Automata Completion and Regularity Preservation

Thomas Genet

Automata Completion and Regularity Preservation

Thomas Genet

IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France, `genet@irisa.fr`

Abstract

When rewriting a regular language with a left-linear term rewriting system, if the set of reachable terms is regular, we show that equational tree automata completion can compute it. This was known to be true for some known TRS classes preserving regularity, but was still an open question in the general case. The proof is not constructive: it assumes that the set of reachable terms is regular, which is undecidable. Despite being non constructive, the proof of this result has a strong practical impact: it shows how to tune completion to get the best possible precision w.r.t. sets of reachable terms. In particular, to carry out the proof, it was necessary to generalize and improve two results of completion: the termination and the precision theorems.

1998 ACM Subject Classification I.2.3 Deduction and Theorem Proving, F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases term rewriting systems, regularity preservation, tree automata, tree automata completion

1 Introduction

Given a term rewriting system (TRS for short) $\mathcal{R}$ and a tree automaton $\mathcal{A}$ recognizing a regular tree language $\mathcal{L}(\mathcal{A})$, the set of reachable terms (terms reachable by rewriting terms of $\mathcal{L}(\mathcal{A})$ with $\mathcal{R}$) is $\mathcal{R}^*(\mathcal{L}(\mathcal{A})) = \{t \mid s \to_{\mathcal{R}}{}^* t\}$. In this paper, we show that if $\mathcal{R}$ is left-linear (not two occurrences of the same variable in any left-hand side of a rule) and $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular then the *equational tree automata completion algorithm* [15] can build a tree automaton $\mathcal{B}$ recognizing $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. On the one hand, automata built by completion-like algorithms proved to be capable of recognizing *exactly* the set of reachable terms, for some *restricted* classes of TRS (for instance, see [18, 22, 11, 13]). On the other hand, automata completion was able to build *over-approximation* for *any* left-linear TRS [12, 21, 15], and even for non left-linear TRSs [3]. Such approximations are used for program verification [5, 4, 2, 13] and to automate termination proofs [17, 20]. To define approximations, completion uses an additional set of equations E and builds a tree automaton $\mathcal{A}^*_{\mathcal{R},E}$ such that $\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. Knowing if completion was able to compute exactly the set of reachable terms, if it is regular, was still an open question. The main contribution of this paper is to answer positively to this question for left-linear TRS. Since deciding whether $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular is not possible in general, the proof is not constructive. We show that for any left-linear TRS and any tree automaton $\mathcal{A}$, if $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular then *there exists* a set of equations E so that completion of $\mathcal{A}$ with $\mathcal{R}$ and E terminates on a tree automaton $\mathcal{A}^*_{\mathcal{R},E}$ such that $\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E}) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$.

Even if the proof is not constructive, the proof of this result is of practical interest. First, it *formally demonstrates* that the automata completion algorithm has the best possible precision, when computing or over-approximating $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. Second, the proof principle gives some clues on how to build a set of equations w.r.t. a set of reachable terms to attain. Finally, to carry out this proof, we needed to prove two theorems that are of interest in practice:

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(a) if the set of E equivalence classes $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite, then it is possible to build from E a set of equations E', equivalent to E, such that completion of any (reduced) automaton $\mathcal{A}$ by any TRS $\mathcal{R}$ with E' always terminates. This generalizes the termination theorem of [13];

(b) if $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite, then it is possible to build from E and $\mathcal{A}$ a tree automaton $\mathcal{B}$ recognizing the same language as $\mathcal{A}$ and such that the completed automaton $\mathcal{B}^*_{\mathcal{R},E}$ has the following precision property: $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E}) \subseteq \mathcal{R}^*_E(\mathcal{L}(\mathcal{A}))$, where $\mathcal{R}^*_E(\mathcal{L}(\mathcal{A}))$ is the set of reachable terms by rewriting modulo E. This improves the applicability of the upper-bound (or precision) theorem of [15].

Both results use the Myhill-Nerode theorem [19, 7] relating finiteness of $\mathcal{T}(\mathcal{F})/_{=_E}$ and tree automata. In particular, we extend and adapt to our objectives two algorithms of [19]. Then, proving that completion can compute any $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ if it is regular is done as follows. If $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular, then we know that there exists an automaton $\mathcal{A}'$ such as $\mathcal{L}(\mathcal{A}') = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. From $\mathcal{A}'$, using the Myhill-Nerode theorem, we can infer a set of equations E such that equivalence classes of $\mathcal{T}(\mathcal{F})/_{=_E}$ and languages recognized by states of $\mathcal{A}'$ coincide. Using (a) we obtain a set of equations E', equivalent to E, such that completion of any reduced tree automaton, by any TRS and E' terminates. Using (b), we can build $\mathcal{B}$ from $\mathcal{A}$ and E'. Since we know that completion with E' terminates, we know that $\mathcal{B}^*_{\mathcal{R},E'}$ exists. Then, (b) entails that $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E'}) \subseteq \mathcal{R}^*_{E'}(\mathcal{L}(\mathcal{A}))$. Since equivalence classes of E' and languages of $\mathcal{L}(\mathcal{A}')$ coincide, $R^*_{E'}(\mathcal{L}(\mathcal{A})) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. Thus, $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E'}) \subseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. Besides, the lower-bound Theorem of [15] implies that $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E'}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ which concludes the proof.

Section 2 defines some basic notions in term rewriting and tree automata and Section 3 recalls the tree automata algorithm and the related theorems. Section 4 recalls the Myhill-Nerode theorem for trees and defines the functions to transform a set of equations into a tree automaton and vice versa. Section 5 proves the result (a). Section 6 shows the result (b). Finally, Section 7 assembles those results into the proof sketched above.

2 Preliminaries

In this section, we introduce some definitions and concepts that will be used throughout the rest of the paper (see also [1, 7]). Let $\mathcal{F}$ be a finite set of symbols, each associated with an arity function. For brevity, we note $f : n$ if f is a symbol of arity n, and $\mathcal{F}^n = \{f \in \mathcal{F} \mid f : n\}$. Let $\mathcal{X}$ be a countable set of *variables*, $\mathcal{T}(\mathcal{F}, \mathcal{X})$ denotes the set of *terms* and $\mathcal{T}(\mathcal{F})$ denotes the set of *ground terms* (terms without variables). The set of variables of a term t is denoted by $\mathcal{V}ar(t)$. A *substitution* is a function σ from $\mathcal{X}$ into $\mathcal{T}(\mathcal{F}, \mathcal{X})$, which can be uniquely extended to an endomorphism of $\mathcal{T}(\mathcal{F}, \mathcal{X})$. A *position* p for a term t is a finite word over $\mathbb{N}$. The empty sequence λ denotes the top-most position. The set $\mathcal{P}os(t)$ of positions of a term t is inductively defined by $\mathcal{P}os(t) = \{\lambda\}$ if $t \in \mathcal{X}$ or t is a constant and $\mathcal{P}os(f(t_1, \ldots, t_n)) = \{\lambda\} \cup \{i.p \mid 1 \leq i \leq n \text{ and } p \in \mathcal{P}os(t_i)\}$ otherwise. If $p \in \mathcal{P}os(t)$, then $t(p)$ denotes the symbol at position p in t, $t|_p$ denotes the subterm of t at position p, and $t[s]_p$ denotes the term obtained by replacing the subterm $t|_p$ at position p by the term s. A ground context $C[\,]$ is a term of $\mathcal{T}(\mathcal{F} \cup \{\Box\})$ containing exactly one occurrence of the symbol $\Box$. If $t \in \mathcal{T}(\mathcal{F})$ then $C[t]$ denotes the term obtained by the replacement of $\Box$ by t in $C[\,]$. A context is empty if it is equal to $\Box$. If $C[\,]$ is a context, then $C^1[\,] = C[\,]$ and for $n > 1$, $C^n[\,] = C[C^{n-1}[\,]]$.

A *term rewriting system* (TRS) $\mathcal{R}$ is a set of *rewrite rules* $l \to r$, where $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$, and $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$. A rewrite rule $l \to r$ is *left-linear* if each variable occurs only once in l. A TRS $\mathcal{R}$ is left-linear if every rewrite rule $l \to r$ of $\mathcal{R}$ is left-linear. The TRS

$\mathcal{R}$ induces a rewriting relation $\rightarrow_{\mathcal{R}}$ on terms as follows. Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $l \rightarrow r \in \mathcal{R}$, $s \rightarrow_{\mathcal{R}} t$ denotes that there exists a position $p \in \mathcal{P}os(s)$ and a substitution σ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$. The set of term irreducible by a TRS $\mathcal{R}$ is $\textsc{Irr}(\mathcal{R})$. The reflexive transitive closure of $\rightarrow_{\mathcal{R}}$ is denoted by $\rightarrow^*_{\mathcal{R}}$, and $s \rightarrow^!_{\mathcal{R}} t$ denotes that $s \rightarrow^*_{\mathcal{R}} t$ and t is irreducible by $\mathcal{R}$. The set of $\mathcal{R}$-descendants of a set of ground terms I is $\mathcal{R}^*(I) = \{t \in \mathcal{T}(\mathcal{F}) \mid \exists s \in I \text{ s.t. } s \rightarrow^*_{\mathcal{R}} t\}$.

An *equation set* E is a set of equations $l = r$, where $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. The relation $=_E$ is the smallest congruence such that for all equations $l = r$ of E and for all substitutions σ we have $l\sigma =_E r\sigma$. The set of equivalence classes defined by $=_E$ on $\mathcal{T}(\mathcal{F})$ is noted $\mathcal{T}(\mathcal{F})/_{=_E}$. Given a TRS $\mathcal{R}$ and a set of equations E, a term $s \in \mathcal{T}(\mathcal{F})$ is rewritten modulo E into $t \in \mathcal{T}(\mathcal{F})$, denoted $s \rightarrow_{\mathcal{R}/E} t$, if there exist $s' \in \mathcal{T}(\mathcal{F})$ and $t' \in \mathcal{T}(\mathcal{F})$ such that $s =_E s' \rightarrow_{\mathcal{R}} t' =_E t$. The reflexive transitive closure $\rightarrow^*_{\mathcal{R}/E}$ of $\rightarrow_{\mathcal{R}/E}$ is defined as usual except that reflexivity is extended to terms equal modulo E, i.e. for all $s, t \in \mathcal{T}(\mathcal{F})$ if $s =_E t$ then $s \rightarrow^*_{\mathcal{R}/E} t$. The set of $\mathcal{R}$-descendants modulo E of a set of ground terms I is $\mathcal{R}^*_E(I) = \{t \in \mathcal{T}(\mathcal{F}) \mid \exists s \in I \text{ s.t. } s \rightarrow^*_{\mathcal{R}/E} t\}$.

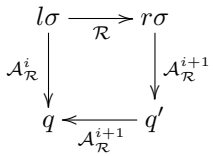
Let $\mathcal{Q}$ be a countably infinite set of symbols with arity 0, called *states*, such that $\mathcal{Q} \cap \mathcal{F} = \emptyset$. Terms of $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ are called *configurations*. A *transition* is a rewrite rule $c \rightarrow q$, where c is a configuration and q is state. A transition is *normalized* when $c = f(q_1, \ldots, q_n)$, $f \in \mathcal{F}$ is of arity n, and $q_1, \ldots, q_n \in \mathcal{Q}$. An *$\epsilon$-transition* is a transition of the form $q \rightarrow q'$ where q and q' are states. A bottom-up non deterministic finite tree automaton (tree automaton for short) over the alphabet $\mathcal{F}$ is a tuple $\mathcal{A} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$, where $\mathcal{Q}_f$ is a finite subset of $\mathcal{Q}$, Δ is a finite set of normalized transitions and ϵ-transitions. An automaton is *epsilon-free* if it is free of ϵ-transitions. The transitive and reflexive *rewriting relation* on $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ induced by the set of transitions Δ (resp. all transitions except ϵ-transitions) is denoted by $\rightarrow^*_{\Delta}$ (resp. $\rightarrow^{\not\epsilon *}_{\Delta}$). When Δ is attached to a tree automaton $\mathcal{A}$ we also note those two relations $\rightarrow_{\mathcal{A}}{}^*$ and $\rightarrow^{\not\epsilon *}_{\mathcal{A}}$, respectively. A tree automaton $\mathcal{A}$ is complete if for all $s \in \mathcal{T}(\mathcal{F})$ there exists a state q of $\mathcal{A}$ such that $s \rightarrow_{\mathcal{A}}{}^* q$. The language recognized by $\mathcal{A}$ in a state q is $\mathcal{L}(\mathcal{A}, q) = \{t \in \mathcal{T}(\mathcal{F}) \mid t \rightarrow^*_{\mathcal{A}} q\}$. We define $\mathcal{L}(\mathcal{A}) = \bigcup_{q \in \mathcal{Q}_f} \mathcal{L}(\mathcal{A}, q)$. A state q of an automaton $\mathcal{A}$ is *reachable* if $\mathcal{L}(\mathcal{A}, q) \neq \emptyset$. An automaton is *reduced* if all its states are reachable. An automaton $\mathcal{A}$ is *$\not\epsilon$-reduced* if for all state q of $\mathcal{A}$ there exists a ground term $t \in \mathcal{T}(\mathcal{F})$ such that $t \rightarrow^{\not\epsilon *}_{\mathcal{A}} q$. An automaton $\mathcal{A}$ is deterministic if for all ground terms $s \in \mathcal{T}(\mathcal{F})$ and all states q, q' of $\mathcal{A}$, if $s \rightarrow_{\mathcal{A}}{}^* q$ and $s \rightarrow_{\mathcal{A}}{}^* q'$ then $q = q'$.

3 Equational Tree Automata Completion

3.1 Tree Automata Completion General Principle

Starting from a tree automaton $\mathcal{A}_0 = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta_0 \rangle$ and a left-linear TRS $\mathcal{R}$, the aim of the completion algorithm is to compute a tree automaton $\mathcal{A}^*$ such that $\mathcal{L}(\mathcal{A}^*) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}_0))$ or $\mathcal{L}(\mathcal{A}^*) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}_0))$. Tree automata completion successively computes tree automata $\mathcal{A}^1_{\mathcal{R}}$, $\mathcal{A}^2_{\mathcal{R}}$, ...such that $\forall i \geq 0 : \mathcal{L}(\mathcal{A}^i_{\mathcal{R}}) \subseteq \mathcal{L}(\mathcal{A}^{i+1}_{\mathcal{R}})$ and if $s \in \mathcal{L}(\mathcal{A}^i_{\mathcal{R}})$, and $s \rightarrow_{\mathcal{R}} t$ then $t \in \mathcal{L}(\mathcal{A}^{i+1}_{\mathcal{R}})$. For $k \in \mathbb{N}$, if $\mathcal{L}(\mathcal{A}^k_{\mathcal{R}}) = \mathcal{L}(\mathcal{A}^{k+1}_{\mathcal{R}})$ then $\mathcal{A}^k_{\mathcal{R}}$ is a fixpoint and we note it $\mathcal{A}^*_{\mathcal{R}}$. To construct $\mathcal{A}^{i+1}_{\mathcal{R}}$ from $\mathcal{A}^i_{\mathcal{R}}$, we achieve a *completion step* which consists in finding *critical pairs* between $\rightarrow_{\mathcal{R}}$ and $\rightarrow_{\mathcal{A}^i_{\mathcal{R}}}$. For a substitution $\sigma : \mathcal{X} \mapsto \mathcal{Q}$ and a rule $l \rightarrow r \in \mathcal{R}$, a critical pair is an instance $l\sigma$ of l such that there exists $q \in \mathcal{Q}$ satisfying $l\sigma \rightarrow^*_{\mathcal{A}^i_{\mathcal{R}}} q$ and $r\sigma \not\rightarrow^*_{\mathcal{A}^i_{\mathcal{R}}} q$. For $r\sigma$ to be recognized by the same state and thus model the rewriting of $l\sigma$ into $r\sigma$, it is enough to add the necessary transitions to $\mathcal{A}^i_{\mathcal{R}}$ to obtain $\mathcal{A}^{i+1}_{\mathcal{R}}$ such that $r\sigma \rightarrow^*_{\mathcal{A}^{i+1}_{\mathcal{R}}} q$.

In [22, 15], critical pairs are joined in the following way:

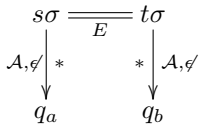


From an algorithmic point of view, there remains two problems to solve: find all the critical pairs $(l \to r, \sigma, q)$ and find the transitions to add to $\mathcal{A}^i_{\mathcal{R}}$ to have $r\sigma \to^*_{\mathcal{A}^{i+1}_{\mathcal{R}}} q$. The first problem, called matching, can be efficiently solved using a specific algorithm [11]. The second problem is solved using a normalization algorithm [13]. To have $r\sigma \to^*_{\mathcal{A}^{i+1}_{\mathcal{R}}} q'$ we need a transition of the form $r\sigma \to q'$ in $\mathcal{A}^{i+1}_{\mathcal{R}}$. However, this transitions may not be normalized. In this case, it is necessary to introduce new states and new transitions. For instance, to normalize a transition $f(g(a), h(q_1)) \to q'$ w.r.t. a tree automaton $\mathcal{A}^i_{\mathcal{R}}$ with transitions $a \to q_1$, $b \to q_1$, $g(q_1) \to q_1$, we first rewrite $f(g(a), h(q_1))$ with transitions of $\mathcal{A}^i_{\mathcal{R}}$. We obtain $f(q_2, h(q_1))$. Then we introduce the new state q_2 and the new transition $h(q_1) \to q_2$ to recognize the term $h(q_1)$. The new transitions to add to $\mathcal{A}^i_{\mathcal{R}}$ are thus: $h(q_1) \to q_2$, $f(q_1, q_2) \to q'$ and $q' \to q$.

3.2 Simplification of Tree Automata by Equations

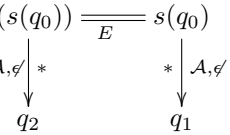
Since completion adds new transitions and new states to the automaton to join critical pairs, it may diverge. Divergence is avoided by *simplifying* the tree automaton w.r.t. a set of equations E. This operation permits to over-approximate languages that cannot be recognized *exactly* using tree automata completion, *e.g.* non regular languages. The simplification operation consists in finding E-equivalent terms recognized in $\mathcal{A}$ by different states and then by merging those states together.

▶ **Definition 1** (Simplification relation)**.** Let $\mathcal{A} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a tree automaton and E be a set of equations. For $s = t \in E$, $\sigma : \mathcal{X} \mapsto \mathcal{Q}$, $q_a, q_b \in \mathcal{Q}$ such that $s\sigma \to^{\not\epsilon *}_{\mathcal{A}} q_a$, $t\sigma \to^{\not\epsilon *}_{\mathcal{A}} q_b$, i.e.



and $q_a \neq q_b$ then $\mathcal{A}$ is *simplified* into $\mathcal{A}'$, denoted by $\mathcal{A} \leadsto_E \mathcal{A}'$, where $\mathcal{A}'$ is $\mathcal{A}$ where q_b is replaced by q_a in $\mathcal{Q}$, $\mathcal{Q}_f$ and Δ. ◇

▶ **Example 2.** Let $E = \{s(s(x)) = s(x)\}$ and $\mathcal{A}$ be the tree automaton with $\mathcal{Q}_f = \{q_2\}$ and set of transitions $\Delta = \{a \to q_0, s(q_0) \to q_1, s(q_1) \to q_2\}$. Hence $\mathcal{L}(\mathcal{A}) = \{s(s(a))\}$. We can perform a simplification step using the equation $s(s(x)) = s(x)$ because we found a substitution $\sigma = \{x \mapsto q_0\}$ such that:



Hence, $\mathcal{A} \leadsto_E \mathcal{A}'$ where $\mathcal{A}'$ is $\mathcal{A}$ where q_2 is replaced by q_1, *i.e.* $\mathcal{A}'$ is the automaton with $\mathcal{Q}'_f = \{q_1\}$, $\Delta = \{a \to q_0, s(q_0) \to q_1, s(q_1) \to q_1\}$. Note that $\mathcal{L}(\mathcal{A}') = \{s^*(s(a))\}$.

The simplification relation $\rightsquigarrow_E$ is terminating and confluent (modulo state renaming) [15]. In the following, we note $\mathcal{S}_E(\mathcal{A})$ the unique automaton (modulo renaming) $\mathcal{A}'$ such that $\mathcal{A} \rightsquigarrow_E^* \mathcal{A}'$ and $\mathcal{A}'$ is irreducible (it cannot be simplified further).

3.3 The full Completion Algorithm

▶ **Definition 3** (Automaton completion)**.** Let $\mathcal{A}$ be a tree automaton, $\mathcal{R}$ a left-linear TRS and E a set of equations.

- $\mathcal{A}^0_{\mathcal{R},E} = \mathcal{A}$,
- $\mathcal{A}^{n+1}_{\mathcal{R},E} = \mathcal{S}_E\left(\mathcal{C}_\mathcal{R}(\mathcal{A}^n_{\mathcal{R},E})\right)$, for $n \geq 0$ where $\mathcal{C}_\mathcal{R}(\mathcal{A}^n_{\mathcal{R},E})$ is a tree automaton such that all critical pairs of $\mathcal{A}^n_{\mathcal{R},E}$ are joined.

If there exists $k \in \mathbb{N}$ such that $\mathcal{A}^k_{\mathcal{R},E} = \mathcal{A}^{k+1}_{\mathcal{R},E}$, then we note $\mathcal{A}^*_{\mathcal{R},E}$ for $\mathcal{A}^k_{\mathcal{R},E}$.

▶ **Example 4.** Let $\mathcal{R} = \{f(x,y) \to f(s(x),s(y))\}$, $E = \{s(s(x)) = s(x)\}$ and $\mathcal{A}^0$ be the tree automaton with set of transitions $\Delta = \{f(q_a,q_b) \to q_0, a \to q_a, b \to q_b\}$, i.e. $\mathcal{L}(\mathcal{A}^0) = \{f(a,b)\}$. The completion ends after two completion steps on $\mathcal{A}^2_{\mathcal{R},E}$ which is a fixpoint $\mathcal{A}^*_{\mathcal{R},E}$. Completion steps are summed up in the following table. To simplify the presentation, we do not repeat the common transitions: $\mathcal{A}^i_{\mathcal{R},E}$ and $\mathcal{C}_\mathcal{R}(\mathcal{A}^i)$ columns are supposed to contain all transitions of $\mathcal{A}^0, \ldots, \mathcal{A}^{i-1}_{\mathcal{R},E}$.

$\mathcal{A}^0$	$\mathcal{C}_\mathcal{R}(\mathcal{A}^0)$	$\mathcal{A}^1_{\mathcal{R},E}$	$\mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})$	$\mathcal{A}^2_{\mathcal{R},E}$
$f(q_a,q_b) \to q_0$	$f(q_1,q_2) \to q_3$	$f(q_1,q_2) \to q_3$	$f(q_4,q_5) \to q_6$	$f(q_1,q_2) \to q_6$
$a \to q_a$	$s(q_a) \to q_1$	$s(q_a) \to q_1$	$s(q_1) \to q_4$	$s(q_1) \to q_1$
$b \to q_b$	$s(q_b) \to q_2$	$s(q_b) \to q_2$	$s(q_2) \to q_5$	$s(q_2) \to q_2$
	$q_3 \to q_0$	$q_3 \to q_0$	$q_6 \to q_3$	

On $\mathcal{A}^0$, there is one critical pair $f(q_a,q_b) \to^*_{\mathcal{A}_0} q_0$ and $f(q_a,q_b) \to_\mathcal{R} f(s(q_a),s(q_b))$. The automaton $\mathcal{C}_\mathcal{R}(\mathcal{A}^0)$ contains all the transitions of $\mathcal{A}^0$ with the new transitions (and the new states) necessary to join the critical pair, *i.e.* to have $f(s(q_a),s(q_b)) \to^*_{\mathcal{C}_\mathcal{R}(\mathcal{A}^0)} q_0$. The automaton $\mathcal{A}^1_{\mathcal{R},E}$ is exactly $\mathcal{C}_\mathcal{R}(\mathcal{A}^0)$ because simplification by equations do not apply. Then, $\mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})$ contains all the transitions of $\mathcal{A}^1_{\mathcal{R},E}$ and $\mathcal{A}^0$ plus those obtained by the resolution of the critical pair $f(q_1,q_2) \to^*_{\mathcal{A}^1_{\mathcal{R},E}} q_3$ and $f(q_1,q_2) \to_\mathcal{R} f(s(q_1),s(q_2))$. On $\mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})$ simplification using the equation $s(s(x)) = s(x)$ can be applied on the following instances: $s(s(q_a)) = s(q_a)$ and $s(s(q_b)) = q_b$. Since $s(s(q_a)) \to^*_{\mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})} q_4$ and $s(q_a) \to^*_{\mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})} q_1$, simplification merges q_4 with q_1. Similarly, simplification on $s(s(q_b)) = q_b$ merges q_5 with q_2. Thus, $\mathcal{A}^2_{\mathcal{R},E} = \mathcal{C}_\mathcal{R}(\mathcal{A}^1_{\mathcal{R},E})$ where q_4 is replaced by q_1 and q_5 is replaced q_2. This automaton is a fixed point because it has no other critical pairs (they are all joined).

3.4 Three Theorems on Completion

Tree automata completion enjoys three theorems defining its main properties. The first theorem is about *termination*. It defines a sufficient condition on E for completion to terminates. The second is a *sound approximation* theorem guaranteeing that completion always compute a tree automaton recognizing an over-approximation of reachable terms. This is the lower bound theorem. The third one, a *precision* theorem, guarantees that the computed automaton recognizes no more terms than $\mathcal{R}/E$ reachable terms. This is the upper bound theorem. We first state the soundness theorem.

▶ **Theorem 5** (Lower bound [15]). *Let $\mathcal{R}$ be a left-linear TRS, $\mathcal{A}$ be a tree automaton and E be a set of equations. If completion terminates on $\mathcal{A}^*_{\mathcal{R},E}$ then*

$$\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$$

To state the upper bound theorem, we need the notion of $\mathcal{R}/E$-coherent tree automaton we now define.

▶ **Definition 6** (Coherent automaton). Let $\mathcal{A} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ a tree automaton, $\mathcal{R}$ a TRS and E a set of equations. The automaton $\mathcal{A}$ is said to be $\mathcal{R}/E$-*coherent* if $\forall q \in \mathcal{Q} : \exists s \in \mathcal{T}(\mathcal{F})$:

$$s \to_{\mathcal{A}}^{\not\epsilon *} q \wedge [\forall t \in \mathcal{T}(\mathcal{F}) : (t \to_{\mathcal{A}}^{\not\epsilon *} q \implies s =_E t) \wedge (t \to_{\mathcal{A}}{}^{*} q \implies s \to_{\mathcal{R}/E}^{*} t)].$$

The intuition behind $\mathcal{R}/E$-coherence is the following. A $\mathcal{R}/E$-coherent is $\not\epsilon$reduced, its ϵ-transitions represent rewriting steps and normalized transitions recognize E-equivalence classes. More precisely, in a $\mathcal{R}/E$-coherent tree automaton, if two terms s, t are recognized into the same state q using only normalized transitions then they belong to the same E-equivalence class. Otherwise, if at least one ϵ-transition is necessary to recognize, say, t into q then at least one step of rewriting was necessary to obtain t from s.

▶ **Example 7.** Let $\mathcal{R} = \{a \to b\}$, $E = \{c = d\}$ and $\mathcal{A} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ with $\Delta = \{a \to q_0, b \to q_1, c \to q_2, d \to q_2, q_1 \to q_0\}$. The automaton $\mathcal{A}$ is $\mathcal{R}/E$-coherent because it is $\not\epsilon$reduced and the state q_2 recognizes with $\to_{\Delta}^{\not\epsilon}$ two terms c and d but they satisfy $c =_E d$. Finally, $a \to_{\Delta}^{*} q_0$ and $b \to_{\Delta}^{*} q_0$ but $a \to_{\Delta}^{\not\epsilon} q_0$, $b \to_{\Delta}^{\not\epsilon} q_1 \to q_0$ and $a \to_{\mathcal{R}} b$.

▶ **Theorem 8** (Upper bound [15]). *Let $\mathcal{R}$ be a left-linear TRS, E a set of equations and $\mathcal{A}$ a $\mathcal{R}/E$-coherent tree automaton. For any $i \in \mathbb{N}$:*

$$\mathcal{L}(\mathcal{A}^i_{\mathcal{R},E}) \subseteq \mathcal{R}^*_E(\mathcal{L}(\mathcal{A})) \qquad \text{and} \qquad \mathcal{A}^i_{\mathcal{R},E} \text{ is } \mathcal{R}/E\text{-coherent}$$

Finally, we state the termination theorem which relies on the E-compatibility assumption. Roughly, E-compatibility is the symmetric of E-coherence. An automaton $\mathcal{A}$ is E-compatible if for all states $q_1, q_2 \in \mathcal{B}$ and all terms $s, t \in \mathcal{T}(\mathcal{F})$ such that $s \to_{\mathcal{A}}^{\not\epsilon *} q_1$, $t \to_{\mathcal{A}}^{\not\epsilon *} q_2$ and $s =_E t$ then we have $q_1 = q_2$.

▶ **Theorem 9** (Termination of completion [13]). *Let $\mathcal{A}$ be a $\not\epsilon$-reduced tree automaton, $\mathcal{R}$ a left-linear TRS, $j \in \mathbb{N}$, and E a set of equations such that $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite. If for all $i \geq j$, $\mathcal{A}^i_{\mathcal{R},E}$ is E-compatible then there exists an integer n such that $\mathcal{A}^n_{\mathcal{R},E}$ is a fixpoint.*

To prove our final result, we first have to generalize Theorems 8 and 9 to discard the technical $\mathcal{R}/E$-coherence and E-compatibility assumptions. This is the objective of the next sections.

4 From automata to equations and vice versa

The above termination theorem uses the assumption that automata $\mathcal{A}^i_{\mathcal{R},E}$ are all E-compatible. This assumption is not true in general and is not preserved by tree automaton completion: an automaton $\mathcal{A}^i_{\mathcal{R},E}$ can be E-compatible but $\mathcal{A}^{i+1}_{\mathcal{R},E}$ may not be as we show on the following running example.

▶ **Example 10.** Let $\mathcal{F} = \{f : 1, a : 0, b : 0, c : 0\}$, $\mathcal{R} = \{f(x) \to f(f(x)), f(f(x)) \to a\}$, $\mathcal{A}$ be the automaton such that $\Delta = \{a \to q_1, c \to q_1, f(q_1) \to q_f\}$ and $E = \{f(x) = b\}$. Note that $\mathcal{T}(\mathcal{F})/_{=_E}$ has 3 equivalence classes: the class of $\{a\}$, the class of $\{b, f(a), f(b), f(c), \ldots\}$ and

the class of $\{c\}$. However, completion does not terminate on this example. Automaton $\mathcal{A}$ is E-compatible ($f(a) =_E f(c)$ and they are recognized by the same state: q_f) but $\mathcal{A}^1_{\mathcal{R},E}$ is not: it has one new state q_2 and contains additional transitions $\{f(q_f) \to q_1, q_1 \to q_f\}$. We thus have $f(f(a)) \to^{\not\epsilon *}_{\mathcal{A}^1_{\mathcal{R},E}} q_1$ and $f(a) \to^{\not\epsilon *}_{\mathcal{A}^1_{\mathcal{R},E}} q_f$ and $f(f(a)) =_E f(a)$ but $q_1 \neq q_f$. In fact, since b is not recognized by $\mathcal{A}^n_{\mathcal{R},E}$ for any natural n, the equation $f(x) = b$ of E cannot be used for simplification and completion diverges.

Note that, E-compatibility can be satisfied and preserved for particular cases of $\mathcal{R}$ and E. For instance, it is preserved by completion when $\mathcal{R}$ encodes a typed functional program [13]. Here, we show how to remove the E-compatibility assumption by transforming the set E into a set $E_{\mathcal{B}}$ such that completion preserves $E_{\mathcal{B}}$-compatibility and thus terminates with $E_{\mathcal{B}}$[1]. We also build $E_{\mathcal{B}}$ so that its precision is similar to E, *i.e.* $=_E \equiv =_{E_{\mathcal{B}}}$. This transformation is based on the Myhill-Nerode theorem for trees [19, 7]. We first produce a tree automaton $\mathcal{B}$ whose states recognize the equivalence classes of E. Then, from $\mathcal{B}$, we perform the inverse operation and obtain a set $E_{\mathcal{B}}$ whose set of equivalence classes is similar to the classes of E, but whose equations avoid the problem shown in Example 10. Since deciding finiteness of $\mathcal{T}(\mathcal{F})/_{=_E}$ and deciding $=_E$ is not always possible, we also propose an alternative version of this transformation using standard tools of rewriting : termination proofs and normalization. With this alternative transformation, $E_{\mathcal{B}}$ still ensure the termination of completion but can be more precise than E, *i.e.* $=_E \supseteq =_{E_{\mathcal{B}}}$.

4.1 From equations to automata

Provided that $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite, the Myhill-Nerode theorem for trees [19, 7] strongly relates $\mathcal{T}(\mathcal{F})/_{=_E}$ with tree automata. This theorem is constructive and provides an algorithm to switch from one form to the other, provided that $=_E$ is decidable. In the following we call **MN** the function building a tree automaton from a set of equations E, using the algorithm of [19].

▶ **Theorem 11** (Myhill-Nerode theorem for trees [19]). *If $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite and $=_E$ decidable, $\mathcal{B} = \mathbf{MN}(E)$ is a reduced, deterministic, epsilon-free and complete tree automaton such that for all $s, t \in \mathcal{T}(\mathcal{F})$, $s =_E t \iff (\exists q : \{s, t\} \subseteq \mathcal{L}(\mathcal{B}, q))$.*

However, determining whether $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite is not decidable in general. This is an unpublished result of S. Tison [23]. Since termination of the translation from E to $\mathcal{B}$ depends on the finiteness of $\mathcal{T}(\mathcal{F})/_{=_E}$, to use this algorithm we need, at least, a criterion for this property. Besides, the algorithm proposed in [19] needs $=_E$ to be decidable, which is not always true.

Thus, we propose an alternative technique based on the TRS $\overrightarrow{E} = \{u \to v \mid u = v \in E\}$, where all equations $u = v$ are oriented so that $\overrightarrow{E}$ can be shown terminating. If we can orient E in $\overrightarrow{E}$ so that it is weakly terminating and $IRR(\overrightarrow{E})$ is finite then so is $\mathcal{T}(\mathcal{F})/_{=_E}$. This due to the fact that $card(\mathcal{T}(\mathcal{F})/_{=_E}) \leq card(IRR(\overrightarrow{E}))$[2]. Note that the opposite is not

[1] Another solution is to complete the automaton $\mathcal{A}^n_{\mathcal{R},E}$ with transitions recognizing the complement of $\mathcal{L}(\mathcal{A}^n_{\mathcal{R},E})$, so that all equations can be applied. This solves the termination problem presented in Example 10, but it may introduce additional approximations. In particular, with this solution, the precision theorem of completion (Theorem 8) no longer holds.

[2] For all terms $s \in \mathcal{T}(\mathcal{F})$, since $\overrightarrow{E}$ is weakly terminating, there exist a natural number k and a finite rewriting sequence $s \to_{\overrightarrow{E}} s_1 \to_{\overrightarrow{E}} \ldots \to_{\overrightarrow{E}} s_k$ such that $s_k \in IRR(\overrightarrow{E})$. Thus, we can build an equational derivation $s =_E s_1 =_E \ldots =_E s_k$. Since $s =_E s_k$ and $s_k \in IRR(\overrightarrow{E})$ there cannot be more than $card(IRR(\overrightarrow{E}))$ equivalence classes in $\mathcal{T}(\mathcal{F})/_{=_E}$.

true and, in particular, that $IRR(\overrightarrow{E})$ may be infinite though $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite[3]. Hopefully, finiteness of $IRR(\overrightarrow{E})$ is decidable [7]. Besides, we replace checking $s =_E t$ by a (weaker) test on normal forms of s and t. If $\overrightarrow{E}$ is weakly terminating, then we check if there exists a term u such that $s \to^!_{\overrightarrow{E}} u$, $t \to^!_{\overrightarrow{E}} u$. Note that, without confluence of $\overrightarrow{E}$, irreducible terms of $IRR(\overrightarrow{E})$ may not coincide with equivalence classes of $\mathcal{T}(\mathcal{F})/_{=_E}$. We will see in Section 5.2 that weak termination of $\overrightarrow{E}$ and finiteness of $IRR(\overrightarrow{E})$ are, in fact, sufficient to guarantee termination of the completion. Now, we propose a function **E2A** which is a relaxed version of **MN**, *i.e.* $\mathbf{E2A}(\overrightarrow{E})$ can have more states that $\mathbf{MN}(E)$: $\mathbf{MN}(E)$ has $card(\mathcal{T}(\mathcal{F})/_{=_E})$ states and $\mathbf{E2A}(\overrightarrow{E})$ has $card(IRR(\overrightarrow{E}))$ states, where $card(\mathcal{T}(\mathcal{F})/_{=_E}) \leq card(IRR(\overrightarrow{E}))$, as shown above.

▶ **Definition 12** (function **E2A**). Let $\overrightarrow{E}$ be a TRS, $\mathcal{Q}$ be a set of states and Δ be a set of transitions. Let $state : \mathcal{T}(\mathcal{F}) \mapsto \mathcal{Q}$ be an injective function mapping ground terms to state symbols. $\mathbf{E2A}(\overrightarrow{E}) = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}, \Delta \rangle$ where $\mathcal{Q} = \{state(u) \mid u \in IRR(\overrightarrow{E})\}$ and $\Delta = \{f(state(u_1), \ldots, state(u_n)) \to state(u) \mid u_1, \ldots, u_n, u \in IRR(\overrightarrow{E})$ and $f(u_1, \ldots, u_n) \to^!_{\overrightarrow{E}} u\}$

Note that **E2A** builds a finite automaton as soon as $IRR(\overrightarrow{E})$ is finite. Weak termination of $\overrightarrow{E}$ is enough because for each term s we only need one term t such that $s \to^!_{\overrightarrow{E}} t$. We assume that the strategy for rewriting s into t is known. Besides, for $\mathbf{E2A}(\overrightarrow{E})$ to be deterministic, we need the assumption that $\to^!_{\overrightarrow{E}}$ is, itself, deterministic. Thus, we assume that $\to^!_{\overrightarrow{E}}$ uses a deterministic strategy leading to irreducible terms. For instance, if $\overrightarrow{E}$ is innermost terminating, we can assume that $\to^!_{\overrightarrow{E}}$ uses leftmost innermost rewriting.

▶ **Example 13.** Consider the $E = \{f(x) = b\}$ of Example 10. Let us choose $\overrightarrow{E} = \{f(x) \to b\}$. Since $\overrightarrow{E}$ is left-linear, we can build a tree automaton recognizing $IRR(\overrightarrow{E})$ in an effective way [8, 6]. This tree automaton recognize only 3 irreducible terms a, b, c. Furthermore $\overrightarrow{E}$ is terminating, we can thus build the automaton $\mathbf{E2A}(\overrightarrow{E})$. It has 3 states q_0, q_1, q_2 such that $state(a) = q_0$, $state(b) = q_1$ and $state(c) = q_2$. It has six transitions: $a \to q_0$ (because $a \to^!_{\overrightarrow{E}} a$), $b \to q_1$ (because $b \to^!_{\overrightarrow{E}} b$), $c \to q_2$ (because $c \to^!_{\overrightarrow{E}} c$), $f(q_0) \to q_1$ (because $f(a) \to^!_{\overrightarrow{E}} b$), $f(q_1) \to q_1$ (because $f(b) \to^!_{\overrightarrow{E}} b$), $f(q_2) \to q_1$ (because $f(c) \to^!_{\overrightarrow{E}} b$).

The automaton $\mathbf{E2A}(\overrightarrow{E})$ enjoys properties close to the ones of $\mathbf{MN}(E)$.

▶ **Lemma 14.** *Let $\mathcal{B} = \mathbf{E2A}(\overrightarrow{E})$. For all states $q \in \mathcal{B}$ there exists an irreducible term $u \in IRR(\overrightarrow{E})$ such that $u \to^*_{\mathcal{B}} q$ and $state(u) = q$.*

Proof. For any state q, there exists an irreducible term u such that $q = state(u)$. Now, we prove by induction on the height of u that $u \to^*_{\mathcal{B}} q$. If u is a constant then, by definition of $\mathcal{B}$, for all irreducible term t such that $u \to^!_{\overrightarrow{E}} t$, the transition $u \to state(t)$ belongs to $\mathcal{B}$. Since u is irreducible then $u = t$. If $u = f(u_1, \ldots, u_n)$, since u is irreducible then so are u_i, $1 \leq i \leq n$. Applying the induction hypothesis on the u_i's, we get that $u_i \to^*_{\mathcal{B}} q_i$ where $q_i = state(u_i)$ for $1 \leq i \leq n$. We conclude the proof by remarking that, by construction of $\mathcal{B}$, the transition $f(q_1, \ldots, q_n) \to q$ necessarily belongs to $\mathcal{B}$. ◀

▶ **Lemma 15.** *If $IRR(\overrightarrow{E})$ is finite, $\overrightarrow{E}$ weakly terminates, and $\to^!_{\overrightarrow{E}}$ is deterministic then $\mathcal{B} = \mathbf{E2A}(\overrightarrow{E})$ is a reduced, deterministic, epsilon-free and complete tree automaton.*

---

[3] For instance, if $E = \{f(a) = b, a = b\}$ and $\overrightarrow{E} = \{f(a) \to b, a \to b\}$.

Proof. The first two assumptions are necessary to be sure that $\mathcal{B}$ exists. Automaton $\mathcal{B}$ is reduced because of Lemma 14. If $\rightarrow^!_{\overrightarrow{E}}$ is deterministic, then in the definition of Δ, there is only one possible term t s.t. $f(t_1, \ldots, t_n) \rightarrow^!_{\overrightarrow{E}} t$. Thus, for each configuration $f(q_1, \ldots, q_n)$ there is only one possible state q. This proves that $\mathcal{B}$ is deterministic. Automaton $\mathcal{B}$ is trivially epsilon-free. Finally, for completeness of $\mathcal{B}$, we can show that for all terms $s \in \mathcal{T}(\mathcal{F})$, there exists a state q s.t. $s \rightarrow^*_{\mathcal{B}} q$ by induction on the height of s. If s is a constant a, then by definition of $\mathcal{B}$, we know that there exists a transition $a \rightarrow state(t)$. For the inductive case, if $s = f(s_1, \ldots, s_n)$ then we know that there exists irreducible terms t_i and states q_i, $1 \leq i \leq n$ such that $s_i \rightarrow^*_{\mathcal{B}} q_i$. Then, by construction of $\mathcal{B}$, we know that there exists a transition $f(q_1, \ldots, q_n) \rightarrow q$, which concludes the proof. ◀

Theorem 11 tightly relates equivalence classes of $\mathcal{T}(\mathcal{F})/_{=_E}$ with languages recognized by $\mathbf{MN}(E)$. This relation exists in $\mathbf{E2A}(\overrightarrow{E})$ but is slightly relaxed.

▶ **Lemma 16.** *Let E be a set of equations such that $IRR(\overrightarrow{E})$ is finite, $\overrightarrow{E}$ weakly terminates and $\mathcal{B} = \mathbf{E2A}(\overrightarrow{E})$. For all states $q \in \mathcal{B}$, for all terms $s, t \in \mathcal{T}(\mathcal{F})$ if $s \rightarrow^*_{\mathcal{B}} q$ and $t \rightarrow^*_{\mathcal{B}} q$ then $s =_E t$.*

Proof. First we prove that for all states $q \in \mathcal{B}$, for all terms $s \in \mathcal{T}(\mathcal{F})$ such that $s \rightarrow^*_{\mathcal{B}} q$, there exists an irreducible term u such that $q = state(u)$ and $s \rightarrow^!_{\overrightarrow{E}} u$. We prove this property by induction on the height of s. If s is a constant a, for $a \rightarrow^*_{\mathcal{B}} q$ to hold we know that there is necessarily an irreducible term u and a transition $a \rightarrow q$ in $\mathcal{B}$ such that $q = state(u)$ and $a \rightarrow^!_{\overrightarrow{E}} u$. For the inductive case, let $s = f(s_1, \ldots, s_n)$. Since $s \rightarrow^*_{\mathcal{B}} q$ we know that $s_i \rightarrow^*_{\mathcal{B}} q_i$ and there exists a transition $f(q_1, \ldots, q_n) \rightarrow q \in \mathcal{B}$. Applying the induction hypothesis on s_i, $1 \leq i \leq n$, we get that there exists irreducible terms u_i such that $q_i = state(u_i)$ and $s_i \rightarrow^!_{\overrightarrow{E}} u_i$ for $1 \leq i \leq n$. Thus $f(s_1, \ldots, s_n) \rightarrow^!_{\overrightarrow{E}} f(u_1, \ldots, u_n)$. Besides, for transition $f(q_1, \ldots, q_n) \rightarrow q$ to belong to $\mathcal{B}$, we know that there exists an irreducible term u such that $q = state(u)$ and $f(u_1, \ldots, u_n) \rightarrow^!_{\overrightarrow{E}} u$. We thus have $f(s_1, \ldots, s_n) \rightarrow^!_{\overrightarrow{E}} f(u_1, \ldots, u_n) \rightarrow^!_{\overrightarrow{E}} u$. This ends the proof by induction. Now, since $s \rightarrow^*_{\mathcal{B}} q$ and $t \rightarrow^*_{\mathcal{B}} q$, we know that exists irreducible terms u and u' such that $q = state(u)$, $s \rightarrow^!_{\overrightarrow{E}} u$, $q = state(u')$, $t \rightarrow^!_{\overrightarrow{E}} u'$. Since $state$ is an injective function, we get $u = u'$. Finally, $s \rightarrow^!_{\overrightarrow{E}} u$, $t \rightarrow^!_{\overrightarrow{E}} u$ implies $s =_E u =_E t$. ◀

4.2 From automata to equations

In the other direction, from a tree automaton $\mathcal{B}$ it is possible to build a set of equations $E_{\mathcal{B}}$ such that languages recognized by states of $\mathcal{B}$ and equivalence classes of $\mathcal{T}(\mathcal{F})/_{=_{E_{\mathcal{B}}}}$ coincide. This is the function **A2E** that is also described in [19]. We reformulate this function because we need some additional properties on the generated set of equations for completion to terminate. For simplicity we assume that $\mathcal{B}$ is **R**educed and epsilon **F**ree. Some properties of $E_{\mathcal{B}}$ will hold only if $\mathcal{B}$ is also **C**omplete and **D**eterministic. In the following, we use the **RF**, and **RDFC** short-hands for automata having the related properties. Recall that for all tree automaton, there exists an equivalent **RF** or **RDFC** automaton [7].

For **RF** automata, the construction of $E_{\mathcal{B}}$ is rather straightforward and is based on the same idea as in [19]: for all states q we identify a ground term recognized by q, a representative, and for all transitions $f(q_1, \ldots, q_n) \rightarrow q$ we generate an equation $f(t_1, \ldots, t_n) = t$ where t_i, $1 \leq i \leq n$ are representatives for q_i and t is a representative for q. However, for this set of equations to guarantee termination of completion it needs some redundancy: for each state we generate a *set* of *state representatives* and the equations are defined for each representative of the set. As shown in Example 10, the equation $f(x) = b$ cannot

be applied during completion because b does not occur in the tree automaton. However, a logic consequence of this equation is that $f(f(a)) =_E f(a)$ and terms $f(f(a))$ and $f(a)$ that occur in the tree automaton could be merged. In our setting the term $f(a)$ will be a state representative and the equation $f(f(a)) = f(a)$ will appear in the set of generated equations. Roughly, every constant symbol a appearing in a transition $a \to q$ is a state representative for q, and every term of the form $f(u_1, \ldots, u_n)$ is a state representative for q if there is a transition $f(q_1, \ldots, q_n) \to q$, the u_i's are state representatives for the q_i's and subterms of the u_i's are not state representatives of q. The fact that u_i's should not be state representatives of q will ensure finiteness of this set (Lemma 19). Now, we formally define state representatives.

▶ **Definition 17** (State representatives)**.** Let $\mathcal{B} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a **RF** tree automaton and $q \in \mathcal{Q}$. The set of state representatives of q of height lesser or equal to n, denoted by $[\![q]\!]^n_{\mathcal{B}}$, is inductively defined by:

- $[\![q]\!]^1_{\mathcal{B}} = \{a \mid a \to q \in \Delta\}$
- $[\![q]\!]^n_{\mathcal{B}} = [\![q]\!]^{n-1}_{\mathcal{B}} \cup \{f(u_1, \ldots, u_n) \mid f(q_1, \ldots, q_n) \to q \in \Delta \text{ and } \forall i \in \{1, \ldots, n\} : u_i \in [\![q_i]\!]_{\mathcal{B}},$ and $\forall p \in \mathcal{P}os(u_i) : u_i|_p \notin [\![q]\!]^{n-1}_{\mathcal{B}}\}$

In the above definition, the fact that $\mathcal{B}$ is reduced and epsilon free ensures that there exists at least one (non epsilon) transition for every state and that each state has at least one state representative.

▶ **Example 18.** Let $\mathcal{B}$ be the **RF** automaton that we obtained in Example 13 and whose set of transitions is $a \to q_0$, $b \to q_1$, $c \to q_2$, $f(q_0) \to q_1$, $f(q_1) \to q_1$, $f(q_2) \to q_1$.

- $[\![q_0]\!]^1_{\mathcal{B}} = \{a\}$, $[\![q_1]\!]^1_{\mathcal{B}} = \{b\}$, and $[\![q_2]\!]^1_{\mathcal{B}} = \{c\}$.
- $[\![q_0]\!]^2_{\mathcal{B}} = [\![q_0]\!]^1_{\mathcal{B}}$, $[\![q_1]\!]^2_{\mathcal{B}} = \{b, f(a), f(c)\}$, and $[\![q_2]\!]^2_{\mathcal{B}} = [\![q_2]\!]^1_{\mathcal{B}}$. The term $f(b)$ of height 2 and recognized by q_1 is not added to $[\![q_1]\!]^2_{\mathcal{B}}$ because its subterm b belongs to $[\![q_1]\!]^1_{\mathcal{B}}$.
- The fixpoint is reached because terms $f(f(a))$ and $f(f(c))$ recognized by q_1 are not added to $[\![q_1]\!]^3_{\mathcal{B}}$ because $f(a)$ and $f(c)$ belong to $[\![q_1]\!]^2_{\mathcal{B}}$.

We denote by $[\![q]\!]_{\mathcal{B}}$ the set of all state representatives for the state q, *i.e.* the fixpoint of the above equations. Now, we show that for all reduced and epsilon free automaton, such a fixpoint exists and is always a finite set.

▶ **Lemma 19** (The set of state representatives is finite)**.** *For all* **RF** *tree automaton $\mathcal{B}$, for all state $q \in \mathcal{B}$ there exists a natural number n for which the set $[\![q]\!]^n_{\mathcal{B}}$ is a fixpoint.*

Proof. We make a proof by contradiction. Assume that one set of state representative is infinite, say $[\![q]\!]_{\mathcal{B}}$. Since the set of terms belonging to $[\![q]\!]_{\mathcal{B}}$ is a regular language, we can use the (classical) pumping lemma on recognizable tree languages [7]. Since the language $[\![q]\!]_{\mathcal{B}}$ is infinite, from this lemma, we know that there exists a ground context $C_1[\,]$, a non empty ground context $C_2[\,]$ and a ground term u such that (big) state representatives belonging to $[\![q]\!]_{\mathcal{B}}$ are of the form $C_1[C_2^n[u]]$ for $n \geq 0$. Since $C_1[C_2^n[u]]$ belongs to $[\![q]\!]_{\mathcal{B}}$, we know that $C_1[C_2^n[u]] \to^*_{\mathcal{B}} q$. Thus, there exists a state $q' \in \mathcal{B}$ such that $C_2^n[u] \to^*_{\mathcal{B}} q'$, for $n \geq 0$ and $C_1[q'] \to^*_{\mathcal{B}} q$. Now, we focus on $C_2^n[u] \to^*_{\mathcal{B}} q'$. For this derivation to hold, we need to have states $q_1, \ldots, q_n$ such that: $C_2^n[u] \to^*_{\mathcal{B}} C_2^{n-1}[q_1] \to^*_{\mathcal{B}} \ldots \to^*_{\mathcal{B}} C_2[q_n] \to^*_{\mathcal{B}} q$. Let us choose for n a natural k that is strictly greater than the number of states of $\mathcal{B}$. By the pigeon hole principle, we know that at least two states in $q_1, \ldots, q_k$ are identical. Assume that those states are q_i and q_j with $i < j$. Thus, we have $C_2^i[u] \to^*_{\mathcal{B}} q_i$, and $C_2^j[u] \to^*_{\mathcal{B}} q_j$ with $i < j$ and $q_i = q_j$. By Definition 17, we know that each subterm of a state representative is, itself,

a representative. Thus, $C_2^i[u]$ and $C_2^j[u]$ are state representatives, and they belong to the same set $[\![q_i]\!]_{\mathcal{B}}$ since $C_2^i[u] \to_{\mathcal{B}}^* q_i$ and $C_2^j[u] \to_{\mathcal{B}}^* q_i$. This is a contradiction because $C_2^i[u]$ is a subterm of $C_2^j[u]$ and Definition 17 forbids that a term and one of its subterm belong to the same set of representatives. ◀

▶ **Definition 20** (function **A2E**: infers a set of equations $E_{\mathcal{B}}$ from a tree automaton $\mathcal{B}$)**.** Let $\mathcal{B} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a **RF** tree automaton. The set of equations $E_{\mathcal{B}}$ inferred from $\mathcal{B}$ is $\mathbf{A2E}(\mathcal{B}) = E_{\mathcal{B}} = \{f(u_1, \ldots, u_n) = u \mid f(q_1, \ldots, q_n) \to q \in \mathcal{B},\ u \in [\![q]\!]_{\mathcal{B}} \text{ and } u_i \in [\![q_i]\!]_{\mathcal{B}} \text{ for } 1 \leq i \leq n\}$.

▶ **Example 21.** Starting back from the automaton $\mathcal{B}$ and the state representatives of Example 18, the set $\mathbf{A2E}(\mathcal{B})$ contains the following equations: $a = a$ (because of transition $a \to q_0$), $c = c$ (because of transition $c \to q_2$), $b = b$, $b = f(a)$, $b = f(c)$ (because of transition $b \to q_1$), $f(a) = f(a)$, $f(a) = b$, $f(a) = f(c)$ (because of transition $f(q_0) \to q_1$), $f(f(a)) = f(a)$, $f(f(a)) = b$, $f(f(a)) = f(c)$, $f(b) = f(a)$, $f(b) = b$, $f(b) = f(c)$, $f(f(c)) = f(a)$, $f(f(c)) = b$, $f(f(c)) = f(c)$ (because of transition $f(q_1) \to q_1$, and $f(c) = f(a)$, $f(c) = b$, $f(c) = f(c)$ (because of transition $f(q_2) \to q_1$).

Since $\mathcal{B}$ is finite and sets of state representatives are, so is $E_{\mathcal{B}}$. Note that many equations of $E_{\mathcal{B}}$ are useless w.r.t. the underlying equational theory. This is the case, in the above example, for equations of the form $a = a$ as well as the equation $f(a) = f(c)$ which is redundant w.r.t. $b = f(a)$ and $b = f(c)$. However, as shown in Example 10 those equations are necessary for equational simplification to be complete w.r.t. $E_{\mathcal{B}}$ and completion to terminate. Indeed, with the above $E_{\mathcal{B}}$, completion of Example 10 terminates. Below, Theorem 26 shows that, if $\mathcal{B}$ is **RDFC** then completion with $\mathbf{A2E}(\mathcal{B})$ always terminates. Unsurprisingly, if $\mathcal{B}$ is deterministic then equivalence classes of $E_{\mathcal{B}}$ coincide with languages recognized by states of $\mathcal{B}$. This is the purpose of the next two lemmas. They both assume that $\mathcal{B}$ is complete which is a necessary assumption for the lemmas to hold for all terms $s \in \mathcal{T}(\mathcal{F})$. To prove that equivalence classes and recognized languages coincide, we first need an intermediate result.

▶ **Lemma 22.** *Let $\mathcal{B} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a* **RDFC** *tree automaton and $E_{\mathcal{B}} = \mathbf{A2E}(\mathcal{B})$. For all $s \in \mathcal{T}(\mathcal{F})$, there exists a unique state $q \in \mathcal{Q}$ such that $s \to_{\mathcal{B}}^* q$ and for all state representatives $u \in [\![q]\!]_{\mathcal{B}}$, $s =_{E_{\mathcal{B}}} u$.*

Proof. We make a proof by induction on the height of s. If s is a constant, since $\mathcal{B}$ is complete and deterministic there exists a unique transition $s \to q \in \Delta$. By construction of $E_{\mathcal{B}}$, we know that there are equations with s on the left-hand side and all state representatives of $[\![q]\!]_{\mathcal{B}}$ on the right-hand side. For all equation $s = u$ with $u \in [\![q]\!]_{\mathcal{B}}$ we thus trivially have $s =_{E_{\mathcal{B}}} u$. This concludes the base case.

Now, we assume that the property is true for terms of height lesser or equal to n. Let $s = f(t_1, \ldots, t_n)$ where $t_1, \ldots, t_n$ are terms of height lesser or equal to n. Since $\mathcal{B}$ is complete, we know that there exists a state q such that $f(t_1, \ldots, t_n) \to_{\mathcal{B}}^* q$, *i.e.* there exists states $q_1, \ldots, q_n$ such that $f(q_1, \ldots, q_n) \to q \in \Delta$ and $t_i \to_{\mathcal{B}}^* q_i$ for $1 \leq i \leq n$. Using the induction hypothesis we get that there exist states q_i' in $\mathcal{B}$ and terms $[\![q_i']\!]_{\mathcal{B}}$ such that $t_i \to_{\mathcal{B}}^* q_i$ and $t_i =_{E_{\mathcal{B}}} u_i$ for $u_i \in [\![q_i']\!]_{\mathcal{B}}$ and for $1 \leq i \leq n$. Since $\mathcal{B}$ is deterministic, from $t_i \to_{\mathcal{B}}^* q_i$ and $t_i \to_{\mathcal{B}}^* q_i'$ we get that $q_i = q_i'$ and thus $t_i =_{E_{\mathcal{B}}} u_i$ for $u_i \in [\![q_i]\!]_{\mathcal{B}}$, with $1 \leq i \leq n$. Besides, since $f(q_1, \ldots, q_n) \to q \in \Delta$, we know that $E_{\mathcal{B}}$ contains the equations $f(u_1, \ldots, u_n) = u$ for all $u_i \in [\![q_i]\!]_{\mathcal{B}}$, for all $1 \leq i \leq n$ and for all $u \in [\![q]\!]_{\mathcal{B}}$. Thus q is the unique state such that $f(t_1, \ldots, t_n) \to_{\mathcal{B}}^* q$. Furthermore, $f(t_1, \ldots, t_n) =_{E_{\mathcal{B}}} f(u_1, \ldots, u_n) =_{E_{\mathcal{B}}} u$ for all $u_i \in [\![q_i]\!]_{\mathcal{B}}$, for all $1 \leq i \leq n$ and for all $u \in [\![q]\!]_{\mathcal{B}}$. ◀

Now we can state the lemma relating equivalence classes of $E_{\mathcal{B}}$ and languages recognized by states of $\mathcal{B}$.

▶ **Lemma 23** (Equivalence classes of $E_{\mathcal{B}}$ coincide with languages recognized by states of $\mathcal{B}$). *Let $\mathcal{B} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a* **RDFC** *tree automaton and $E_{\mathcal{B}} =$ **A2E**$(\mathcal{B})$. For all $s, t \in \mathcal{T}(\mathcal{F})$, $s =_{E_{\mathcal{B}}} t \iff (\exists q : \{s, t\} \subseteq \mathcal{L}(\mathcal{B}, q))$.*

Proof. For s and t, using Lemma 22, we know that there exist unique states $q, q' \in \mathcal{Q}$ such that $s \to_{\mathcal{B}}^* q$, $t \to_{\mathcal{B}}^* q'$ and for all state representatives $u \in [\![q]\!]_{\mathcal{B}}$ and $v \in [\![q']\!]_{\mathcal{B}}$, we have $s =_{E_{\mathcal{B}}} u$ and $t =_{E_{\mathcal{B}}} v$. We first prove the left to right implication. From $s =_{E_{\mathcal{B}}} t$ we obtain that $u =_{E_{\mathcal{B}}} v$, where u and v are state representatives. By construction of term representatives, for all states q we know that $[\![q]\!]_{\mathcal{B}}$ only contains terms recognized by q in $\mathcal{B}$. Since $\mathcal{B}$ is deterministic, if $q \neq q'$ then we can conclude that $[\![q]\!]_{\mathcal{B}} \cap [\![q']\!]_{\mathcal{B}} = \emptyset$. Thus, the only possibility to have $u =_{E_{\mathcal{B}}} v$ is to have an equation $u = v$ in $E_{\mathcal{B}}$. This entails that u and v belong to the same set of representatives: $[\![q]\!]_{\mathcal{B}} = [\![q']\!]_{\mathcal{B}}$, which entails that $q = q'$. Then $s \to_{\mathcal{B}}^* q$ and $t \to_{\mathcal{B}}^* q$ entails that $\{s, t\} \subseteq \mathcal{L}(\mathcal{B}, q)$. To prove the right to left implication, it is enough to point out that because of the determinism of $\mathcal{B}$ having $t \to_{\mathcal{B}}^* q'$ (the initial assumption) and having $t \to_{\mathcal{B}}^* q$ (the fact that $t \in \mathcal{L}(\mathcal{B}, q)$) is possible only if $q = q'$. This entails that u and v have a common set of representatives and thus for all representatives u of this set $s =_{E_{\mathcal{B}}} u =_{E_{\mathcal{B}}} t$. ◀

▶ **Corollary 24** ($\mathcal{T}(\mathcal{F})/_{=_{E_{\mathcal{B}}}}$ is finite). *Let $\mathcal{B} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \Delta \rangle$ be a* **RDFC** *tree automaton. If $E_{\mathcal{B}}$ is the set of equations inferred from $\mathcal{B}$ then $\mathcal{T}(\mathcal{F})/_{=_{E_{\mathcal{B}}}}$ is finite.*

Proof. Using Lemma 22, we know that for all terms $t \in \mathcal{T}(\mathcal{F})$ there exist a state $q \in \mathcal{Q}$ and a state representative $u \in [\![q]\!]_{\mathcal{B}}$ such that $t \to_{\mathcal{B}}^* q$ and $t =_{E_{\mathcal{B}}} u$. Since the number of states of $\mathcal{B}$ is finite, and since the set of state representatives u is finite for all states of $\mathcal{B}$ (Lemma 19), so is the number of equivalence classes of $\mathcal{T}(\mathcal{F})/_{=_{E_{\mathcal{B}}}}$. ◀

5 Generalizing the termination theorem

5.1 Proving termination with $E_{\mathcal{B}}$

Now, we prove that if completion is performed using the set of equations $E_{\mathcal{B}}$, built from a **RDFC** tree automaton $\mathcal{B}$, then it always terminates. To prove this, we need to show several results on the limit automaton of completion. In the following, the automaton $\mathcal{A}^*$ is the limit of the (possibly) infinite completion of an initial $\not\epsilon$-reduced tree automaton $\mathcal{A}$ with $\mathcal{R}$ and $E_{\mathcal{B}}$. If the initial automaton is not $\not\epsilon$-reduced then completion may diverge. For instance, completion of automaton whose set of transitions is $\{f(q_0) \to q_1\}$, with $\mathcal{R} = \{f(x) \to f(f(x))\}$ and $E = \{f(a) = a\}$ diverges (simplification never happens because q_0 does not recognize any term). Now we show that all state representatives are recognized by epsilon-free derivations in $\mathcal{A}^*$.

▶ **Lemma 25** (All states of $\mathcal{A}^*$ recognize at least one state representative). *Let $\mathcal{R}$ be a TRS, $\mathcal{A}$ an $\not\epsilon$-reduced tree automaton, $\mathcal{B}$ an* **RDFC** *tree automaton and $E_{\mathcal{B}} =$ **A2E**$(\mathcal{B})$. Let $\mathcal{A}^*$ be the limit of the completion of $\mathcal{A}$ by $\mathcal{R}$ and $E_{\mathcal{B}}$. For all states $q \in \mathcal{A}^*$, for all terms $s \in \mathcal{T}(\mathcal{F})$ such that $s \to_{\mathcal{A}^*}^{\not\epsilon *} q$, there exists a state $q'_{\mathcal{B}} \in \mathcal{B}$, a term $u \in [\![q'_{\mathcal{B}}]\!]_{\mathcal{B}}$ such that $u =_{E_{\mathcal{B}}} s$ and $u \to_{\mathcal{A}^*}^{\not\epsilon *} q$.*

Proof. Note that if $\mathcal{A}$ is $\not\epsilon$-reduced, then so is $\mathcal{A}^*$. This is Lemma 44 of [13]. This is easy to figure out since all states added during completion recognize at least one term with $\to_{\mathcal{A}}^{\not\epsilon *}$,

and this is trivially preserved by simplification. By induction on the height of s we show that the representative u exists and is recognized by q. If s is of height 1 (it is a constant) then, by construction of state representatives, we know that s is a representative. Thus $s = u \to^{\not\varepsilon *}_{\mathcal{A}^*} q$.

For the inductive case, assume that the property is true for all terms of height lesser or equal to n. Let $s = f(s_1, \ldots, s_n)$ be a term of height $n+1$. By assumption, we know that $f(s_1, \ldots, s_n) \to^{\not\varepsilon *}_{\mathcal{A}^*} q$. From $f(s_1, \ldots, s_n) \to^{\not\varepsilon *}_{\mathcal{A}^*} q$, we obtain that there exists states $q_1, \ldots, q_n$ of $\mathcal{A}^*$ such that $s_i \to^{\not\varepsilon *}_{\mathcal{A}^*} q_i$ for $i = 1, \ldots, n$ and a transition $f(q_1, \ldots, q_n) \to q$ in $\mathcal{A}^*$. Using the induction hypothesis on q_i, $i = 1, \ldots, n$ we get that there exist state representatives u_i such that $s_i =_{E_\mathcal{B}} u_i$ and $u_i \to^{\not\varepsilon *}_{\mathcal{A}^*} q_i$ for $i = 1, \ldots, n$. Then, since $f(q_1, \ldots, q_n) \to q$ in $\mathcal{A}^*$ we thus have $f(u_1, \ldots, u_n) \to^{\not\varepsilon *}_{\mathcal{A}^*} q$. If $f(u_1, \ldots, u_n)$ is a state representative, then we are done since $f(s_1, \ldots, s_n) =_{E_\mathcal{B}} f(u_1, \ldots, u_n)$ and $f(u_1, \ldots, u_n) \to^{\not\varepsilon *}_{\mathcal{A}^*} q$. Otherwise, by definition of state representatives, for $u = f(u_1, \ldots, u_n)$ not to belong to the representatives there is a position p in u, different from the root position such that the subterm $u|_p$ is itself a state representative and it belongs to the same class as u, *i.e.* $u =_{E_\mathcal{B}} u|_p$. Since $u_1, \ldots, u_n$ are state representatives and $f(u_1, \ldots, u_n)$ is in the same equivalence class than $u|_p$ which is a state representative, we know that the equation $f(u_1, \ldots, u_n) = u|_p$ necessarily belong to $E_\mathcal{B}$. Besides, for $u \to^{\not\varepsilon *}_{\mathcal{A}^*} q$ to hold, we know that there exists a state q' such that $u[u|_p]_p \to^{\not\varepsilon *}_{\mathcal{A}^*} u[q']_p \to^{\not\varepsilon *}_{\mathcal{A}^*} q$. Thus, $f(u_1, \ldots, u_n) \to^{\not\varepsilon *}_{\mathcal{A}^*} q$ and $u|_p \to^{\not\varepsilon *}_{\mathcal{A}^*} q'$. Then, since $E_\mathcal{B}$ contains the equation $f(u_1, \ldots, u_n) = u|_p$, and since $\mathcal{A}^*$ is simplified w.r.t. $E_\mathcal{B}$, we necessarily have $q = q'$ in $\mathcal{A}^*$. Finally, we have $f(s_1, \ldots, s_n) =_{E_\mathcal{B}} f(u_1, \ldots, u_n) =_{E_\mathcal{B}} u|_p$ and $u|_p \to^{\not\varepsilon *}_{\mathcal{A}^*} q$ where $u|_p$ is a state representative. ◀

Now, we can state the termination Theorem with $E_\mathcal{B}$.

▶ **Theorem 26** (Completion with $E_\mathcal{B}$ terminates)**.** *Let $\mathcal{R}$ be a TRS, $\mathcal{A}$ an $\not\varepsilon$-reduced tree automaton, $\mathcal{B}$ be a* **RDFC** *tree automaton and $E_\mathcal{B} =$* **A2E**$(\mathcal{B})$*. Let n be the number of all states representatives of $\mathcal{B}$. The automaton $\mathcal{A}^*$, limit of the completion of $\mathcal{A}$ with $\mathcal{R}$ and $E_\mathcal{B}$, has a n states or less.*

Proof. Recall that the number n of state representatives is finite, this is Lemma 19. Assume that $\mathcal{A}^*$ has m distinct states with $m > n$. By Lemma 25, we know that for all state $q \in \mathcal{A}^*$, there exists a state representative u such that $u \to^{\not\varepsilon *}_{\mathcal{A}^*} q$. Since there are only n state representatives, by pigeon hole principle, we know that there is necessarily one state representative u recognized by two distinct states q_1 and q_2 of $\mathcal{A}^*$. Thus, $u \to^{\not\varepsilon *}_{\mathcal{A}^*} q_1$ and $u \to^{\not\varepsilon *}_{\mathcal{A}^*} q_2$. Besides, by construction of $E_\mathcal{B}$, we know that the equation $u = u$ is part of $E_\mathcal{B}$. This contradicts the fact that $\mathcal{A}^*$ is simplified w.r.t. $E_\mathcal{B}$. ◀

5.2 Building $E_\mathcal{B}$ from any set of equations E

Now, we combine the transformations **A2E** and **E2A** (or **A2E** and **MN**) to produce, from E, a set of equations $E_\mathcal{B}$ that ensures termination of completion and that is equivalent to E. We first prove that, whatever the combination may be, the transformation preserves precision of the approximation, *i.e.* that $E_\mathcal{B}$ is at least as precise as E.

▶ **Lemma 27.** *Let E be a set of equations. If $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite and $=_E$ is decidable then $E_\mathcal{B} =$* **A2E**$($**MN**$(E))$ *is such that $=_E \equiv =_{E_\mathcal{B}}$.*

Proof. Let $\mathcal{B} =$ **MN**$(\overrightarrow{E})$. From Lemma 15, we know that $\mathcal{B}$ is **RDFC** and languages recognized by states of $\mathcal{B}$ coincide with equivalence classes of E. Then, let $E_\mathcal{B} =$ **A2E**$(\mathcal{B})$. Using Lemma 23, we get that languages of $\mathcal{B}$ coincide with equivalence classes of $E_\mathcal{B}$. Thus, $=_E \equiv =_{E_\mathcal{B}}$. ◀

▶ **Lemma 28.** *Let E be a set of equations. If $IRR(\overrightarrow{E})$ is finite and $\overrightarrow{E}$ weakly terminating then $E_{\mathcal{B}} = \mathbf{A2E}(\mathbf{E2A}(\overrightarrow{E}))$ is such that $=_E \supseteq =_{E_{\mathcal{B}}}$.*

Proof. Assume that $s =_{E_{\mathcal{B}}} t$. From Lemma 23, we get that there exists a state q in $\mathcal{B}$ such that $s \to^*_{\mathcal{B}} q$ and $t \to^*_{\mathcal{B}} q$. Then, with Lemma 16, we get that $s =_E t$. ◀

Now we can state the generalized termination theorem for tree automata completion.

▶ **Theorem 29** (Generalized termination theorem for completion)**.** *Let E be a set of equations such that $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite and $=_E$ is decidable (resp. $IRR(\overrightarrow{E})$ is finite and $\overrightarrow{E}$ weakly terminating). For all $\not{\epsilon}$-reduced tree automaton $\mathcal{A}$ and TRS $\mathcal{R}$, completion of $\mathcal{A}$ with $\mathcal{R}$ and $\mathbf{A2E}(\mathbf{MN}(E))$ (resp. $\mathbf{A2E}(\mathbf{E2A}(\overrightarrow{E}))$) terminates.*

Proof. Let $\mathcal{B} = \mathbf{MN}(E)$ (resp. $\mathcal{B} = \mathbf{E2A}(\overrightarrow{E})$). Using Lemma 15 (resp. Lemma 14), we know that $\mathcal{B}$ is **RDFC**. Let $E_{\mathcal{B}}$ be the set of equations $ATE(\mathcal{B})$. Using Theorem 26, we know that completion of $\mathcal{A}$ with $\mathcal{R}$ and $E_{\mathcal{B}}$ is terminating. ◀

The above theorem shows how to tune a set of equations E into $E_{\mathcal{B}}$ to guarantee termination of completion. Note that tuning E into $E_{\mathcal{B}}$ does not jeopardize the precision of the completion since Lemma 28 guarantees that $=_E \supseteq =_{E_{\mathcal{B}}}$. This Lemma combined with the upper-bound Theorem 8 ensure that completion with $E_{\mathcal{B}}$ can only be more precise than completion with E. In the next section, we improve the precision theorem itself.

6 Improving the Precision of Equational completion

If we look back to our overall goal, we are half way. If $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular then it can be recognized by an automaton $\mathcal{B}$ and, using the results of the last section, we can build a set of equations $E_{\mathcal{B}}$ guaranteeing termination of completion. What remains to be proved is that completion with $E_{\mathcal{B}}$ ends on a tree automaton recognizing exactly $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. As it is, the precision Theorem 8 fails to tackle this goal because it needs $\mathcal{R}/E$-coherence $\mathcal{A}$. However, if $\mathcal{A}$ is not $\mathcal{R}/E$-coherent the full precision, granted by this theorem, cannot be obtained.

▶ **Example 30.** We can start again from Example 10 with set of equations $E_{\mathcal{B}}$ of Example 21. The initial tree automaton is not $\mathcal{R}/E_{\mathcal{B}}$-coherent (nor $\mathcal{R}/E$-coherent): $a \to^{\not{\epsilon}*}_{\mathcal{A}} q_1$ and $c \to^{\not{\epsilon}*}_{\mathcal{A}} q_1$ though $a \neq_{E_{\mathcal{B}}} c$. As a consequence, if we complete $\mathcal{A}$ with $\mathcal{R}$ and $E_{\mathcal{B}}$, we obtain an automaton that approximates $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ more roughly than expected. In particular, this automaton recognizes the term c that is not reachable by rewriting the initial language $\mathcal{L}(\mathcal{A}) = \{f(a), f(c)\}$ with $\mathcal{R}$ (nor by rewriting with $\mathcal{R}/E_{\mathcal{B}}$). The completed automaton can be obtained using the Timbuk tool [14]:

States `q0 q1` **Final States** `q0` **Transitions** `c->q1 a->q1 c->q0 f(q0)->q0 f(q1)->q0 a->q0`

When E is an empty set of equations, it is possible to transform $\mathcal{R}$ and $\mathcal{A}$ to have a $\mathcal{R}/E$-coherent initial completion setting [13]. However, such a transformation is not usable, in general, when $E \neq \emptyset$. Here, for a given E possibly not empty, we propose to transform $\mathcal{A}$ so that it becomes $\mathcal{R}/E$-coherent: we build the product between $\mathcal{A}$ and either $\mathbf{MN}(E)$ or $\mathbf{E2A}(\overrightarrow{E})$. We recall the definition of product automata and we show that the product is $\mathcal{R}/E$-coherent.

▶ **Definition 31** (Product automaton [7])**.** Let $\mathcal{A} = (\mathcal{F}, Q, Q_F, \Delta_{\mathcal{A}})$ and $\mathcal{B} = (\mathcal{F}, P, P_F, \Delta_{\mathcal{B}})$ be two automata. The product automaton of $\mathcal{A}$ and $\mathcal{B}$ is $\mathcal{A} \times \mathcal{B} = (\mathcal{F}, Q \times P, Q_F \times P_F, \Delta)$ where $\Delta = \{f((q_1, p_1), \ldots, (q_k, p_k)) \to (q', p') \mid f(q_1, \ldots, q_k) \to q' \in \Delta_{\mathcal{A}}$ and $f(p_1, \ldots, p_k) \to p' \in \Delta_{\mathcal{B}}\}$.

▶ **Theorem 32** (Generalized upper-bound). *Let $\mathcal{R}$ be a left-linear TRS, $\mathcal{A}$ an epsilon free tree automaton and E a set of equations such that $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite and $\mathcal{D} = \mathbf{MN}(E)$ exists (resp. $IRR(\overrightarrow{E})$ is finite and $\mathcal{D} = \mathbf{E2A}(\overrightarrow{E})$ exists). If $\mathcal{B} = \mathcal{A} \times \mathcal{D}$ then for any $i \in \mathbb{N}$:*

$$\mathcal{L}(\mathcal{B}^i_{\mathcal{R},E}) \subseteq \mathcal{R}^*_E(\mathcal{L}(\mathcal{A}))$$

Proof. Since $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}\times\mathcal{D}) = \mathcal{L}(\mathcal{A})\cap\mathcal{L}(\mathcal{D})$ and $\mathcal{L}(\mathcal{D}) = \mathcal{T}(\mathcal{F})$, we get that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$. For $\mathcal{R}/E$-coherence of $\mathcal{B}$, since both $\mathcal{A}$ and $\mathcal{D}$ are epsilon free, so is $\mathcal{B}$. Thus, to prove $\mathcal{R}/E$-coherence of $\mathcal{B}$, we only have to prove that for all states q of $\mathcal{B}$ and for all two terms $s, t \in \mathcal{T}(\mathcal{F})$ such that (1) $s \to^{\not\epsilon *}_{\mathcal{B}} q$ and (2) $t \to^{\not\epsilon *}_{\mathcal{B}} q$ then $s =_E t$. Since $\mathcal{B}$ is a product automaton, q is a pair of the form (q_1, q_2) and from (1) and (2) we can deduce that $s \to^{\not\epsilon *}_{\mathcal{D}} q_2$ and $t \to^{\not\epsilon *}_{\mathcal{D}} q_2$. Then, using Lemma 11 (resp. Lemma 16), we thus have $s =_E t$. Then, since $\mathcal{B}$ is $\mathcal{R}/E$-coherent, from Theorem 8, we get that $\mathcal{L}(\mathcal{B}^i_{\mathcal{R},E}) \subseteq \mathcal{R}^*_E(\mathcal{L}(\mathcal{B}))$. The fact that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$ concludes the proof. ◀

▶ **Example 33.** Back to Example 30, we can build the product between $\mathcal{A}$ and the automaton $\mathcal{B}$ found in Example 13. In $\mathcal{A}\times\mathcal{B}$, a and c are recognized by two different states, avoiding the $\mathcal{R}/E$-coherence problem of Example 30. Using Timbuk we can build the $\not\epsilon$reduced product $\mathcal{A} \times \mathcal{B}$, where product states are renamed by new state labels:

States `q2 q1 q0` **Final States** `q2` **Transitions** `c->q0 a->q1 f(q0)->q2 f(q1)->q2`

Then, if we complete $\mathcal{A} \times \mathcal{B}$ with $\mathcal{R}$ and $E_{\mathcal{B}}$, we obtain a completed automaton whose precision is now bounded by $\mathcal{R}^*_{E_{\mathcal{B}}}(\mathcal{L}(\mathcal{A}))$ and do not recognize c in a final state:

States `q0 q1 q2` **Final States** `q0` **Transitions** `a->q1 f(q0)->q0 f(q1)->q0 f(q2)->q0 a->q0 c->q2`

Now, we thus have hints to define a set of equations E to complete an automaton. For instance, it is possible to start from an automaton $\mathcal{D}$ defining a rough approximation of the target language and build $E = \mathbf{A2E}(\mathcal{D})$. Then, we complete $\mathcal{B} = \mathcal{A} \times \mathcal{D}$ with $\mathcal{R}$ and E and obtain a tree automaton $\mathcal{B}^*_{\mathcal{R},E}$ whose precision is better or equal to $\mathcal{D}$. The set $\mathcal{R}^*_E(\mathcal{L}(\mathcal{A}))$ only acts as a safeguard for completion (See Figure 1). In particular, terms of $\mathcal{R}^*_E(\mathcal{L}(\mathcal{A}))$ may not belong to $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E})$. This is the case in Example 33, where the term b belongs to $\mathcal{R}^*_{E_{\mathcal{B}}}(\mathcal{L}(\mathcal{A}))$ but not to $\mathcal{L}(\mathcal{B}^*_{\mathcal{R},E})$.

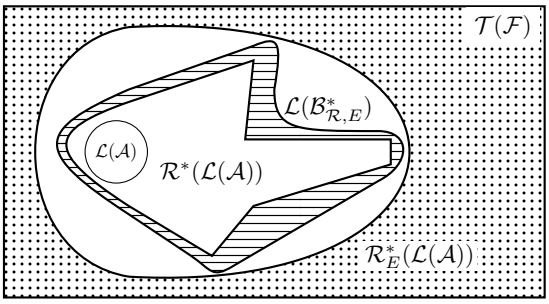


Figure 1 The generalized upper-bound theorem (precision of completion)

In fact, there are practical settings where defining the target language (or at least a subset) is possible. This is the case for TRS encoding terminating and complete functional programs, where the set of normal forms consists only of terms built on constructor symbols. By choosing an automaton $\mathcal{B}$ recognizing the language of constructor terms, we can generate $E_{\mathcal{B}} = \mathbf{E2A}(\mathcal{B})$ and easily generate the remainder of the equation set because it is fixed by the theory [13].

7 Completion covers all the left-linear TRS classes preserving regularity

In this section, we take advantage of the above theorems (lower-bound, generalized upper-bound and generalized termination) to prove that, for left-linear TRS, completion can compute the set of reachable terms if it is regular. As a particular case, this shows that completion is complete w.r.t. *all left-linear TRS classes preserving regularity*. Since the upper-bound of completion depends on $\mathcal{R}/E^*$, we first need an intermediate lemma showing that if E corresponds to the language $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$, then $\mathcal{R}/E^*$ is equal to $\mathcal{R}^*$.

▶ **Lemma 34.** *Let $\mathcal{R}$ be a TRS on $\mathcal{F}$, $S \subseteq \mathcal{T}(\mathcal{F})$, and $\mathcal{B}$ a* **RDFC** *automaton such that $\mathcal{L}(\mathcal{B}) = \mathcal{R}^*(S)$. If $E_\mathcal{B} =$ **A2E**$(\mathcal{B})$ then $(\mathcal{R}/E_\mathcal{B})^*(S) = \mathcal{R}^*(S)$.*

Proof. The inclusion $(\mathcal{R}/E_\mathcal{B})^*(S) \supseteq \mathcal{R}^*(S)$ trivially holds. Let us prove the opposite direction: $(\mathcal{R}/E_\mathcal{B})^*(S) \subseteq \mathcal{R}^*(S)$. We make a proof by contradiction. Assume that there exist terms $s, t \in \mathcal{T}(\mathcal{F})$ such that $s \in \mathcal{R}^*(S)$, $s =_{E_\mathcal{B}} t$, $t \in (\mathcal{R}/E_\mathcal{B})^*(S)$ but $t \notin \mathcal{R}^*(S)$. Using Lemma 23 on $s =_{E_\mathcal{B}} t$, we get that there exists a state q of $\mathcal{B}$ such that $s \to^*_\mathcal{B} q$ and $t \to^*_\mathcal{B} q$. Since $s \in \mathcal{R}^*(S)$, then $s \in \mathcal{L}(\mathcal{B})$, *i.e.* there exists a final state q_f of $\mathcal{B}$ such that $s \to^*_\mathcal{B} q_f$. Besides, we know that $\mathcal{B}$ is deterministic and thus $q = q_f$. As a result, $t \to^*_\mathcal{B} q_f$ with q_f final state, which implies that $t \in \mathcal{L}(\mathcal{B}) = \mathcal{R}^*(S)$ which is a contradiction. ◀

▶ **Theorem 35.** *Let $\mathcal{A}$ be a reduced epsilon-free tree automaton and $\mathcal{R}$ a left-linear TRS. If $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular then it is possible to compute a tree automaton recognizing $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ by equational tree automata completion.*

Proof. Assuming that $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ is regular, we know that there exists a **RDFC** tree automaton, say $\mathcal{B}$, recognizing $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. From $\mathcal{B}$ we can infer $E_\mathcal{B} =$ **A2E**$(\mathcal{B})$ and then use completion to compute reachable terms. From Theorem 26, we know that completion of automaton $\mathcal{A}$ with $\mathcal{R}$ and the set of equations $E_\mathcal{B}$ always terminates on a tree automaton $\mathcal{A}^*_{\mathcal{R},E_\mathcal{B}}$. What remains to be proved is that $\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E_\mathcal{B}}) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. From Theorem 5, we know that $\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E_\mathcal{B}}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. From Theorem 8, we know that $\mathcal{L}(\mathcal{A}^*_{\mathcal{R},E_\mathcal{B}}) \subseteq (\mathcal{R}/E_\mathcal{B})^*(\mathcal{L}(\mathcal{A}))$ provided that $\mathcal{A}$ is $\mathcal{R}/E_\mathcal{B}$-coherent. To enforce $\mathcal{R}/E_\mathcal{B}$-coherence of $\mathcal{A}$, we apply the transformation presented in Section 6. Let $\mathcal{A}^T = \mathcal{A} \times$ **MN**$(E_\mathcal{B})$. Note that since $E_\mathcal{B}$ is obtained by using the **A2E** transformation, $\mathcal{T}(\mathcal{F})/_{=_E}$ is finite (Corollary 24) and since equations of $E_\mathcal{B}$ are ground, $=_{E_\mathcal{B}}$ is decidable using a congruence closure algorithm. The resulting automaton $\mathcal{A}^T$ is $\mathcal{R}/E_\mathcal{B}$-coherent. Besides, Theorem 26 also applies on $\mathcal{A}^T$. Hence, completion of $\mathcal{A}^T$ with $\mathcal{R}$ and $E_\mathcal{B}$ always ends on an automaton $\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}$ which satisfies $\mathcal{L}(\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}^T))$ and $\mathcal{L}(\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}) \subseteq (\mathcal{R}/E_\mathcal{B})^*(\mathcal{L}(\mathcal{A}^T))$. Since $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^T)$, we thus have $\mathcal{L}(\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}) \supseteq \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$ and $\mathcal{L}(\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}) \subseteq (\mathcal{R}/E_\mathcal{B})^*(\mathcal{L}(\mathcal{A}))$. With Lemma 34, we get that $(\mathcal{R}/E_\mathcal{B})^*(\mathcal{L}(\mathcal{A})) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. Finally $\mathcal{L}(\mathcal{A}^{T,*}_{\mathcal{R},E_\mathcal{B}}) = \mathcal{R}^*(\mathcal{L}(\mathcal{A}))$. ◀

8 Conclusion and perspectives

Completion was known to cover many TRS classes preserving regularity [11, 13]. For some other classes, such as the linear subclass of [9], the question was still opened. Now, we know that for those known classes (and for those not known yet), given $\mathcal{A}$ and $\mathcal{R}$, there exists a set of equations E such that $\mathcal{A}^*_{\mathcal{R},E}$ recognizes the set of reachable terms. An open question is whether it is possible to infer automatically E from $\mathcal{A}$ and $\mathcal{R}$. As mentioned in the end of Section 6, we are currently working on this topic in the particular case of functional programs, for which the strong restrictions on the possible sets of equations to

consider make inference possible. Another question is whether we can extend this result to the case of innermost reachable terms using completion proposed in [16] that enjoys a precision theorem similar to the one used here. In particular, it is possible to get rid of the $\mathcal{R}/E$-coherence assumption of this theorem as we have done here.

On a practical point of view, this paper removes two strong technical restrictions of the termination and precision theorems. This widely enlarges the application scope of those theorems and open new ways in computing or approximating reachable terms for termination proof or software verification. For static analysis of functional programs, having a non $\mathcal{R}/E$-coherent initial tree automaton lead to bad precision of the analysis (see [13]). Using a transformed initial automaton (Theorem 32) dramatically improves the precision of those analysis.

A perspective is to extend the coverage result of completion to non left-linear TRSs. Dealing with regular languages and non left-linear rules is known to be more challenging than the left-linear case [22, 3, 10]. Nevertheless, there could be a surprise here. For non left-linear TRSs, completion is known to be correct and complete as long as the completed tree automaton is kept deterministic [11]. Completion itself does not preserve determinism. However, in this paper, all the completed automata are $\not\epsilon$deterministic. This is an easy consequence of the fact that $E_{\mathcal{B}} = \mathbf{A2E}(\mathcal{B})$ contains equations of the form $s = s$ for all state representative s, and all states recognize at least one state representative. To remove epsilon transitions and have, thus, deterministic completed automata, it would be enough to add a set $E_{\mathcal{R}} = \{l = r \mid l \to r \in \mathcal{R}\}$ to $E_{\mathcal{B}}$. What remains to be proved is that precision results of [11] on (possibly) non left-linear TRSs can be lifted to equational completion and that adding $E_{\mathcal{R}}$ to $E_{\mathcal{B}}$ does not add approximation to $\mathcal{A}^*_{\mathcal{R},E}$ w.r.t. $\mathcal{R}^*(\mathcal{L}(\mathcal{A}))$.

Acknowledgments

I would like to thank Sophie Tison for discussions about finiteness of $\mathcal{T}(\mathcal{F})/_{=_E}$.

References

1 F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
2 Y. Boichut, B. Boyer, T. Genet, and A. Legay. Equational Abstraction Refinement for Certified Tree Regular Model Checking. In *ICFEM'12*, volume 7635 of *LNCS*. Springer, 2012.
3 Y. Boichut, R. Courbis, P.-C. Héam, and O. Kouchnarenko. Handling non left-linear rules when completing tree automata. *IJFCS*, 20(5), 2009.
4 Y. Boichut, T. Genet, T. Jensen, and L. Leroux. Rewriting Approximations for Fast Prototyping of Static Analyzers. In *RTA'07*, volume 4533 of *LNCS*, pages 48–62. Springer, 2007.
5 Y. Boichut, P.-C. Héam, and O. Kouchnarenko. Automatic Approximation for the Verification of Cryptographic Protocols. In *Proc. AVIS'2004, joint to ETAPS'04, Barcelona (Spain)*, 2004.
6 H. Comon. Sequentiality, Monadic Second-Order Logic and Tree Automata. *Inf. Comput.*, 157(1-2):25–51, 2000.
7 H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, C. Löding, S. Tison, and M. Tommasi. Tree automata techniques and applications. `http://tata.gforge.inria.fr`, 2008.
8 H. Comon and Jean-Luc Rémy. How to characterize the language of ground normal forms. Technical Report 676, INRIA-Lorraine, 1987.
9 I. Durand and M. Sylvestre. Left-linear bounded trss are inverse recognizability preserving. In *RTA'11*, volume 10 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.
10 B. Felgenhauer and R. Thiemann. Reachability Analysis with State-Compatible Automata. In *LATA'14*, volume 8370 of *LNCS*, pages 347–359. Springer, 2014.

11 G. Feuillade, T. Genet, and V. Viet Triem Tong. Reachability Analysis over Term Rewriting Systems. *Journal of Automated Reasonning*, 33 (3-4):341–383, 2004.

12 T. Genet. Decidable Approximations of Sets of Descendants and Sets of Normal Forms. In *RTA'98*, volume 1379 of *LNCS*, pages 151–165. Springer, 1998.

13 T. Genet. Termination Criteria for Tree Automata Completion. *Journal of Logical and Algebraic Methods in Programming*, 85, Issue 1, Part 1:3–33, 2016.

14 T. Genet, Y. Boichut, B. Boyer, V. Murat, and Y. Salmon. Reachability Analysis and Tree Automata Calculations. IRISA / Université de Rennes 1. `http://www.irisa.fr/celtique/genet/timbuk/`.

15 T. Genet and R. Rusu. Equational tree automata completion. *Journal of Symbolic Computation*, 45:574–597, 2010.

16 T. Genet and Y. Salmon. Reachability Analysis of Innermost Rewriting. In *RTA'15*, volume 36 of *LIPIcs*, Warshaw, 2015. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

17 A. Geser, D. Hofbauer, J. Waldmann, and H. Zantema. On tree automata that certify termination of left-linear term rewriting systems. In *RTA'05*, volume 3467 of *LNCS*, pages 353–367. Springer, 2005.

18 F. Jacquemard. Decidable approximations of term rewriting systems. In H. Ganzinger, editor, *Proc. 7th RTA Conf., New Brunswick (New Jersey, USA)*, pages 362–376. Springer-Verlag, 1996.

19 Dexter Kozen. On the Myhill-Nerode theorem for trees. *Bull. Europ. Assoc. Theor. Comput. Sci.*, 47:170–173, June 1992.

20 A. Middeldorp. Approximations for strategies and termination. *ENTCS*, 70(6):1–20, 2002.

21 T. Takai. A Verification Technique Using Term Rewriting Systems and Abstract Interpretation. In *RTA'04*, volume 3091 of *LNCS*, pages 119–133. Springer, 2004.

22 T. Takai, Y. Kaji, and H. Seki. Right-linear finite-path overlapping term rewriting systems effectively preserve recognizability. In *RTA'11*, volume 1833 of *LNCS*. Springer, 2000.

23 S. Tison. Finiteness of the set of E-equivalence classes is undecidable, 2010. Private communication.