_1 = \{g(x_1)/x\}$, $\sigma_2 = \{h(y_1, y_2)/y\}$, $\sigma_3 = \{b/y_1, c/y_2\}$, and $\sigma_4 = \{a/x_1\}$. Thus, back-and-forth narrowing finds the solution $\sigma = (\sigma_4 \circ \sigma_3 \circ \sigma_2 \circ \sigma_1)|_{\{x, y\}}$, while the narrowing procedure of Section 5 doesn't.

Soundness: We now prove the soundness of back-and-forth narrowing of reachability goals. First, following is the analogue of Lemma 2 for the extended narrowing relation on terms.

Lemma 6. *If $t \rightarrow_R t'$; H and ρ is a solution of H , then $\rho(t) \rightarrow_R^* \rho(t')$. \square*

The lemma above is lifted to goals as expected.

Theorem 4. *If $G \xrightarrow{\sigma}_R G'$ and ρ is a solution of G' , then $\rho \circ \sigma$ is a solution of G .*

Proof. For brevity, we consider only the cases **Narrow-right** and **Match-left** of Definition 5.

- Suppose $G = G_1 \wedge t \rightarrow^* t'$, $\sigma = id$ and $G' = G_1 \wedge t \rightarrow^* t' \wedge H^{-1}$, where $t' \rightarrow_{R^{-1}} t''$; H . We are done if we show that $\rho(t) \rightarrow_R^* \rho(t')$. Since ρ is an R -solution of H^{-1} it is also an R^{-1} -solution of H . Then, by Lemma 6, $\rho(t') \rightarrow_{R^{-1}}^* \rho(t'')$, which implies that $\rho(t'') \rightarrow_R^* \rho(t')$. Now, since ρ is a solution of G' , we also have $\rho(t) \rightarrow_R^* \rho(t')$. Putting these observations together, we get $\rho(t) \rightarrow_R^* \rho(t')$.
- Suppose $G = G_1 \wedge x \rightarrow^* f(t_1, \dots, t_n)$, $\sigma = \{f(x_1, \dots, x_n)/x\}$, $G' = \sigma(G_1) \wedge x_1 \rightarrow^* \sigma(t_1) \wedge \dots \wedge x_n \rightarrow^* \sigma(t_n)$. We are done if we show that $\rho \circ \sigma(x) \rightarrow_R^* \rho \circ \sigma(f(t_1, \dots, t_n))$, a sufficient condition for which is $\rho(x_i) \rightarrow_R^* \rho \circ \sigma(t_i)$ for $1 \leq i \leq n$. But this is indeed true, since ρ is a solution of G' . \square

Completeness: Recall that in Section 5 the main idea behind establishing weak completeness of narrowing was to associate to each rewrite step on terms a corresponding narrowing step on terms (Lemma 4). To establish the completeness of back-and-forth narrowing, we generalize this idea to associate to a sequence of rewrites starting from $\rho(t)$, where all but the last rewrite occur at positions $\omega \notin FuPos(t)$, a single extended narrowing step starting from t . This is formalized in the following lemma. It is important to note that, unlike in Lemma 4, rewrite rules are now *allowed* to have extra variables in their righthand side³.

Lemma 7. *Let V be a finite set of variables containing $Var(t)$, and let $\rho(t) \xrightarrow{[\omega_1]}_R \dots \xrightarrow{[\omega_n]}_R \xrightarrow{[\omega]}_R t'$ such that $\omega_i \notin FuPos(t)$ for $1 \leq i \leq n$ and $\omega \in FuPos(t)$. Then there are t'' , H , η such that $t \rightarrow_R t''$; H , η is a solution of H , $\eta|_V = \rho|_V$, and $\eta(t'') = t'$. \square*

Lifting the above lemma to goals is a bit more complicated than its analogue, Lemma 5. Suppose ρ is a solution of G , and π is a rewrite sequence

$$\rho(G) \xrightarrow{[\omega_1]}_R G_1 \xrightarrow{[\omega_2]}_R \dots \xrightarrow{[\omega_n]}_R \Lambda$$

We call π a *witness* for the solution ρ of G . Define the metrics $d(\pi) = \sum_{i=1}^{|\pi|} |\omega_i|$ and $\mu(\pi) = (|\pi|, d(\pi))$. Let \preceq be the usual lexicographic ordering on pairs of natural numbers,

³ The no-extra-variable assumption was necessary in Lemma 4 to guarantee that η is R -normalized. This was in turn required for the assumption that ρ is R -normalized while inductively composing several applications of Lemma 4 to obtain Lemma 5. In contrast, Lemma 7 neither assumes ρ to be R -normalized, nor does it guarantee that η is R -normalized.

i.e., $(m_1, n_1) \preceq (m_2, n_2)$ if $m_1 < m_2$, or $m_1 = m_2$ and $n_1 \leq n_2$. Define $(m_1, n_1) \prec (m_2, n_2)$ if $(m_1, n_1) \preceq (m_2, n_2)$ and $(m_1, n_1) \neq (m_2, n_2)$. Note that \prec is a well-founded relation with $(0, 0)$ as the least element.

Lemma 8. *Let G be a non-trivial reachability goal, V a finite set of variables containing $\text{Var}(G)$, ρ a solution of G , and π a witness for the solution ρ . Then there are σ, η, G' such that $G \xrightarrow{\sigma}_R G'$, σ is away from V , $\rho|_V = (\eta \circ \sigma)|_V$, η is a solution of G' , and there is a witness π' for η such that $\mu(\pi') \prec \mu(\pi)$.*

Proof. Since G is non-trivial, it is of the form $G = G' \wedge t \rightarrow^* t'$, where at least one of t, t' is not a variable. Suppose π involves the rewrites

$$\rho(t) \xrightarrow{[\omega_1]}_R \dots \xrightarrow{[\omega_k]}_R \rho(t')$$

Note that it is possible that $k = 0$, i.e., ρ is a unifier of $t = t'$. By reshuffling the rewrites in π , we can assume that all the rewrites in $\rho(t) \rightarrow^*_R \rho(t')$ occur at the beginning of π , i.e., that π is of the form

$$\rho(G) \rightarrow^*_R \rho(G_1) \wedge \rho(t') \rightarrow^* \rho(t') \rightarrow_R \rho(G_1) \rightarrow^*_R \Lambda$$

Note that such re-shuffling of rewrites in π does not change $\mu(\pi)$. Now, we have the following exhaustive analysis of cases:

- $k = 0$: Then ρ is a unifier of $t = t'$, and for $\sigma = MGU(t = t', V)$ we have $\sigma|_V \ll \rho|_V$. Let η be such that $\rho|_V = (\eta \circ \sigma)|_V$. Then we have $G \xrightarrow{\sigma}_R \sigma(G_1)$ and η is a solution of $\sigma(G_1)$. Further if we take π' to be the rewrite sequence $\rho(G_1) \rightarrow^*_R \Lambda$, we have that π' is a witness for the solution η of $\sigma(G_1)$, $|\pi'| = |\pi| - 1$, and therefore $\mu(\pi') \prec \mu(\pi)$.
- $k > 0$ and there is $1 \leq i \leq k$ such that $\omega_i \in \text{FuPos}(t)$. This case is similar to the next one below, and hence we skip it.
- $k > 0$ and there is $1 \leq i \leq k$ such that $\omega_i \in \text{FuPos}(t')$. Let j be the largest such i , and

$$\rho(t) \xrightarrow{[\omega_1]}_R \dots \xrightarrow{[\omega_{j-1}]}_R u \xrightarrow{[\omega_j]}_R v \xrightarrow{[\omega_{j+1}]}_R \dots \xrightarrow{[\omega_k]}_R \rho(t')$$

Then we have

$$\rho(t') \xrightarrow{[\omega_k]}_{R^{-1}} \dots v \xrightarrow{[\omega_j]}_{R^{-1}} u \xrightarrow{[\omega_{j-1}]}_{R^{-1}} \dots \xrightarrow{[\omega_1]}_{R^{-1}} \rho(t)$$

Then, by Lemma 7, there are u', η, H such that $t' \rightarrow_{R^{-1}} u'$; H, η is an R^{-1} -solution of H , $\eta|_V = \rho|_V$, and $\eta(u') = u$. Then for $\sigma = id$, $G' = G_1 \wedge t \rightarrow^* u' \wedge H^{-1}$, we have $G \xrightarrow{\sigma}_R G'$, and η is a solution of G' . Further, from π we can, in the obvious way, obtain a witness π' for the solution η of G' , and $|\pi'| = |\pi| - 1$, i.e. $\mu(\pi') \prec \mu(\pi)$.

Specifically, π' has a rewrite corresponding to every rewrite in π except $u \xrightarrow{[\omega_j]}_R v$. In particular, for the rewrites $v \rightarrow^*_R \rho(t')$, π' will have corresponding rewrites in H^{-1} .

- $k > 0$ and for all $1 \leq i \leq k$ we have $\omega_i \notin \text{FuPos}(t, t')$. Since at least one of t, t' is not a variable, we have three subcases:

- Both t and t' are not variables. Then it is the case that $t = f(u_1, \dots, u_n)$ and $t' = f(v_1, \dots, v_n)$ for some f , u_i, v_i , and $\rho(u_i) \rightarrow^*_R \rho(v_i)$ for $1 \leq i \leq n$. Then for $\sigma = id$, and $G' = G_1 \wedge u_1 \rightarrow^* v_1 \wedge \dots \wedge u_n \rightarrow^* v_n$, we have $G \xrightarrow{\sigma}_R G'$, and ρ is a solution of G' . Further, from π we can, in the obvious way, derive a witness π' for the solution ρ of G' such that $|\pi'| = |\pi|$ and $d(\pi') < d(\pi)$, i.e., $\mu(\pi') \prec \mu(\pi)$. Now, the statement holds by taking $\eta = \rho$.

- t is a variable, say, x , and $t' = f(v_1, \dots, v_n)$ for some f, v_1, \dots, v_n . Then $\rho(x) = f(u_1, \dots, u_n)$ for some u_1, \dots, u_n and $u_i \rightarrow_R^* \rho(v_i)$ for $1 \leq i \leq n$. Let y_1, \dots, y_n be variables that are fresh with respect to V , $\sigma = \{f(y_1, \dots, y_n)/x\}$, $G' = \sigma(G_1) \wedge y_1 \rightarrow^* \sigma(v_1) \wedge \dots \wedge y_n \rightarrow^* \sigma(v_n)$. Let $\eta = \rho|_V \cup \{u_1/y_1, \dots, u_n/y_n\}$. Then $G \xrightarrow{\sigma}_R G'$, $\rho|_V = (\eta \circ \sigma)|_V$, and η is a solution of G' . Further, from π we can derive a witness π' for the solution η of G' such that $|\pi'| = |\pi|$ and $d(\pi') < d(\pi)$, i.e. $\mu(\pi') \prec \mu(\pi)$.
- $t = f(v_1, \dots, v_n)$ for some f, v_1, \dots, v_n , and t' is a variable. This case is similar to the one above. \square

We are now ready to state the completeness of back-and-forth narrowing.

Theorem 5 (Completeness). *Let ρ be a solution of a reachability goal G , and let V be a finite set of variables containing $\text{Var}(G)$. Then there are σ and G' such that $G \xrightarrow{\sigma}_R^* G'$, σ is away from V , G' is a trivial goal, and there is a solution η of G' such that $\rho|_V = (\eta \circ \sigma)|_V$.*

Proof. The proof is by noetherian induction on $\mu(\pi)$ using Lemma 8, for some witness π of the solution ρ . \square

A Complete Algorithm for Solvability of Reachability Goals:

Theorem 6. *Let V be a finite set of variables containing $\text{Var}(G)$, and let S be the set of all substitutions of the form $(\eta \circ \sigma)|_{\text{Var}(G)}$, where $G \xrightarrow{\sigma}_R^* G'$, σ is away from V , G' is a trivial goal, and $\eta \in \text{CSS}(G', V \cup \text{Ran}(\sigma) \cup \text{Var}(G'))$. Then S is a complete set of solutions of G away from V .*

Proof. From Theorems 4 and 5. \square

Thus, if we are given a procedure for enumerating complete sets of solutions of trivial goals, then we also have a procedure for enumerating complete sets of solutions for any goal. In addition, since for the trivial goal $x_1 \rightarrow^* y_1 \wedge \dots \wedge x_n \rightarrow^* y_n$, the substitution σ such that $\sigma(x_i) = \sigma(y_i) = z$ is a solution, it follows from Theorems 4 and 5 that we have a complete procedure for *solvability* of reachability goals. That is, if a given goal G has a solution, then the procedure finds some solution of G .

7 Related Work

Reachability analysis techniques in general can be broadly classified into two categories. The first is based on constructing a finite approximation of the system under consideration, and then systematically exploring the state space of the approximate model; various abstraction techniques [6, 13, 20] fall in this category. The second approach is based on directly analyzing the infinite state space; examples include decision procedures for model checking special classes of infinite state systems [3, 5, 11, 31], tree automata based techniques [12, 24, 29], and theorem proving [27, 26]. Back-and-forth narrowing falls in this second category.

Backward and forward narrowing may seem familiar in the context of equational unification [16, 18, 23, 25], where a unification goal $\exists \vec{x}. t_1 = t_2$ is transformed into the reachability goal $\exists \vec{x}. eq(t_1, t_2) \rightarrow^* tt$, and is then (naively) narrowed using $R \cup \{eq(t, t) \rightarrow tt\}$. Note that in the transformed goal one can narrow both the lefthand and righthand

sides t_1 and t_2 using R , but both only in the *forward* direction⁴. Further, linearization is not necessary in the equational setting where under-the-feet rewrites are inconsequential due to the confluence assumption. But in a general setting where such assumptions are dropped, linearization becomes essential. In summary, equational unification procedure should *not* be confused with back-and-forth narrowing which is much more general. Specifically, the equational unification procedure just amounts to naive narrowing, and as shown by the examples in Sections 5 and 6, back-and-forth narrowing can solve goals which naive narrowing cannot.

Symbolic reachability analysis using narrowing is also reminiscent of tree-automata (TA) based techniques for reachability analysis. The n^{th} unfolding of the narrowing tree roughly corresponds to the TA recognizing the states that are reachable within n steps. However, there are important differences between the two, which we highlight after briefly recalling the main TA based approaches. In the TA setting, given a rewrite system \mathcal{R} and a regular tree language L , one considers the set $[\rightarrow_{\mathcal{R}}^*]L = \{t \in T_{\Sigma} \mid \exists u \in L \text{ s.t. } u \rightarrow_{\mathcal{R}}^* t\}$. Then, given regular tree languages I and F , the reachability problem is posed as the question of whether the intersection $[\rightarrow_{\mathcal{R}}^*]I \cap F$ is nonempty. In general, $[\rightarrow_{\mathcal{R}}^*]I$ is not a regular tree language and this problem is undecidable. A first approach is to characterize classes of rewrite systems \mathcal{R} for which, given any regular tree language L , the set $[\rightarrow_{\mathcal{R}}^*]L$ is also regular and we can effectively construct a tree automaton recognizing it if we are given a tree automaton recognizing L . Since the set of instances of a nonlinear term is not regular, some linearity assumptions are placed on \mathcal{R} to characterize suitable classes (see [30, 29] for some of the most general classes known so far). A second, more generally applicable approach is to iteratively compute tree automata to recognize $[\rightarrow_{\mathcal{R}}^n]L$ (terms reachable from L in at most n steps). Since $[\rightarrow_{\mathcal{R}}^*]L = \cup_n [\rightarrow_{\mathcal{R}}^n]L$, this yields a semidecision procedure for reachability analysis provided each $[\rightarrow_{\mathcal{R}}^n]L$ is regular; for this again some linearity assumptions on \mathcal{R} are needed, and in some approaches [15] non-linearity is dealt with by over approximations. A third related approach is to compute tree-automata-based *abstractions* that approximate the reachability set [12, 24, 29].

In comparison with back-and-forth narrowing, the main differences have to do with the quite restricted assumptions on term rewriting systems required by TA approaches in order to ensure preservation of the *regularity* of the relevant sets of terms involved in the reachability analysis. By contrast, back-and-forth narrowing is a complete semidecision procedure for *arbitrary* rewrite systems; in particular, regularity-preserving restrictions on a term rewriting system are typically non-symmetric, whereas inverting the rules is part of the back-and-forth narrowing procedure. Under regularity-preserving conditions allowing the use of the first TA approach, the reachability problem is decidable, whereas back-and-forth narrowing is only a semidecision procedure. The third TA approach works by over-approximation, which ensures correctness of negative answers, but can result in false positives; instead, with back-and-forth narrowing a positive solution is always correct and is always found if there is one.

8 Conclusions and Future Work

We have presented back-and-forth narrowing as a semidecision procedure for solving reachability goals in unsorted and unconditional rewrite systems, and we have proved

⁴ The idea behind the transformation is that for a confluent equational theory E , $\sigma(t_1) =_E \sigma(t_2)$ if and only if $\sigma(t_1) \rightarrow_E^* t$ and $\sigma(t_2) \rightarrow_E^* t$ for some term t . The proof of Lemma 8 should shed some light on the fact this idea is totally different from the one behind back-and-forth narrowing.

its completeness in the solvability sense. Although we have given an unsorted treatment using standard rewriting, our method can be extended to general order-sorted rewrite theories of the form (Σ, E, R) with equations E , under appropriate assumptions along the lines adopted in [22]. These assumptions include *pre-regularity* of Σ , that $E = \Delta \cup B$, where the equations Δ are confluent and terminating modulo B , and that Δ and R satisfy certain coherence properties relative to B . Such an extension, that we plan to document in a subsequent paper, will make our results available for many other systems.

Another important direction of research is to investigate efficient strategies for back-and-forth narrowing. Several lazy narrowing strategies are known in the functional-logic programming context [1, 14, 9]. These strategies are all complete for special classes of rewrite systems, typically for left-linear and constructor-based systems. These assumptions are quite reasonable for functional-logic programming applications, but not so in non-equational contexts. In recent work with S. Escobar [10], we have proposed a lazy narrowing strategy called *natural narrowing* for general term rewrite systems that is complete in the weak sense, in that it is guaranteed to find all R -normalized solutions. We conjecture that natural narrowing can be extended to the back-and-forth setting so that completeness is regained even for non-normalized solutions. This problem will be dealt with in subsequent papers.

References

- [1] S. Antoy, R. Echahed, and M. Hanus. A needed narrowing strategy. *Journal of the ACM*, 47(4):776–822, 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 Institute of Technology, Zurich, May 2003.
- [3] Ahmed Bouajjani and Richard Mayr. Model checking lossy vector addition systems. In *STACS*, pages 323–333, 1999.
- [4] Ahmed Bouajjani and Tayssir Touili. Extrapolating tree transformations. In *Proc. 14th Int. Conf. on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, 2002.
- [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] 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.
- [7] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transaction on Information Theory*, 29(2):198–208, 1983.
- [8] A. Emerson and K. Namjoshi. On model checking for nondeterministic infinite state systems. In *IEEE Symposium on Logic in Computer Science*, 1998.
- [9] S. Escobar. Refining weakly outermost-needed rewriting and narrowing. In D. Miller, editor, *Proc. of 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, PPDP'03*, pages 113–123. ACM Press, New York, 2003.
- [10] S. Escobar, J. Meseguer, and P. Thati. Natural narrowing for general term rewriting systems. In *International Conference on Rewriting Techniques and applications (RTA)*, 2005. also available at <http://www.dsic.upc.es/users/elp/papers.html>.
- [11] Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1):63–92, 2001.
- [12] T. Genet and F. Klay. Rewriting for cryptographic protocol verification. In *Automated Deduction—CADE-17*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 271–290. Springer-Verlag, 2000.

- [13] 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.
- [14] M. Hanus. The integration of functions into logic programming: From theory to practice. *Journal of Logic Programming*, 19(20):583–628, 1994.
- [15] Hiroyuki Seki Hitoshi Ohsaki and Toshinori Takai. ACTAS: A system design for associative and commutative tree automata theory. In *Proc. 5th Intl. Workshop on Rule-Based Programming (RULE 2004)*. Elsevier, ENTCS, 2004.
- [16] J.M. Hullot. Canonical forms and unification. In W. Bibel and R. Kowalski, editors, *5th Conference on Automated Deduction*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334. Springer, 1980.
- [17] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Compiling and verifying security protocols. In *Logic Programming and Automated Reasoning*, pages 131–160, 2000.
- [18] Jean-Pierre Jouannaud, Claude Kirchner, and Helene Kirchner. Incremental construction of unification algorithms in equational theories. In *10th International Colloquium on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 361–373. Springer, 1983.
- [19] Y. Kesten, O. Maler, M. Marcus, A. Pnueli, and E. Shahar. Symbolic model checking with rich assertional languages. *Theoretical Computer Science*, 256:93–112, 2001.
- [20] 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.
- [21] Catherine Meadows. The NRL protocol analyzer: An overview. *Journal of logic programming*, 26(2):113–131, 1996.
- [22] José Meseguer and Prasanna Thati. Symbolic reachability analysis using narrowing and its application to analysis of cryptographic protocols. In *Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science. Elsevier, 2004. To appear, also available at <http://osl.cs.uiuc.edu/docs/wr1a04/main.ps>.
- [23] A. Middeldorp and E. Hamoen. Counterexamples to completeness results for basic narrowing. In *Proceedings of the 3rd International Conference on Algebraic and Logic Programming*, Lecture Notes in Computer Science 632, pages 244–258, 1992.
- [24] D. Monniaux. Abstracting cryptographic protocols with tree automata. In *Proc.6th SAS*, pages 149–163. Springer LNCS 1694, 1999.
- [25] S. Okui, A. Middeldorp, and T. Ida. Lazy narrowing: Strong completeness and eager variable elimination. In *Proceedings of the 20th Colloquium on Trees in Algebra and Programming*, Lecture Notes in Computer Science 915, pages 394–408, 1995.
- [26] 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.
- [27] Larry Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.
- [28] G. E. Peterson and M. N. Wegman. Linear unification. *Journal of Computer and Systems Sciences*, 16:158–167, 1978.
- [29] T. Takai. A verification technique using term rewriting systems and abstract interpretation. In *Proc. RTA 2004*, pages 119–133. Springer LNCS 3091, 2004.
- [30] T. Takai, Y. Kaji, and H. Seki. Right-linear finite path overlapping term rewriting systems effectively preserve recognizability. In *Proc. RTA 2000*, pages 246–260. Springer LNCS 1833, 2000.
- [31] P. Wolper and B. Boigelot. Verifying systems with infinite but regular state spaces. In *International Conference on Computer-Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 88–97. Springer Verlag, 1998.

A Appendix

Proof of Lemma 4: Without loss generality we may assume that $Dom(\rho) \subseteq V$, otherwise we can consider $V \cup Dom(\rho)$ instead of V . We may also assume $V \cap Var(l) = \emptyset$. Now, since ρ is R -normalized, the rewrite $\rho(t) \rightarrow_R t'$ occurs at some position $\omega \in FuPos(t)$. Then there is ρ' such that $Dom(\rho') \subseteq Var(l)$, $\rho'(l) = \rho(t)|_\omega = \rho(t|_\omega)$, and $t' = \rho(t)[\omega \leftarrow \rho'(r)]$. Let $W = Var(t|_\omega) \cup Var(l)$, and $\sigma = MGU(t|_\omega = l, V \cup Var(l))$. Then $\sigma|_W \ll (\rho \cup \rho')|_W$. Since $\sigma(t|_\omega) = \sigma(l)$ we have $Var(\sigma(t|_\omega)) = Var(\sigma(l))$. But since $V \cap Var(l) = \emptyset$, σ is away from $V \cup Var(l)$, and $Dom(\sigma) \subseteq W$, we deduce $Dom(\sigma) = W$ and $Ran(\sigma) = Ran(\sigma|_{Var(t|_\omega)})$. Let η' be such that $(\rho \cup \rho')|_W = (\eta' \circ \sigma)|_W$, and $\eta = \eta'|_{Ran(\sigma) \cup \rho|_V}$. Then we have $\rho|_V = (\eta \circ \sigma)|_V$, and $\rho'|_{Var(l)} = (\eta \circ \sigma)|_{Var(l)}$ (note that $Dom(\sigma) = W \supseteq Var(l)$). Then for $t'' = \sigma(t[\omega \leftarrow r])$, we have $t \xrightarrow{\sigma}_R t''$, and further, since $Var(r) \subseteq Var(l)$, we have $\eta(t'') = t'$. Now, we prove by contradiction that η is R -normalized. Suppose it is not. Then since $Dom(\eta) \subseteq Ran(\sigma) \cup V$, $\eta|_V = \rho|_V$, and ρ is R -normalized it follows that there is $x \in Ran(\sigma)$ such that $\eta(x)$ is not R -normalized. Now since $Ran(\sigma) = Ran(\sigma|_{Var(t|_\omega)})$, it follows that there is $y \in V$ such that $\eta \circ \sigma(y)$ is not R -normalized. But since $\rho(y) = \eta \circ \sigma(y)$, we have that $\rho(y)$ is not R -normalized, a contradiction. \square

Proof of Lemma 6: Let \bar{t} be a linearized form of t , $\bar{t} \xrightarrow{\sigma}_R t'$ at position ω , and H be the corresponding reachability goal as constructed in Definition 4. By Lemma 2, we have $\sigma(\bar{t}) \rightarrow_R t'$. Suppose the occurrences of a variable $x \in Var(t)$ are renamed to fresh variables x_1, \dots, x_n . Then since ρ is a solution of H , we have $\rho(x) \rightarrow_R^* \rho \circ \sigma(x_i)$ for $1 \leq i \leq n$. Then it follows that $\rho(t) \rightarrow_R^* \rho \circ \sigma(\bar{t})$. Further, since $\sigma(\bar{t}) \rightarrow_R t'$, we have $\rho \circ \sigma(\bar{t}) \rightarrow_R \rho(t')$. Putting together these observations we get $\rho(t) \rightarrow_R^* \rho(t')$. \square

Lemma 9. *Let R be a set of rules (possibly with extra variables in their righthand sides), and let $\rho(t) \xrightarrow{[\omega]}_R t'$ for some $\omega \in FuPos(t)$ using the rule $l \rightarrow r$. Then for any finite set of variables V containing $Var(t)$, there are σ, t'', η such that: (i) $t \xrightarrow{\sigma}_R t''$ using the same rule, σ away from V , (ii) $\eta(t'') = t'$, and (iii) $\rho|_V = (\eta \circ \sigma)|_V$.*

Proof. A simple modification of the proof of Lemma 4. \square

Proof of Lemma 7: Let \bar{t} be a linearized form of t where the occurrences of each variable $x \in Var(t)$ are renamed to distinct variables that are fresh with respect to V , i.e., $Var(\bar{t}) \cap V = \emptyset$. From the hypothesis of the statement above, it follows that there is a substitution ρ' such that $\rho(t) \xrightarrow{[\omega_1]}_R \dots \xrightarrow{[\omega_n]}_R \rho'(\bar{t}) \xrightarrow{[\omega]}_R t'$ and $\omega \in FuPos(\bar{t})$. In fact, if a variable $x \in Var(t)$ is renamed to x' in \bar{t} , then $\rho(x) \rightarrow_R^* \rho'(x')$. Let the rule used in $\rho'(\bar{t}) \rightarrow_R t'$ be $l \rightarrow r$. We may assume $(V \cup (Var(\bar{t}))) \cap Var(l, r) = \emptyset$. Now, by Lemma 9, there are σ, ξ , and t'' such that $\bar{t} \xrightarrow{\sigma}_R t''$ using the rule $l \rightarrow r$, $\rho'|_{Var(\bar{t})} = (\xi \circ \sigma)|_{Var(\bar{t})}$, and $\xi(t'') = t'$. Without loss of generality we may also assume that σ is away from V . Let H be the reachability goal such that for every variable $x \in Var(t)$ whose occurrences in t are renamed to, say, x_1, \dots, x_n , to obtain \bar{t} , H contains the subgoal $x \rightarrow \sigma(x_1) \wedge \dots \wedge x \rightarrow \sigma(x_n)$. Then $t \rightarrow_R t''; H$. Let $W = Var(\bar{t}) \cup Ran(\sigma) \cup Var(r)$. Then for $\eta = \rho|_V \cup \xi|_W$, we have η is a solution of H , $\eta|_V = \rho|_V$ and $\eta(t'') = t'$. \square