



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Closure of Tree Automata Languages under Innermost Rewriting

Citation for published version:

Gascón, A, Godoy, G & Jacquemard, F 2009, 'Closure of Tree Automata Languages under Innermost Rewriting', *Electronic Notes in Theoretical Computer Science*, vol. 237, pp. 23-38.
<https://doi.org/10.1016/j.entcs.2009.03.033>

Digital Object Identifier (DOI):

[10.1016/j.entcs.2009.03.033](https://doi.org/10.1016/j.entcs.2009.03.033)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Publisher's PDF, also known as Version of record

Published In:

Electronic Notes in Theoretical Computer Science

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.





ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in
Theoretical Computer
Science

Electronic Notes in Theoretical Computer Science 237 (2009) 23–38

www.elsevier.com/locate/entcs

Closure of Tree Automata Languages under Innermost Rewriting

Adrià Gascón^{a,1}, Guillem Godoy^{a,1} and Florent Jacquemard^{b,1}

^a *Technical University of Catalonia, Jordi Girona 1, Barcelona, Spain.* ²

^b *INRIA Futurs & LSV, UMR CNRS/ENS Cachan, France.*

Abstract

Preservation of regularity by a term rewriting system (TRS) states that the set of reachable terms from a tree automata (TA) language (aka regular term set) is also a TA language. It is an important and useful property, and there have been many works on identifying classes of TRS ensuring it; unfortunately, regularity is not preserved for restricted classes of TRS like shallow TRS. Nevertheless, this property has not been studied for important strategies of rewriting like the innermost strategy – which corresponds to the *call by value* computation of programming languages.

We prove that the set of innermost-reachable terms from a TA language by a shallow TRS is not necessarily regular, but it can be recognized by a TA with equality and disequality constraints between brothers. As a consequence we conclude decidability of regularity of the reachable set of terms from a TA language by innermost rewriting and shallow TRS. This result is in contrast with plain (not necessarily innermost) rewriting for which we prove undecidability. We also show that, like for plain rewriting, innermost rewriting with linear and right-shallow TRS preserves regularity.

Keywords: Term Rewriting, Tree Automata, Rewrite Strategies.

Introduction

Finite representations of infinite sets of terms are useful in many areas of computer science. The choice of a formalism for this purpose depends on its expressiveness, but also on its computational properties. Finite-state Tree Automata (TA) [3] are a well studied formalism for representing term languages, due to their good computational and expressiveness properties. They are used in many fields of computer science, from a theoretical and a practical point of view. For instance, for the analysis of systems or programs, when configurations can be represented by trees (e.g. concurrent processes with parallel and sequential composition operators) TAs provide a finite representation of possibly infinite sets of configurations.

¹ adriagascon@gmail.com, ggodoy@lsi.upc.edu, florent.jacquemard@lsv.ens-cachan.fr

² The first two authors were supported by Spanish Min. of Educ. and Science by the FORMALISM project (TIN2007-66523) and by the LOGICTOOLS-2 project (TIN2007-68093-C02-01)

Term rewriting is a general formalism for the symbolic evaluation of terms by replacement of some patterns by others, following oriented equations, or rewrite rules, given in a finite set (a term rewriting system, or TRS). Plain rewriting is sometimes too general, and in many contexts rewriting is applied with specific strategies giving a finer representation of the system behaviour. This is the case of the innermost strategy, which corresponds to the *call by value* computation of programming languages, where arguments are fully evaluated before the application of the function.

In the above application to system verification, transitions in infinite state systems can usually be represented by rewrite rules. There have been many studies of the connections between TA and rewriting, and a central property in this domain is the *preservation of regularity*. It states that for any given regular language L (which means that L is accepted by a TA), the set of reachable terms from L by a TRS \mathcal{R} , denoted $\mathcal{R}^*(L)$ is also regular. Preservation of regularity has been widely studied. The first result of this kind was that preservation of regularity holds for every ground TRS, as shown in [17]. In [15] this property was established for linear (variables occur at most once in every left-hand and right-hand side of a rule) and right-flat (the right-hand sides of the rules have height 0 or 1) TRS. There have been several extensions of this result, e.g. [6,11,14,16,5], and [14] represents a breakthrough since the left-linearity condition (linearity of left-hand sides of rules of the TRS) was dropped. However, in all the above cases, the condition of right-linearity remains necessary and in fact, a rewrite rule like $g(x) \rightarrow f(x, x)$ does not preserve regularity. Moreover, only plain rewriting is considered in these works, except in [5] where the bottom-up strategy is considered; there have been (up to our knowledge) no studies of regularity preservation under the innermost strategy.

The aim of this work is to study the preservation of regularity for innermost rewriting, and to identify a class of TRS for which better results can be found under the innermost strategy than under plain rewriting. We consider the class of shallow (all variables occur at depth 0 or 1 in the terms of the rules) TRS. Although the shallow case seems restrictive, for plain rewriting, shallow TRS do not preserve regularity. Moreover, several interesting properties of TRS, like reachability, joinability, confluence [13] and termination [9], are undecidable for shallow TRS, while adding certain linearity restrictions allows the decidability of all these problems [14,16,10,9]. Hence, from a theoretical point of view, the shallow case draws a frontier for decidability when one considers classes of TRS defined by syntactic restrictions.

Our main result (Theorem 4.9, Section 4.2) is that, given a regular language L and a *shallow* TRS \mathcal{R} , the set $R^\wedge(L)$ of terms reachable from L using \mathcal{R} with the innermost strategy is recognized by tree automata extended with equality and disequality constraints between brothers in their state transitions. This kind of automata, which we call BTTA, was introduced in [2] as an extension of TA, and it has also good closure and decidability properties, but with worst complexity than standard TA. This is in contrast with the situation with plain rewriting: $\mathcal{R}^*(L)$ (the set of terms reachable from L using \mathcal{R} with plain rewriting) is in general neither a

TA nor a BTTA language under the same hypotheses (Proposition 3.2, Section 3).

One of the classical techniques for proving results of preservation of regularity consists of adding transitions to the automaton recognizing the starting language L , in order to simulate rule applications of \mathcal{R} and recognize also all the terms reachable from L . Apparently, this completion technique which works well for standard TAs (in all the regularity preservation results cited so far) does not work for general shallow TRS. Innermost rewriting cannot be simulated by TA transitions, despite it does operate almost in a bottom-up fashion for shallow TRS [5]. The reason follows from two other results of the paper:

- First, we show that innermost rewriting with flat TRS (TRS whose all left-hand-side and right-hand-sides of rules have depth at most one) does not preserve regularity (Proposition 4.2, Section 4). As a consequence, we need to consider BTTAs instead of standard TAs.
- Second, flat and linear TRS do neither preserve BTTA-recognizably (Proposition 4.3, Section 4.1). Consequently, TA completion cannot work in this case.

The main result is obtained in two steps. First, we reduce the problem of representing the reachable terms from a regular set to the reachable terms from a constant. Next, we give a direct construction of a BTTA recognizing the reachable terms from a constant. It is based on a representation of the set of reachable terms introduced in [8] using constrained terms. As an immediate consequence of the main result, we obtain from [1] that given a regular language L and a shallow TRS \mathcal{R} , it is decidable whether $\mathcal{R}^\lambda(L)$ is regular in the case of innermost rewriting. In contraposition, we prove undecidability of regularity of $\mathcal{R}^*(L)$ when plain rewriting is considered.

Another positive result (Theorem 5.2, Section 5.1) is that, like for plain rewriting, innermost rewriting with linear and right-shallow TRS preserves regular languages. This result has been independently obtained in [12]. In our case it is proved with a non trivial adaptation of the TA completion technique of e.g. [15,11]. The cases of plain and innermost rewriting are different in essence to treat, and some subtle differences need to be introduced. We show in particular that even though TA completion permits to establish that right-linear and right-flat TRS (*i.e.* when left-hand sides of rules might be not linear) preserve regular languages under plain rewriting, we show that this property is no longer true for under innermost rewriting (Proposition 5.3, Section 5.2).

A long version of this paper is available as a Research Report [7].

1 Preliminaries

We use standard notation from the term rewriting literature [4]. A *signature* Σ is a finite set of function symbols with arity. We write Σ_m for the subset of function symbols of Σ of arity m . Given an infinite set \mathcal{V} of variables, the set of terms built over Σ and \mathcal{V} is denoted $\mathcal{T}(\Sigma, \mathcal{V})$, and the subset of ground terms is denoted $\mathcal{T}(\Sigma)$. The set of variables occurring in a term $t \in \mathcal{T}(\Sigma, \mathcal{V})$ is denoted $vars(t)$. A

substitution σ is a mapping from \mathcal{V} to $\mathcal{T}(\Sigma, \mathcal{V})$. The application of a substitution σ to a term t is written $\sigma(t)$, and is the homomorphic extension of σ to $\mathcal{T}(\Sigma, \mathcal{V})$.

A term t is identified as usual to a function from its set of *positions* (strings of positive integers) $Pos(t)$ to symbols of \mathcal{F} and \mathcal{V} . We note Λ the empty string (root position). The length of a position p is denoted $|p|$ and also called *depth*. The *height* of a term t , denoted $h(t)$, is the maximum of $\{|p| \mid p \in Pos(t)\}$. A subterm of t at position p is written $t|_p$, and the replacement in t of the subterm at position p by u denoted $t[u]_p$.

A *term rewriting system* (TRS) over a signature Σ is a finite set of rewrite rules $\ell \rightarrow r$, where $\ell \in \mathcal{T}(\Sigma, \mathcal{V}) \setminus \mathcal{V}$ (it is called left-hand side of the rule) and $r \in \mathcal{T}(\Sigma, vars(\ell))$ (it is called right-hand side). A term $s \in \mathcal{T}(\Sigma, \mathcal{V})$ rewrites to t by a TRS \mathcal{R} at a position p of s with a substitution σ , denoted $s \xrightarrow{\mathcal{R}, p, \sigma} t$ (p and σ may be omitted in this notation) if there is a rewrite rule $\ell \rightarrow r \in \mathcal{R}$ such that $s|_p = \sigma(\ell)$ and $t = s[\sigma(r)]_p$. In this case, s is said to be *reducible*. The set of irreducible terms, also called \mathcal{R} -*normal-forms*, is denoted by $NF_{\mathcal{R}}$. The transitive and reflexive closure of $\xrightarrow{\mathcal{R}}$ is denoted $\xrightarrow{*}_{\mathcal{R}}$. Given $L \subseteq \mathcal{T}(\Sigma)$, we note $\mathcal{R}^*(L) = \{t \mid \exists s \in L, s \xrightarrow{*}_{\mathcal{R}} t\}$. The above rewrite step is called *innermost* if all proper subterms of $s|_p$ are \mathcal{R} -normal forms. In this case, we write $s \xrightarrow{i}_{\mathcal{R}} t$, and $\xrightarrow{\wedge}_{\mathcal{R}}$ for the the transitive and reflexive closure of this relation, and $\mathcal{R}^{\wedge}(L)$ for $\{t \mid \exists s \in L, s \xrightarrow{\wedge}_{\mathcal{R}} t\}$. We shall also use the notations $NF_{\mathcal{R}}^*(s)$ and $NF_{\mathcal{R}}^{\wedge}(s)$ (with $s \in \mathcal{T}(\mathcal{F}, \mathcal{V})$) for resp. $\mathcal{R}^*({s}) \cap NF_{\mathcal{R}}$ and $\mathcal{R}^{\wedge}({s}) \cap NF_{\mathcal{R}}$.

A TRS is called *linear* (resp. *right-linear*, *left-linear*) if every variable occurs at most once in each term (resp. right-hand side, left-hand side) of the rules. It is called *shallow* (resp. *right-shallow*, *left-shallow*) if variables occur at depth 0 or 1 in the terms (resp. in the right-hand sides, in the left-hand sides) of the rules and *flat* (resp. *right-flat*, *left-flat*) if the terms (resp. the right-hand sides, the left-hand sides) in the rules have height at most 1. A rule $\ell \rightarrow r$ is called *collapsing* if r is a variable.

2 Tree automata with constraints between brothers

A *tree automaton* (TA) \mathcal{A} on a signature Σ is a tuple (Q, Q^f, Δ) where Q is a finite set of nullary state symbols, disjoint from Σ , $Q^f \subseteq Q$ is the subset of final states and Δ is a set of ground rewrite rules of the form: $f(q_1, \dots, q_m) \rightarrow q$, or $q_1 \rightarrow q$ (ε -transition) where $f \in \Sigma_m$, and $q_1, \dots, q_m, q \in Q$ (q is called the *target* state of the rule).

A *Bogaert-Tison tree automaton* (BT TA, or tree automaton with constraints between brothers) is defined like a TA except that its states are unary and its transitions are constrained rewrite rules of the form $f(q_1(x_1), \dots, q_m(x_m)) \rightarrow q(f(x_1, \dots, x_m)) \llbracket c \rrbracket$, or ε -transitions $q_1(x_1) \rightarrow q(x_1)$, where x_1, \dots, x_m are distinct variables and the constraint c is a Boolean combination of equalities $x_i = x_j$. Equivalently, the constraint c can be defined as a partition P of $\{1, \dots, m\}$ with the same meaning as a conjunction of equalities $x_i = x_j$ for the indexes i, j such that $i \equiv_P j$, and disequalities $x_i \neq x_j$ for the indexes i, j such that $i \not\equiv_P j$. Follow-

ing the notations of [2,3], the above transitions are written $f(q_1, \dots, q_m) \xrightarrow{c} q$ (or $f(q_1, \dots, q_m) \rightarrow q$ when c is *true*) and $q_1 \rightarrow q$, and every equality $x_i = x_j$ (resp. disequality $x_i \neq x_j$) in the constraint c is written $i = j$ (resp. $i \neq j$). Note that every TA is the special case of BT TA whose constraints are all equal to *true*.

The language $L(\mathcal{A}, q)$ of a BT TA \mathcal{A} in state q is the set of ground terms *accepted* in state q by \mathcal{A} , i.e. the terms t such that $t \xrightarrow{\Delta^*} q(t)$. The language $L(\mathcal{A})$ of \mathcal{A} is $\bigcup_{q \in Q^f} L(\mathcal{A}, q)$ and a set of ground terms is called *regular* (resp. *BT-regular*) if it is the language of a TA (resp. BT TA).

Example 2.1 The following BT TA recognizes the set of *Bin* of complete binary trees over $\Sigma = \{a : 0, f : 2\}$, whose internal nodes are labeled by f and whose leaves are labeled by a : $(\{q\}, \{q\}, \{a \rightarrow q, f(q, q) \xrightarrow{1=2} q\})$. It is well known that *Bin* is not regular. \diamond

A BT TA \mathcal{A} is called *deterministic* (resp. *complete*) if for every term $t \in \mathcal{T}(\Sigma)$, there is at most (resp. at least) one state q such that $t \in L(\mathcal{A}, q)$. If \mathcal{A} is deterministic and complete, this unique state is denoted $\mathcal{A}(t)$. A BT TA \mathcal{A} is *normalized* if it does not contain ϵ -transitions, constraints in transitions are defined using partitions, and for every function symbol f with arity m , states q_1, \dots, q_m and partition P of $\{1, \dots, m\}$, \mathcal{A} contains exactly one rule of the form $f(q_1, \dots, q_m) \xrightarrow{P} q$. A normalized BT TA \mathcal{A} is deterministic and complete, and any BT TA can be transformed into a normalized one recognizing the same language. If \mathcal{A} is normalized, we write $\mathcal{A}(t, P)$, for a flat term $t \in \mathcal{T}(\Sigma \cup Q)$, to denote the unique state q such that $t \xrightarrow{P} q$ is a transition of \mathcal{A} . BT TA are useful for representing the set of normal forms of certain classes of TRSs, like flat TRSs, see e.g. [3].

Lemma 2.2 [3] *Let \mathcal{R} be a flat TRS over Σ . There exists a normalized BT TA $\mathcal{B} = (Q_{\mathcal{B}}, Q_{\mathcal{B}}^f, \Delta_{\mathcal{B}})$ on Σ which recognizes the set of ground \mathcal{R} -normal forms. Moreover $|Q_{\mathcal{B}} \setminus Q_{\mathcal{B}}^f| = 1$.*

Proof. The construction of $\mathcal{B} = (Q_{\mathcal{B}}, Q_{\mathcal{B}} \setminus \{q_{\text{reject}}\}, \Delta_{\mathcal{B}})$ on Σ is as follows. Its set of states $Q_{\mathcal{B}}$ is $\{q_c \mid c \in \Sigma_0\} \cup \{q, q_{\text{reject}}\}$ where all of them except q_{reject} are accepting states. Its set of rules $\Delta_{\mathcal{B}}$ contains:

- the rules $c \rightarrow q_c$ for every constant c that is a \mathcal{R} -normal form.
- the rules $c \rightarrow q_{\text{reject}}$ for every constant c that is not a \mathcal{R} -normal form.
- the rules $f(q_1, \dots, q_m) \xrightarrow{c} q_{\text{reject}}$ such that either some q_i is q_{reject} and c is *true*, or every q_i is different from q_{reject} and c is $\bigvee_{f(\ell_1, \dots, \ell_m) \rightarrow r \in \mathcal{R}, \forall 1 \leq i \leq m \ell_i \in \Sigma_0 \Rightarrow q_i = q_{\ell_i} \bigwedge_{1 \leq i < j \leq m \ell_i = \ell_j \in \mathcal{V}} i = j$.
- the rules $f(q_1 \dots q_m) \xrightarrow{c} q$ such that every q_i is different from q_{reject} and c is $\bigwedge_{f(\ell_1, \dots, \ell_m) \rightarrow r \in \mathcal{R}, \forall 1 \leq i \leq m \ell_i \in \Sigma_0 \Rightarrow q_i = q_{\ell_i} \bigvee_{1 \leq i < j \leq m \ell_i = \ell_j \in \mathcal{V}} i \neq j$.

\square

Example 2.3 Let Σ be the signature $\{a : 0, g : 1, f : 2\}$ and \mathcal{R} be the flat TRS $\{g(x) \rightarrow f(x, x)\}$. The BT TA \mathcal{B} recognizing the ground \mathcal{R} -normal forms is $\mathcal{B} = (\{q_a, q, q_{\text{reject}}\}, \{q_a, q\}, \Delta)$ with $\Delta = \{a \rightarrow q_a\} \cup \{g(q_*) \rightarrow q_{\text{reject}}, f(q_*, q_{\text{reject}}) \rightarrow q_{\text{reject}}, f(q_{\text{reject}}, q_*) \rightarrow q_{\text{reject}} \mid q_* = q_a, q, q_{\text{reject}}\} \cup \{f(q', q') \rightarrow q \mid q' = q_a, q\}$. \diamond

3 Closure under plain rewriting with shallow TRS

Right-(shallow and linear) TRSs preserve regularity [14]. It is well known that right-linearity cannot be omitted, as the following example shows.

Example 3.1 Let $\Sigma = \{a : 0, g : 1, f : 2\}$ and $\mathcal{R} := \{g(x) \rightarrow f(x, x)\}$ as in Example 2.3 and let L be the regular language $\{g^n(a) \mid n \geq 0\} = \{a, g(a), g(g(a)), \dots\}$. The set $\mathcal{R}^*(L)$ is not regular because its intersection with the regular set $\mathcal{T}(\{a, f\})$ is the non-regular set Bin of complete binary trees over $\Sigma = \{a : 0, f : 2\}$, and the class of regular tree languages is closed under intersection. \diamond

We show below that considering BTTA does not help in this case.

Proposition 3.2 *In general, $\mathcal{R}^*(L)$ is not BT-regular when L is a regular tree language and \mathcal{R} a flat TRS.*

Proof. Let us consider \mathcal{R} , L and Bin as in Example 3.1. The set $\mathcal{R}^*(L)$ is not BT-regular. Indeed, the intersection of $\mathcal{R}^*(L)$ with the regular (hence BT-regular) set $L_2 := \{f(s, t) \mid s \in \mathcal{T}(\{g, a\}), t \in \mathcal{T}(\{f, a\})\}$ is the subset L' of terms $f(s, t) \in L_2$ with $t \in Bin$ and $h(s) = h(t)$. This latter set is not BT-regular, as shown below. It follows that $\mathcal{R}^*(L)$ is not BT-regular because the class of BT-regular tree languages is closed under intersection [2].

Let us now show that L' is not BT-regular. Assume that it is recognized by a BTTA $\mathcal{A} = (Q, Q^f, \Delta)$ on Σ with n states, and for all $i \geq 1$ let $f(s_i, t_i)$ be the term of L' with $h(s) = h(t) = i$. For each i , there exists a reduction sequence $f(s_i, t_i) \xrightarrow{\Delta^*} q(f(s_i, t_i))$ with $q \in Q^f$, and we consider the last rule ρ_i of Δ applied in this reduction sequence. There exist two distinct indexes $i_1, i_2 \geq 1$ such that $\rho_{i_1} = \rho_{i_2}$. Let $f(q_1(x_1), q_2(x_2)) \xrightarrow{c} q(f(x_1, x_2))$ be this unique rule of Δ . Note that the constraint c does not contain the equality $x_1 = x_2$, actually c may be $x_1 \neq x_2$ or *true*. In both cases, it follows that $f(s_{i_1}, t_{i_2}) \xrightarrow{\Delta^*} f(q_1(s_{i_1}), q_2(t_{i_2})) \xrightarrow{\rho_{i_1}} q(f(s_{i_1}, t_{i_2}))$. This is contradiction with the fact that $f(s_{i_1}, t_{i_2}) \notin L'$ because $h(s_{i_1}) \neq h(t_{i_2})$. \square

4 Closure under innermost rewriting with shallow TRS

The essential problem in Proposition 3.2 relies on the fact that after an application of the rule $g(x) \rightarrow f(x, x)$ on a term $g(t)$, producing $f(t, t)$, the following application of rewrite rules can change the two occurrences of t in different ways, producing terms $f(t_1, t_2)$ with $t_1 \neq t_2$. The equality constraints of BTTA have not the expressive power to capture the relation relating t_1 and t_2 (i.e. that both are reachable from a common term). The situation is getting better when the innermost strategy is applied.

Example 4.1 In Example 3.1, when we apply the rule $g(x) \rightarrow f(x, x)$ to terms of $L = \{g^n(a) \mid n \geq 0\}$ with the innermost strategy, the subterm where it is applied must be $g(t)$ for a \mathcal{R} -normal form t . Hence, in the term $f(t, t)$ obtained, t cannot be modified by rewriting. Hence $\mathcal{R}^\wedge(L) = \{g^n(t) \mid t \in Bin, n \geq 0\}$. This set is

BT-regular; it is indeed recognized by the following BTTA:

$$(\{q, q_g\}, \{q, q_g\}, \{a \rightarrow q, f(q, q) \xrightarrow{1=2} q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_g\}).$$

◇

Note however that $\mathcal{R}^\wedge(L)$ is not regular in the above example.

Proposition 4.2 *In general, $\mathcal{R}^\wedge(L)$ is not regular when L is a regular tree language L and \mathcal{R} a flat TRS.*

4.1 Closure of BTTA languages with flat TRS

Linear and flat TRSs preserve regularity [15]. This result cannot be extended to BT-regularity, neither for plain nor innermost rewriting.

Proposition 4.3 *In general, $\mathcal{R}^*(L)$ and $\mathcal{R}^\wedge(L)$ are not BT-regular when L is BT-regular and \mathcal{R} is a flat and linear TRS.*

Proof. The tree language $L = \{h(f^n(0), f^n(0)) \mid n \geq 0\}$ is recognized by the following BTTA, with one equality constraint tested at the root position: $(\{q, q^f\}, \{q^f\}, \{0 \rightarrow q, f(q) \rightarrow q, h(q, q) \xrightarrow{1=2} q^f\})$. Note that L is not regular.

Let us consider the flat and linear TRS $\mathcal{R} = \{f(x) \rightarrow g(x)\}$ and the regular tree language $L' = \{h(f^n(0), g^m(0)) \mid n \geq 0\}$. The closure $\mathcal{R}^*(L) \cap L' = \{h(f^n(0), g^n(0))\}$ is not BT-recognizable, hence $\mathcal{R}^*(L)$ is neither BT-recognizable. This is also true if we consider innermost rewriting. □

4.2 Closure of TA languages with shallow TRS

The classical approach for proving preservation of regularity [11,14,16] consists in completing a TA recognizing the original language L with new rules inferred using \mathcal{R} . This method cannot be generalized to BT-regular languages, according to Proposition 4.3. Therefore, we follow a different approach.

We prove first that given a regular tree language L and a flat TRS \mathcal{R} on a signature Σ , we can generate a new TRS \mathcal{R}_c over an extended signature including a new constant c such that $\mathcal{R}_c^\wedge(\{c\})$ coincides with $\mathcal{R}^\wedge(L)$ on the given signature Σ . This simple and enabling result permits to represent the set of terms innermost-reachable from a regular term set as the set of terms innermost-reachable from a constant.

Later, we show how to compute a BTTA recognizing the terms innermost-reachable from a constant. To this end we make use of some results in [8] on innermost rewriting with shallow TRSs.

4.2.1 Simplifying assumptions on the signature and the TRS.

All the results of this section are concerned with shallow TRS over an arbitrary signature. In order to simplify the proof, we shall assume fixed from now on in this section a TRS \mathcal{R} over a signature Σ who following these two non restrictive assumptions:

- Σ contains several constant function symbols and only one non-constant symbol f of arity m ,
- the TRS \mathcal{R} is flat.

Such assumptions, already used *e.g.* in [8], can be made without loss of generality for the problem considered here, see [7] for details.

4.2.2 Reduction to terms innermost-reachable from constants.

Our goal is to reduce the effort of characterizing the set of terms innermost-reachable with \mathcal{R} from a regular language L to characterizing the set of terms innermost-reachable from a single constant. The idea is to add to the rewrite system \mathcal{R} the inverse of the transition rules of a TA \mathcal{A} recognizing L . We show then that the generation of the terms of L starting from the final states of \mathcal{A} and using the transitions of \mathcal{A} backward can be performed following the innermost strategy.

Lemma 4.4 *For every flat TRS \mathcal{R} and regular language L , over a signature Σ , there exists an extension $\Sigma' \supset \Sigma$, a constant $c \in \Sigma' \setminus \Sigma$ and a flat TRS \mathcal{R}_c over Σ' such that $\mathcal{R}_c^\wedge(\{c\}) \cap \mathcal{T}(\Sigma) = \mathcal{R}^\wedge(L)$.*

Proof. Let $\mathcal{A} = (Q, Q^f, \Delta)$ be a TA on Σ recognizing L . Without loss of generality, we assume that every state $q \in Q$ is the target of a rule of Δ . Let Σ' be $\Sigma \cup Q \cup \{c\}$, where c is a new constant not in $\Sigma \cup Q$ and let \mathcal{R}_c be $\mathcal{R} \cup \Delta^{-1} \cup \{c \rightarrow q \mid q \in Q^f\}$. We prove that $\mathcal{R}_c^\wedge(\{c\}) \cap \mathcal{T}(\Sigma) = \mathcal{R}^\wedge(L)$.

Direction \supseteq . Let $t \in \mathcal{R}^\wedge(L)$ and let $s \in L$ such that $s \xrightarrow{\wedge} \mathcal{R} t$. We have $s \xrightarrow{\wedge} \Delta q \in Q^f$. Therefore $c \xrightarrow{i} c \rightarrow q \xrightarrow{\Delta^{-1}} s \xrightarrow{\wedge} \mathcal{R} t$ and hence $c \xrightarrow{\wedge} \mathcal{R}_c t$.

Direction \subseteq . Let $t \in \mathcal{T}(\Sigma)$ be such that $c \xrightarrow{\wedge} \mathcal{R}_c t$. Since $c \notin \Sigma$, there is necessarily in this derivation at least one rewrite step of the form $c \xrightarrow{i} \mathcal{R}_c q$ for $q \in Q^f$. The subderivation $q \xrightarrow{\wedge} \mathcal{R}_c t$ can contain alternate rewrite steps using rules of Δ^{-1} or of \mathcal{R} . We want to show that these rewrite steps can be commuted such that a $(\Delta^{-1} \cup \mathcal{R})$ -innermost derivation of the form $q \xrightarrow{\Delta^{-1}}^* s \xrightarrow{\mathcal{R}}^* t$ is possible. To this end it is sufficient to see that any $(\Delta^{-1} \cup \mathcal{R})$ -innermost subderivation $u \xrightarrow{\mathcal{R}, p_1} v \xrightarrow{\Delta^{-1}, p_2} w$ can be commuted to a $(\Delta^{-1} \cup \mathcal{R})$ -innermost subderivation $u \xrightarrow{\Delta^{-1}, p_2} v' \xrightarrow{\mathcal{R}, p_1} w$, which will be straightforward if we prove that the positions p_1 and p_2 are disjoint. Note that the term $u|_{p_1}$ does not contain any symbol $q \in Q$: otherwise the rewrite step $u \xrightarrow{\mathcal{R}, p_1} v$ would not be $(\Delta^{-1} \cup \mathcal{R})$ -innermost since there exists a rule of the form $q \rightarrow r$ in Δ^{-1} according to our assumptions. Hence, $v|_{p_1}$ does not contain any symbol $q \in Q$ neither. Therefore, the rewrite step $v \xrightarrow{\Delta^{-1}, p_2} w$ is produced at a position p_2 disjoint with p_1 , and we are done. Hence, there exists a $(\Delta^{-1} \cup \mathcal{R})$ -innermost derivation $q \xrightarrow{\Delta^{-1}}^* s \xrightarrow{\mathcal{R}}^* t$. In order to prove that $t \in \mathcal{R}^\wedge(L)$ it suffices to see that $s \in L$. Since $s \xrightarrow{\wedge} \Delta q \in Q^f$, it is enough to prove that $s \in \mathcal{T}(\Sigma)$, i.e. that s does not contain any symbol $q \in Q$. Suppose that $s|_p$ is a certain $q \in Q$. Similarly as before, no rule in \mathcal{R} can be applied at a position $p' < p$. Otherwise this rewrite step would not be innermost. Hence, repeated applications of rules of \mathcal{R} under innermost rewriting do not remove q . Therefore, q is a symbol occurring

in t , a contradiction. \square

Note that this reduction works only when the innermost strategy is used. However, it is valid for any class of TRS closed under the addition of ground rules.

Example 4.5 Let $\Sigma = \{a : 0, g : 1, f : 2\}$, $\mathcal{R} := \{g(x) \rightarrow f(x, x)\}$ and $L = \{g^n(a) \mid n \geq 0\}$ as in Example 3.1. In order to fulfil the conditions of Lemma 4.4, we let $\Sigma' = \Sigma \uplus \{c\}$, and $\mathcal{R}_c = \mathcal{R} \cup \{c \rightarrow g(c), c \rightarrow a\}$.

The following BTTA \mathcal{B} is built following the construction in the proof of Lemma 4.4 for the recognition of the ground normal forms for this extended TRS. It extends the BTTA of Example 2.3. $\mathcal{B} = (\{q_a, q_c, q, q_{\text{reject}}\}, \{q_a, q_c, q\}, \Delta)$ with $\Delta = \{a \rightarrow q_a, c \rightarrow q_{\text{reject}}\} \cup \{g(q_*) \rightarrow q_{\text{reject}}, f(q_*, q_{\text{reject}}) \rightarrow q_{\text{reject}}, f(q_{\text{reject}}, q_*) \rightarrow q_{\text{reject}} \mid q_* = q_a, q_c, q, q_{\text{reject}}\} \cup \{f(q', q') \rightarrow q \mid q' = q_a, q_c, q\}$. Note that \mathcal{B} can be cleaned: all the transition rules with q_c in left hand side can be removed since no term is recognized in this state. \diamond

4.2.3 Weak normal forms and constrained terms.

From [8], we have the following definitions and results. A term t is a *weak normal form* if it is either a constant or a term of the form $t = f(t_1, \dots, t_m)$ such that every t_i is either a constant or a normal form.

Let $\mathcal{P}(\Sigma_0)$ denotes the powerset of Σ_0 minus the empty set. A *constraint* C is a partial function $C : \mathcal{V} \rightarrow \mathcal{P}(\Sigma_0)$ i.e. an assignment from variables to non-empty sets of constants. We say that a substitution σ is a *solution* of a constraint C (with respect to a TRS \mathcal{R}) if for all x in $\text{dom}(C)$, $\sigma(x) \in \text{NF}_{\mathcal{R}}^{\lambda}(C(x)) \setminus \Sigma_0$. A *constrained term* is a pair denoted $t|C$, where t is a flat term and C is a constraint, with $\text{dom}(C) = \text{vars}(t)$. A term $\sigma(t)$ is called an *instance* of $t|C$ if σ is a solution of C . Note that every instance of a constrained term is a weak normal form.

In [8] it is shown how to compute for every flat TRS \mathcal{R} and for every constant $c \in \Sigma_0$ two finite sets r_c and \bar{r}_c of flat constrained terms whose set of instances (resp. normal form of instances) contain exactly the weak normal forms (resp. non-constant normal forms) innermost-reachable from c . More precisely, r_c and \bar{r}_c satisfy the following properties.

- (a) for every $t|C \in r_c$, there exists at least one solution of C , and all instances of $t|C$ are innermost-reachable from c .
- (b) for every $t|C \in \bar{r}_c$, all non-constant normal form instances of $t|C$ are innermost-reachable from c .
- (c) for every weak normal form s innermost-reachable from c , there exists some constrained term $t|C \in r_c$ such that s is an instance of $t|C$.
- (d) for every non-constant normal form s innermost-reachable from c , there exists some constrained term $f(t_1, \dots, t_m)|C \in \bar{r}_c$ such that s is an instance of $f(t_1, \dots, t_m)|C$.

Example 4.6 Let us consider the TRS $\mathcal{R}_c = \{g(x) \rightarrow f(x, x), c \rightarrow g(c), c \rightarrow a\}$ of Example 4.5. We have $\text{NF}_{\mathcal{R}_c}^{\lambda}(c) = \text{Bin}$, the set of complete binary trees of $\mathcal{T}(\{a, f\})$.

The set of weak normal forms innermost-reachable from c is $\{c, a, g(c), g(a), f(c, c)\} \cup \{g(t), f(t, t) \mid t \in \text{Bin}\}$, and its subset of normal forms innermost-reachable from c is $\{a\} \cup \{f(t, t) \in \text{Bin}\}$. The following sets satisfy the above properties:

$$r_c = \{c|\emptyset, a|\emptyset, g(c)|\emptyset, g(a)|\emptyset, f(c, c)|\emptyset, f(a, a)|\emptyset, g(x)|\{x \mapsto \{c\}\}, f(x, x)|\{x \mapsto \{c\}\}\},$$

$$\bar{r}_c = \{f(a, a)|\emptyset, f(x, x)|\{x \mapsto \{c\}\}\}, \quad r_a = \{a|\emptyset\}, \quad \bar{r}_a = \emptyset \quad \diamond$$

4.2.4 Recognizing terms innermost-reachable from constants.

We assume some sets r_c and \bar{r}_c as above and we construct a normalized BTТА $\mathcal{A}_{\mathcal{R}}$ which recognizes the terms innermost-reachable from constants in Σ_0 using \mathcal{R} with the innermost strategy. For this purpose, we shall use the BTТА \mathcal{B} of Lemma 2.2 recognizing the ground normal forms of \mathcal{R} . Let $Q_0 = \{q \in Q_{\mathcal{B}} \mid \exists d \in \Sigma_0, B(d) = q\}$, and $Q_1 = Q_{\mathcal{B}}^f \setminus Q_0$. Without loss of generality we assume that only constants lead to states in Q_0 . Thus, the states of Q_1 characterize the set of non-constant \mathcal{R} -normal-form. The states of $\mathcal{A}_{\mathcal{R}}$ are pairs $\langle S, q \rangle$, where $S \subseteq \Sigma_0$ and $q \in Q_{\mathcal{B}}$. The intuitive idea is that a term t will lead to $\langle S, q \rangle$ with $\mathcal{A}_{\mathcal{R}}$ if it leads to q with \mathcal{B} and S is the set of all constants that innermost-reach t with \mathcal{R} . To this end, the set of transition rules contains:

- $b \rightarrow \langle \{d \mid b|\emptyset \in r_d\}, \mathcal{B}(b) \rangle$, for every constant b .
- $f(\langle S_1, q_1 \rangle, \dots, \langle S_m, q_m \rangle) \xrightarrow{P} \langle S, \mathcal{B}(f(q_1, \dots, q_m), P) \rangle$, for every $S_1, \dots, S_m \subseteq \Sigma_0$, $q_1, \dots, q_m \in Q_{\mathcal{B}}$, and partition P of $\{1, \dots, m\}$, and where S is the set of constants $c \in \Sigma_0$ such that there exists $f(\alpha_1, \dots, \alpha_m)|C \in (r_c \cup \bar{r}_c)$ with:
 - i. $\forall 1 \leq i \leq m$, if $\alpha_i \in \Sigma_0$ then $\alpha_i \in S_i$ and if $\alpha_i \in \mathcal{V}$ then $C(\alpha_i) \subseteq S_i$ and $q_i \in Q_1$,
 - ii. $\forall 1 \leq i < j \leq m$, if $\alpha_i = \alpha_j \in \mathcal{V}$ then $i \equiv_P j$ and if $f(\alpha_1, \dots, \alpha_m)|C \in \bar{r}_c \setminus r_c$ then $\mathcal{B}(f(q_1 \dots q_m), P) \in Q_1$ and every $\alpha_i \in \Sigma_0$ ($1 \leq i \leq m$) is in $\text{NF}_{\mathcal{R}}$.

By construction, the automaton $\mathcal{A}_{\mathcal{R}}$ is normalized.

Example 4.7 Let us come back to Examples 4.5 and 4.6. The BTТА constructed as above in this case has the following transitions rules (we forget several useless rules for the sake of clarity):

$$a \rightarrow \langle \{a, c\}, q_a \rangle, \quad c \rightarrow \langle \{c\}, q_{\text{reject}} \rangle,$$

$$f(\langle \{a, c\}, q' \rangle, \langle \{a, c\}, q' \rangle) \xrightarrow{1=2} \langle \{c\}, q \rangle \quad (q' = q_a, q),$$

$$f(\langle \{c\}, q' \rangle, \langle \{c\}, q' \rangle) \xrightarrow{1=2} \langle \{c\}, q \rangle,$$

$$g(\langle \{a, c\}, q_* \rangle) \rightarrow \langle \{c\}, q_{\text{reject}} \rangle, \quad g(\langle \{c\}, q_* \rangle) \rightarrow \langle \{c\}, q_{\text{reject}} \rangle \quad (q_* = q_a, q, q_{\text{reject}}). \quad \diamond$$

The following lemma states the correctness of the construction of the BTТА $\mathcal{A}_{\mathcal{R}}$.

Lemma 4.8 For all $t \in \mathcal{T}(\Sigma)$ $\mathcal{A}_{\mathcal{R}}(t) = \langle \{d \in \Sigma_0 \mid d \xrightarrow{\mathcal{R}} t\}, \mathcal{B}(t) \rangle$.

Proof. Let $\mathcal{A}_{\mathcal{R}}(t) = \langle S, q \rangle$. It is straightforward by construction that $\mathcal{B}(t) = q$. We prove $S = \{d \in \Sigma_0 \mid d \xrightarrow{\mathcal{R}} t\}$ by induction on the size of t .

We first consider the case where t is a constant. By definition of $\mathcal{A}_{\mathcal{R}}$, $S = \{d \in \Sigma_0 \mid (t|\emptyset) \in r_d\}$. It suffices to see that the conditions $d \xrightarrow{\hat{\mathcal{R}}} t$ and $(t|\emptyset) \in r_d$ are equivalent when t is a constant. This is a consequence of conditions (a) and (c) above.

Now, assume that t is of the form $f(t_1, \dots, t_m)$. Let $\langle S_1, q_1 \rangle = \mathcal{A}_{\mathcal{R}}(t_1), \dots, \langle S_m, q_m \rangle = \mathcal{A}_{\mathcal{R}}(t_m)$. By induction hypothesis, $\mathcal{B}(t_i) = q_i$ and $S_i = \{d \in \Sigma_0 \mid d \xrightarrow{\hat{\mathcal{R}}} t_i\}$, for every $i \in \{1, \dots, m\}$. Let $f(\langle S_1, q_1 \rangle, \dots, \langle S_m, q_m \rangle) \xrightarrow{P} \langle S, q \rangle$ be the rule fired in the last applied transition of $\mathcal{A}_{\mathcal{R}}(t)$. Then, it holds that $i \equiv_P j$ iff $t_i = t_j$. We prove the two inclusions of $S = \{d \in \Sigma_0 \mid d \xrightarrow{\hat{\mathcal{R}}} t\}$ separately.

Direction \subseteq . Let $c \in S$. By construction of $\mathcal{A}_{\mathcal{R}}$ there exists a constrained term $f(\alpha_1, \dots, \alpha_m)|C \in (r_c \cup \bar{r}_c)$ for which the above conditions i and ii hold.

Let σ be a substitution of domain $\text{vars}(f(\alpha_1, \dots, \alpha_m))$ with $\sigma(\alpha_i) := t_i$. The substitution σ is well defined: if for different i and j we have $\alpha_i = \alpha_j \in \mathcal{V}$, by condition ii , it implies that $i \equiv_P j$ and then $t_i = t_j$. It holds that every such $\sigma(\alpha_i)$ is a non-constant normal form, because $\alpha_i \in \mathcal{V}$ implies that $q_i = \mathcal{B}(t_i) \in Q_1$ due to condition i . Moreover, $\sigma(\alpha_i)$ is reachable from $C(\alpha_i)$ because $\alpha_i \in \mathcal{V}$ implies $C(\alpha_i) \subseteq S_i$ from condition i and $S_i = \{d \in \Sigma_0 \mid d \xrightarrow{\hat{\mathcal{R}}} t_i\}$ by induction hypothesis.

Altogether, it follows that $\sigma(f(\alpha_1, \dots, \alpha_m))$ is an instance of $f(\alpha_1, \dots, \alpha_m)|C \in (r_c \cup \bar{r}_c)$. We know that either $f(\alpha_1, \dots, \alpha_m)|C \in r_c$ or $f(\alpha_1, \dots, \alpha_m)|C \in \bar{r}_c \setminus r_c$. On the one hand, if $f(\alpha_1, \dots, \alpha_m)|C \in r_c$ then, by condition a , $c \xrightarrow{\hat{\mathcal{R}}} \sigma(f(\alpha_1, \dots, \alpha_m))$. On the other hand, if $f(\alpha_1, \dots, \alpha_m)|C \in \bar{r}_c \setminus r_c$ then condition ii implies that $\mathcal{B}(f(q_1, \dots, q_m), P) \in Q_1$, and hence $q \in Q_1$ and t is a non-constant normal form. Moreover, the α_i 's that are constants are also normal forms. For every one of these constants α_i we know that $\alpha_i \in S_i$, and hence we also have $\alpha_i \xrightarrow{\hat{\mathcal{R}}} t_i$. But since this α_i is a normal form it follows that $\alpha_i = t_i$. This implies that $\sigma(f(\alpha_1, \dots, \alpha_m)) = t$, and hence, that t is a non-constant normal form that is an instance of $f(\alpha_1, \dots, \alpha_m)|C \in \bar{r}_c$, and by condition b , $c \xrightarrow{\hat{\mathcal{R}}} \sigma(f(\alpha_1, \dots, \alpha_m))$.

Hence it suffices to show that $\sigma(f(\alpha_1, \dots, \alpha_m)) \xrightarrow{\hat{\mathcal{R}}} t$ in order to conclude. By definition of σ , terms $\sigma(f(\alpha_1, \dots, \alpha_m))$ and t can only differ in the positions i such that α_i is a constant. But in such cases we know that $\alpha_i \in S_i$, and using $S_i = \{d \in \Sigma_0 \mid d \xrightarrow{\hat{\mathcal{R}}} t_i\}$ we obtain $\alpha_i \xrightarrow{\hat{\mathcal{R}}} t_i$. Hence, $\sigma(f(\alpha_1, \dots, \alpha_m)) \xrightarrow{\hat{\mathcal{R}}} t$ follows.

Direction \supseteq . Let c be such that $c \xrightarrow{\hat{\mathcal{R}}} t$. Since t is not a constant, the previous derivation can be written by making explicit the last rewrite step at position Λ as ($>\Lambda$ represents any position other than Λ):

$$c \xrightarrow{\hat{\mathcal{R}}} \xrightarrow{\mathcal{R}, \Lambda} f(s_1, \dots, s_m) \xrightarrow{\hat{\mathcal{R}}, >\Lambda} t = f(t_1, \dots, t_m)$$

Hence, there exist (sub-)derivations $s_i \xrightarrow{\hat{\mathcal{R}}} t_i$. The term $s = f(s_1, \dots, s_m)$ is a weak normal form, and hence, by condition c , there exists a constrained term $u|C \in r_c$ such that s is an instance of $u|C$. At this point, either there exists such a u of the form $f(\alpha_1, \dots, \alpha_m)$, or every u satisfying this condition is a variable. In the second case, s is necessarily a normal form, and hence, by condition d , there exists a constrained term $f(\alpha_1, \dots, \alpha_m)|C \in \bar{r}_c$ such that s is an instance of $f(\alpha_1, \dots, \alpha_m)|C$. For proving that $c \in S$, it suffices to show that the conditions i and ii hold.

If a certain α_i is a constant, then it coincides with s_i , which \mathcal{R} -reaches t_i . Since $S_i = \{d \in \Sigma_0 \mid d \xrightarrow{\lambda} t_i\}$, it necessarily contains α_i .

If a certain α_i is a variable, then s_i coincides with t_i and is a non-constant normal form reachable from $C(\alpha_i)$. Hence, $q_i = \mathcal{B}(t_i)$ is in Q_1 , and again since $S_i = \{d \in \Sigma_0 \mid d \xrightarrow{\lambda} t_i\}$, it necessarily includes $C(\alpha_i)$.

If $\alpha_i = \alpha_j \in \mathcal{V}$ then $s_i = s_j$ and since both are normal forms we also have $t_i = t_j$, from which $i \equiv_P j$ follows.

In the case where $f(\alpha_1, \dots, \alpha_m)|C$ belongs to $\bar{r}_c \setminus r_c$, $f(s_1, \dots, s_m)$ is a non-constant normal form. Therefore, $q = \mathcal{B}(f(q_1, \dots, q_m), P) \in Q_1$ and all the constants α_i are also normal forms. \square

Given a flat TRS \mathcal{R} and a regular set L , the BTTA $\mathcal{A}_{\mathcal{R}_c}$ (corresponding to the \mathcal{R}_c associated to \mathcal{R} as in Lemma 4.4), restricted to the signature Σ of \mathcal{R} , recognizes $\mathcal{R}^\lambda(L)$ by marking as accepting states the pairs $\langle S, q \rangle$ such that $c \in S$, according to Lemmas 4.4 and 4.8. Together with the simplifying assumptions on the TRS, this permits to conclude the proof of Theorem 4.9.

Theorem 4.9 $\mathcal{R}^\lambda(L)$ is BT-regular when L is regular and \mathcal{R} is shallow.

Ground reachability (is a given ground term t reachable from another given ground term s ?) and joinability (does there exist a term u reachable from two given ground terms s and t ?) are undecidable for flat TRSs [13]. They become decidable when the innermost strategy is applied [8]. We can express this property here as a corollary of Theorem 4.9.

Corollary 4.10 Ground reachability and joinability are decidable for innermost rewriting with shallow TRS.

Proof. When restricting to innermost rewriting, t is reachable from s iff $t \in \mathcal{R}^\lambda(\{s\})$. Since $\{s\}$ is a regular language when s is ground, $\mathcal{R}^\lambda(\{s\})$ is BT-regular by Theorem 4.9. Therefore ground reachability reduces to the membership problem for BTTA, which is decidable.

Similarly, s and t are joinable iff $\mathcal{R}^\lambda(\{s\}) \cap \mathcal{R}^\lambda(\{t\}) \neq \emptyset$. By Theorem 4.9 and closure of BT-languages under Boolean operations [2,3], we obtain a reduction of ground joinability to the emptiness problem for BTTA, which is also decidable. \square

In [1] the decidability of the regularity of a BTTA was shown. Combining this result with Theorem 4.9 we obtain the following corollary.

Corollary 4.11 Given a regular language L and a shallow TRS \mathcal{R} , it is decidable whether $\mathcal{R}^\lambda(L)$ is regular.

This result does not hold when we deal with plain rewriting. In [13] it has been proved that reachability is undecidable for flat TRSs by reduction of the Post correspondence problem into $0 \xrightarrow{*} 1$. We show below how to extend \mathcal{R} into \mathcal{R}_0 such that $\mathcal{R}_0^*(0)$ is regular iff $0 \xrightarrow{*} 1$.

Theorem 4.12 *Given a regular language L and a flat TRS \mathcal{R} , it is undecidable whether $\mathcal{R}^*(L)$ is regular.*

Proof. In [13] it is proved that reachability is undecidable for flat TRSs by reduction of a PCP instance P into a TRS R over a signature including $\{0, 1\}$ such that P has a solution iff $0 \xrightarrow{*}_{\mathcal{R}} 1$. The reduction in [13] also satisfies that if P has no solution, the 0 does not reach any term containing 1 nor any term containing 0 properly.

This reduction can be modified by adding new symbols $\{f, h, g, a, b, c\}$ to the current signature Σ , and adding two new sets of rules to R : $\mathcal{R}_1 = \{0 \rightarrow f(a, b), a \rightarrow g(a), b \rightarrow g(b), a \rightarrow c, b \rightarrow c, f(x, x) \rightarrow h(x, x)\}$ and \mathcal{R}_2 containing all the necessary rules for making $\mathcal{R}_2^*(1)$ to be $\mathcal{T}(\Sigma \cup \{f, h, a, b, c\})$. The rules of \mathcal{R} ensure $(\mathcal{R} \cup \mathcal{R}_1 \cup \mathcal{R}_2)^*(0)$ to be a non-regular language, unless $0 \xrightarrow{*}_{\mathcal{R}} 1$. Note that if P has solution, then $0 \xrightarrow{*}_{\mathcal{R}} 1$, and hence $(\mathcal{R} \cup \mathcal{R}_1 \cup \mathcal{R}_2)^*(0)$ is $\mathcal{T}(\Sigma \cup \{f, h, a, b, c\})$, which is regular. Otherwise, if P has no solution, then 0 does not reach any term containing 1, nor containing 0 properly, and hence $(\mathcal{R} \cup \mathcal{R}_1 \cup \mathcal{R}_2)^*(0) \cap \mathcal{T}(\{h, c\})$ is the set $\{h(g^n(c), g^n(c)) \mid n > 0\}$, which is not regular. \square

5 Innermost rewriting and right-shallow TRS

In this section, we study the closure of regular languages under innermost rewriting with TRS whose right-hand sides of rules are shallow. We show that regularity is preserved by innermost rewriting with linear right-shallow TRSs (Subsection 5.1), but not by innermost rewriting with right-(linear and flat) (non left-linear) TRSs (Subsection 5.2). The first result was also proved independently in [12].

5.1 TA languages and linear and right-shallow TRS

First, we observe that every right-shallow TRS \mathcal{R} can be transformed into a right-flat TRS \mathcal{R}' (on an extended signature) such that for all $s, t \in \mathcal{T}(\Sigma)$, $s \xrightarrow{\wedge}_{\mathcal{R}} t$ iff $s \xrightarrow{\wedge}_{\mathcal{R}'} t$. The idea is to add a new constant c_r and a rule $c_r \rightarrow r$ for every ground proper subterm r of a right-hand side of a rule of \mathcal{R} , and to replace r by c_r in all the right-hand sides of \mathcal{R} . This transformation preserves linearity and reachability between terms of the original signature.

Let $\mathcal{A} = (Q, Q^f, \Delta)$ be a deterministic and complete TA on Σ recognizing a tree language L , and let \mathcal{R} be a linear and right-flat TRS. For all $c \in \Sigma_0$ we denote as q_c the unique state of Q such that $c \rightarrow q_c \in \Delta$. We assume moreover *wlog* that $L(\mathcal{A}, q_c) = \{c\}$.

We construct a finite sequence of TA $\mathcal{A}_0, \mathcal{A}_1, \dots$ whose last element recognizes $\mathcal{R}^\wedge(L)$. The construction of the sequence is incremental. Every \mathcal{A}_{k+1} is obtained from \mathcal{A}_k by the addition of some new transitions, such that if some term s is recognized by \mathcal{A}_k and s rewrites (in one step of innermost rewriting) to t , then t is recognized by \mathcal{A}_{k+1} .

In order to restrict to innermost rewriting, we shall use a complete and deterministic TA $\mathcal{B} = (Q_{\mathcal{B}}, Q_{\mathcal{B}}^f, \Delta_{\mathcal{B}})$ (without ε -transitions) recognizing the ground \mathcal{R} -normal forms (see *e.g.* [3] for its construction). As in Lemma 2.2, we can assume that \mathcal{B} has only one non-accepting state q_{reject} . Let \mathcal{A}_0 be a TA recognizing $L(\mathcal{A})$: $\mathcal{A}_0 := (Q \times Q_{\mathcal{B}}, Q^f \times Q_{\mathcal{B}}, \Delta_0)$ where Δ_0 is the set of transitions $f(\langle q_1, q'_1 \rangle, \dots, \langle q_m, q'_m \rangle) \rightarrow \langle q, q' \rangle$ such that $f(q_1, \dots, q_m) \rightarrow q \in \Delta$ and $f(q'_1, \dots, q'_m) \rightarrow q' \in \Delta_{\mathcal{B}}$.

The addition of transition rules to \mathcal{A}_k , giving \mathcal{A}_{k+1} , is defined by the superposition of rules of \mathcal{R} into a sequence of transitions of Δ_k . More precisely, $\mathcal{A}_{k+1} \setminus \mathcal{A}_k$ contains all the transitions which can be constructed from a rewrite rule $\ell \rightarrow r$ of \mathcal{R} (we let $\ell = f(\ell_1, \dots, \ell_m)$) and a substitution θ of the variables of ℓ into states of $Q \times Q_{\mathcal{B}}^f$ whose accepted language wrt \mathcal{A}_k is not empty, such that: $\theta(\ell) \xrightarrow{\Delta_k^*} \langle q_0, q_{\text{reject}} \rangle$, and the last step of the above reduction is $f(\langle q_1, q'_1 \rangle, \dots, \langle q_m, q'_m \rangle) \xrightarrow{\Delta_k} \langle q_0, q_{\text{reject}} \rangle$ and for all $i \leq m$, $q'_i \neq q_{\text{reject}}$. There are two cases for the transitions of $\mathcal{A}_{k+1} \setminus \mathcal{A}_k$:

- case 1: r is a variable. In this case, $r \in \text{vars}(\ell)$. Let $\langle \bar{q}, \bar{q}' \rangle = \theta(r)$, we add the ε -transition $\langle \bar{q}, \bar{q}' \rangle \rightarrow \langle q_0, \bar{q}' \rangle$.
- case 2: $r = g(r_1, \dots, r_m)$. We add all the transitions $g(\langle \bar{q}_1, \bar{q}'_1 \rangle, \dots, \langle \bar{q}_m, \bar{q}'_m \rangle) \rightarrow \langle q_0, \bar{q}' \rangle$ such that $g(\bar{q}'_1, \dots, \bar{q}'_m) \rightarrow \bar{q}' \in \Delta_{\mathcal{B}}$ and for each $i \leq m$, if r_i is a variable then $\langle \bar{q}_i, \bar{q}'_i \rangle := \theta(r_i)$, otherwise, if r_i is a constant then \bar{q}_i is q_{r_i} and there is no restriction for \bar{q}'_i .

All the TAs have the same state set, hence the construction terminates with a fixpoint denoted \mathcal{A}^* . The number $|\mathcal{A}^*|$ of states of \mathcal{A}^* is at most $|\mathcal{A}| \times |\mathcal{B}|$, and the number of rules of \mathcal{A}^* is polynomial in the same measure³, if we assume as usual that the maximum arity m of a function symbol is fixed for the problem.

We show that $L(\mathcal{A}^*) = \mathcal{R}^\wedge(L(\mathcal{A}))$, more precisely, that for all $t \in \mathcal{T}(\Sigma)$, $t \in L(\mathcal{A}^*, \langle q, q' \rangle)$ iff $t \in L(\mathcal{B}, q')$ and there exists $s \in L(\mathcal{A}, q)$ such that $s \xrightarrow{\hat{\mathcal{R}}} t$. To this end we follow the principle of the proofs given *e.g.* in [15,11,14], but some technical difficulties appear when we try to replace a subterm by another subterm while preserving an execution with Δ^* . They are solved thanks to the following technical Lemma 5.1 (its proof can be found in [7]).

Lemma 5.1 *For all $t \in \mathcal{T}(\Sigma \cup Q_{\mathcal{A}^*})$, if $t[\langle q_0, q_{\text{reject}} \rangle]_p \xrightarrow{\Delta^*} \langle q, q_{\text{reject}} \rangle$ then, for all $q' \in Q_{\mathcal{B}}$, there exists $q'' \in Q_{\mathcal{B}}$ such that $t[\langle q_0, q' \rangle]_p \xrightarrow{\Delta^*} \langle q, q'' \rangle$.*

Theorem 5.2 *$\mathcal{R}^\wedge(L)$ is regular when L is regular and \mathcal{R} is linear and right-shallow.*

Proof. (*sketch*, the complete proof can be found in [7]). The if direction is proved by induction on the number of rewrite steps in $s \xrightarrow{\hat{\mathcal{R}}} t$, using Lemma 5.1.

The other direction is proved by an induction on the multiset associated to the derivation $t \xrightarrow{\Delta^*} \langle q, q' \rangle$ by mapping each transition rule ρ used to the least index i of the \mathcal{A}_i to which ρ belongs. \square

³ Note however that $|\mathcal{B}|$ can be exponential in the size of \mathcal{R} in the worst case.

5.2 Closure of TA languages with right-(linear and flat) TRS

When we drop the restriction that \mathcal{R} is left-linear in Theorem 5.2, we lose regularity preservation with innermost rewriting. This is in contrast with plain rewriting, since regularity is preserved for right-linear and right-shallow TRSs [14].

Proposition 5.3 *In general, $\mathcal{R}^\wedge(L)$ is not BT-regular when L is regular and \mathcal{R} is right-linear and right-flat.*

Proof. Let $L = \{f(f(a, a), c)\}$, and $\mathcal{R} = \{f(x, c) \rightarrow x, f(g(x), x) \rightarrow h(x), h(x) \rightarrow h(x), a \rightarrow g(a), a \rightarrow b\}$. The intersection of $\mathcal{R}^\wedge(L)$ with the language of all terms containing only the symbols f, g, b is the set $\{f(g^n(b), g^m(b)) \mid n \neq m + 1\}$, which is not BT-regular. \square

6 Conclusion and further work

We have covered much of the cases of closure of TA and BTTA languages by innermost rewriting, providing results for each case. The positive results are that the set of terms innermost-reachable from a regular language with a shallow TRS is BT-regular, and it is regular when the TRS is linear and right-shallow. Moreover, given a shallow TRS, regularity of the innermost-reachable terms from a regular language is decidable. Other consequences are the decidability of the problems of *ground reachability*, *ground joinability* and *regular tree model checking* (given two regular languages L_{init} and L_{bad} and the TRS \mathcal{R} , do we have $\mathcal{R}^\wedge(L_{\text{init}}) \cap L_{\text{bad}} = \emptyset$?) for innermost rewriting with TRS in the above classes.

As future work, it could be interesting to consider other variants of TA with more general or different constraints, and to consider other strategies of rewriting different from innermost.

References

- [1] B. Bogaert, F. Seynhaeve, and S. Tison. The recognizability problem for tree automata with comparisons between brothers. In *Second International Conference on Foundations of Software Science and Computation Structure, (FoSSaCS)*, volume 1578 of *Lecture Notes in Computer Science*, pages 150–164. Springer, March 1999.
- [2] B. Bogaert and S. Tison. Equality and disequality constraints on direct subterms in tree automata. In *International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 161–171, 1992.
- [3] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: <http://www.grappa.univ-lille3.fr/tata>, 2007.
- [4] N. Dershowitz and J. P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science (Vol. B: Formal Models and Semantics)*, pages 243–320, North-Holland, 1990.
- [5] I. Durand and G. Sénizergues. Bottom-up rewriting is inverse recognizability preserving. In *18th International Conference on Term Rewriting and Applications (RTA)*, volume 4533 of *LNCS*, pages 107–121. Springer, 2007.
- [6] R. Gilleron and S. Tison. Regular tree languages and rewrite systems. *Fundam. Inform.*, 24(1/2):157–174, 1995.
- [7] G. Godoy, A. Gascón, and F. Jacquemard. Closure of Tree Automata Languages under Innermost Rewriting. Laboratoire Spécification et Vérification, Research Report number 08-31, 2008. http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2008-30.pdf

- [8] G. Godoy and E. Huntingford. Innermost-reachability and innermost-joinability are decidable for shallow term rewrite systems. In *18th International Conference on Term Rewriting and Applications (RTA)*, volume 4533 of *LNCS*, pages 184–199, Springer, June 2007.
- [9] G. Godoy, E. Huntingford, and A. Tiwari. Termination of rewriting with right-flat rules. In *18th International Conference on Term Rewriting and Applications (RTA)*, volume 4533 of *LNCS*, pages 200–213, Springer, 2007.
- [10] G. Godoy and A. Tiwari. Confluence of shallow right-linear rewrite systems. In *19th International Workshop of Computer Science Logic (CSL)*, volume 3634 of *LNCS*, pages 541–556, Springer, 2005.
- [11] F. Jacquemard. Decidable approximations of term rewriting systems. In *7th International Conference on Rewriting Techniques and Applications (RTA)*, volume 1103 of *LNCS*, pages 362–376, Springer, 1996.
- [12] Yoshiharu Kojima and Masahiko Sakai. Innermost reachability and context sensitive reachability properties are decidable for linear right-shallow term rewriting systems. In *19th International Conference on Rewriting Techniques and Applications (RTA)*, volume 5117 of *LNCS*, pages 187–201. Springer, 2008.
- [13] I. Mitsuhashi, M. Oyamaguchi, and F. Jacquemard. The confluence problem for flat TRSs. In *8th International Conference on Artificial Intelligence and Symbolic Computation (AISC)*, volume 4120 of *LNAI*, pages 68–81. Springer, 2006.
- [14] T. Nagaya and Y. Toyama. Decidability for left-linear growing term rewriting systems. In *10th International Conference on Rewriting Techniques and Applications (RTA)*, volume 1631 of *LNCS*, pages 256–270. Springer, 1999.
- [15] K. Salomaa. Deterministic tree pushdown automata and monadic tree rewriting systems. *J. Comput. Syst. Sci.*, 37:367–394, 1988.
- [16] T. Takai, Y. Kaji, and H. Seki. Right-linear finite path overlapping term rewriting systems effectively preserve recognizability. In *11th International Conference on Rewriting Techniques and Applications (RTA)*, volume 1833 of *LNCS*, pages 246–260, 2000.
- [17] W.S. Brainerd. Tree generating regular systems. *Information and Control*, 4:217–231, 1969.