

Florent Jacquemard
Michael Rusinowitch

Closure of Hedge-Automata
Languages by Hedge Rewriting

Research Report LSV-08-05

October 2007

Laboratoire
Spécification
et
Vérification



CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Closure of Hedge-Automata Languages by Hedge Rewriting

Florent Jacquemard¹ and Michael Rusinowitch²

¹ INRIA Futurs & LSV, UMR CNRS, ENS Cachan, France
florent.jacquemard@lsv.ens-cachan.fr

² LORIA & INRIA Lorraine, UMR 7503, rusi@loria.fr

Abstract. We consider rewriting systems for unranked ordered terms, i.e. trees where the number of successors of a node is not determined by its label, and is not a priori bounded. The rewriting systems are defined such that variables in the rewrite rules can be substituted by hedges (sequences of terms) instead of just terms. Consequently, this notion of rewriting subsumes both standard term rewriting and word rewriting.

We investigate some preservation properties for two classes of languages of unranked ordered terms under this generalization of term rewriting. The considered classes include languages of hedge automata (HA) and some extension (called CF-HA) with context-free languages in transitions, instead of regular languages.

In particular, we show that the set of unranked terms reachable from a given HA language, using a so called inverse context-free rewrite system, is a HA language. The proof, based on a HA completion procedure, reuses and combines known techniques with non-trivial adaptations. Moreover, we prove, with different techniques, that the closure of CF-HA languages with respect to context-free rewrite systems, the symmetric case of the above rewrite systems, is a CF-HA language. As a consequence, the problems of ground reachability and regular hedge model checking are decidable in both cases. We give several counter examples showing that we cannot relax the restrictions.

1 Introduction

In many applications the system states can be modeled by words or trees, sets of configurations by word or tree languages and the transitions of the system can be represented by rewrite rules. In this setting verifying whether a system can enter a set of unsafe states can be expressed as a reachability problem. This approach to the analysis of infinite-state systems requires the computation of the closure of languages under rewrite rules or at least an over-approximation of this closure. Since the usually considered languages are regular the approach is called *regular model checking* [2, 1]. Regular model checking has been quite successful in protocol and hardware verification. For increasing the scope of regular model checking it is therefore important to be able to derive new classes of languages and rewrite systems such that the rewrite closure is computable.

Unranked trees as well as ordered sequences of unranked trees called *hedges* [14, 15, 5] are flexible structures that are quite appealing to represent XML documents where the number of nodes can be modified, for instance when these nodes correspond to database records. Unranked trees have also been employed to model multithreaded recursive program configurations where the number of parallel processes is unbounded [3, 19]. Hedge-automata (HA) are considered now as the natural model of automata for unranked trees. A hedge automaton is a variation of tree automata for hedges. Given a hedge, a hedge automata assigns some state to a node whenever the *sequence* of states of the siblings belong to some specified word language (sometimes called horizontal language).

Although regular model checking with languages for words and *ranked* trees (where function symbols have fixed arity) has been widely investigated, very few results are available for *unranked* trees and almost none exists on the *computation of exact reachability sets* for HA languages.

In this paper we tackle the problem above by proving (Theorem 1) that we can compute a HA for recognizing the rewrite closure of a language defined by a given HA, for the class of rewrite systems with inverse context-free rules, which are rules whose right-hand side is of type $f(x)$ where x is a variable. Hence in that case we can compute the exact reachability set from the initial one. The rewriting notion that we consider here for unranked terms generalizes ranked term rewriting and is close to the one that has been introduced by [23]. The idea is that the variables in the rewrite rules can be substituted by hedges (sequences of terms) instead of just terms. Moreover our results cannot be derived from related ones on ranked terms (*e.g.* [16]) using encodings of unranked terms into ranked ones (such as the *First-Child-Next-Sibling* encoding or the encoding used in stepwise automata [4]). Relaxing the condition in the definition in the above class of rewrite systems leads to counterexamples (Propositions 3–6).

We have also considered a more general class of automata for unranked ordered trees, called CF-HA, where word context-free languages are used instead of regular ones at the horizontal level. We show (Theorem 2) that CF-HA are preserved by rewrite closure using context-free rewrite rules. Context-free rewrite rules are the symmetric case inverse context-free rules, *i.e.* rules with left-hand-side of the form $f(x)$. Some additional restrictions are assumed for this result, they cannot be relaxed as show by the counter examples in Proposition 7–10.

Related works. Whether the rewrite closure of regular ranked trees languages is regular too is a problem that has been addressed in [20, 8, 10, 16, 22, 21, 6]. An important breakthrough of the proof in [16] (against former results) is that it works for TRS which are not left-linear. H. Ohsaki introduces equational tree automata for associative and commutative theories in [17] and study their closure properties for Boolean operations. T. Touili has studied the regular model checking problem for HA [23]. She shows how to compute the image of a HA language in one step of rewriting by a right-linear rewrite system. She also gives a procedure to compute an over-approximation of the rewrite closure of a HA. We rather compute exactly this closure for a class of non-linear rewrite systems.

Our first main result (Theorem 1) can be viewed as a non trivial generalization of both [16] and [23], with proof techniques extending both former constructions.

C. Löding and A. Spelten [12] compute exact rewrite closure of HA for extensions of ground term rewriting and prefix word rewriting. These results cannot be compared to ours since in our case variables (that can be substituted by arbitrarily large hedges) allow non local hedge transformations.

There exists other rewriting notions like the top-down XML transformations [13] or the relabeling transducers of [19] but they do not cover our notion since either they use specific hedge traversal strategies or they are structure-preserving.

Layout of the paper. In Section 2 we introduce terms, hedges and the related rewriting concepts. In particular we define hedge rewriting systems (HRS) and context-free rewrite rules. In Section 3 we recall the hedge-automata classes HA and CF-HA that we shall investigate. In Section 4 we show that the class of HA languages, (*i.e.* recognized by HA) is preserved by rewrite closure for rewriting systems containing rules that are either inverse context-free or right-linear and variable-disjoint. In Section 5 we show that a class of context-free hedge rewrite systems preserves CF-HA languages. In both Sections 4 and 5, we also exhibit some counter-examples obtained when trying to relax the conditions on rules.

2 Hedge Rewriting

We consider a finite alphabet Σ and an infinite set of variables \mathcal{X} . The set of *terms* over Σ and \mathcal{X} is $\mathcal{T}(\Sigma, \mathcal{X}) := \mathcal{X} \cup \{f(h) \mid f \in \Sigma, h \in \mathcal{H}(\Sigma, \mathcal{X})\}$ and the set $\mathcal{H}(\Sigma, \mathcal{X})$ of *hedges* over Σ and \mathcal{X} is the set of finite (possibly empty) sequences of terms of $\mathcal{T}(\Sigma, \mathcal{X})$. When h is empty, $f()$ will be simply written f . We will sometimes consider a term as a hedge of length one, *i.e.* consider that $\mathcal{T}(\Sigma, \mathcal{X}) \subset \mathcal{H}(\Sigma, \mathcal{X})$. The sets of ground terms (terms without variables) and ground hedges are respectively denoted $\mathcal{T}(\Sigma)$ and $\mathcal{H}(\Sigma)$. A hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$ is called *linear* if every variable of \mathcal{X} occurs at most once in h .

The set of variables occurring in a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is denoted $var(t)$. A *substitution* σ is a mapping from \mathcal{X} to $\mathcal{H}(\Sigma, \mathcal{X})$ of finite domain. The application of a substitution σ to a hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$, denoted $h\sigma$, is the homomorphic extension of σ to $\mathcal{H}(\Sigma, \mathcal{X})$, defined, for $t_1, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$, with $n \geq 0$, by $(t_1 \dots t_n)\sigma := t_1\sigma \dots t_n\sigma$ and $f(h)\sigma := f(h\sigma)$.

The set of *positions* $\mathcal{Pos}(h)$ of a hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$ is a set of sequences of positive integers. The empty sequence is denoted ε , it is the root position of a term. The subhedge of h at position p , denoted $t|_p$, is defined by $(t_1 \dots t_n)|_{i_p} := t_i|_p$ if $i \leq n$ and, for a term, $f(h)|_p = h|_p$ if $p \neq \varepsilon$ and $f(h)|_\varepsilon := f(h)$. The replacement in $h \in \mathcal{H}(\Sigma, \mathcal{X})$ of the subhedge at position p by $h' \in \mathcal{H}(\Sigma, \mathcal{X})$ is denoted $h[h']_p$. This notation can also be used to indicate that the subhedge of h at position p is h' .

The *depth* of a term is the maximal length of one of its positions. A *context* is a linear hedge of $\mathcal{H}(\Sigma, \{x\})$. The application of a context $C[x]_p$ to a hedge h is $C[h]_p$ (*i.e.* $C\{x \mapsto h\}$).

A hedge rewriting system (HRS) is a set of rewrite rules of the form $\ell \rightarrow r$ where $\ell, r \in \mathcal{T}(\Sigma, \mathcal{X})$ (ℓ and r are respectively called *lhs* and *rhs* of the rule). The rewrite relation $\xrightarrow{\mathcal{R}}$ of an HRS \mathcal{R} is the smallest binary relation on $\mathcal{H}(\Sigma, \mathcal{X})$ containing \mathcal{R} and closed by application of substitutions and contexts. In other words, $h \xrightarrow{\mathcal{R}}^* h'$ iff there exists a position $p \in \text{Pos}(h)$, a rule $\ell \rightarrow r \in \mathcal{R}$ and a substitution σ such that $h = h[\ell\sigma]_p$ and $h' = h[r\sigma]_p$. The transitive closure of $\xrightarrow{\mathcal{R}}$ is denoted $\xrightarrow{\mathcal{R}}^*$.

Example 1. With $\mathcal{R} = \{g(x) \rightarrow x\}$, $\xrightarrow{\mathcal{R}}$ associates to a term $g(h)$ the hedge h of its arguments. With $\mathcal{R} = \{g(x) \rightarrow g(axb)\}$, $g(c) \xrightarrow{\mathcal{R}}^* g(a^n cb^n)$ for every $n \geq 0$.

Given a set of terms $L \subseteq \mathcal{T}(\Sigma, \mathcal{X})$ and an HRS \mathcal{R} , we note $\mathcal{R}^*(L)$ the set $\{t \in \mathcal{T}(\Sigma, \mathcal{X}) \mid \exists s \in L, s \xrightarrow{\mathcal{R}}^* t\}$. We restrict to terms (instead of hedges) because we are mainly interested in term languages below.

A rewrite rule $\ell \rightarrow r$ is called *left-linear* (resp. *right-linear*, *linear*) if ℓ (resp. r , both) is linear, *left-ground* (resp. *right-ground*) if $\ell \in \mathcal{T}(\Sigma)$ (resp. $r \in \mathcal{T}(\Sigma)$), *collapsing* if $r \in \text{var}(\ell)$, *variable-disjoint* if $\text{var}(\ell) \cap \text{var}(r) = \emptyset$, it is called *context-free* if $\ell = f(x)$ with $x \in \text{var}(r)$ and *inverse context-free* if $r \rightarrow \ell$ is context-free, *prefix* (resp. *postfix*) if $r = g(t_0 \dots t_n x)$ (resp. $r = g(x t_0 \dots t_n)$) with $x \in \text{var}(\ell)$ and no variable of ℓ occurs in the terms t_0, \dots, t_n . A rewrite system is said to have one of the above properties if all its rules have this property.

Example 2. We give a few applications of our rewrite rules in the vein of [23]. A context-free rule $\text{doc}(x) \rightarrow \text{doc}(\langle a \rangle x \langle /a \rangle)$ can be employed to introduce tags in an XML document. An inverse context-free rule can be used to eliminate comments $\text{doc}(x \langle \text{comment} \rangle y \langle / \text{comment} \rangle) \rightarrow \text{doc}(x)$. Non left-linear inverse context-free rules are quite useful for processing list of items as in: $\text{doc}(\langle \text{todo} \rangle x \langle / \text{todo} \rangle y \langle \text{done} \rangle x \langle / \text{done} \rangle) \rightarrow \text{doc}(y)$.

Note that hedge rewriting cannot be reduced to term rewriting through encoding of unranked trees into ranked trees like the First-Child/Next-Sibling encoding, or the encoding used in stepwise automata. Consider for instance the hedge rewrite rule $f(axa) \rightarrow f(x)$. For every $n \geq 0$, $f(ab^n a)$ reduces in one step to $f(b^n)$, however there is no finite term rewrite system that can simulate such reductions in one step. Given a term t let us define inductively: $C_t^0 = t$ and $C_t^n = b(\perp, C_t^{n-1})$. Then $f(ab^n a)$ (resp. $f(b^n)$) is encoded by First-Child/Next-Sibling to $f(a(\perp, C_a^n), \perp)$ (resp. $f(C_\perp^n, \perp)$). Applying a term rewrite system to $f(a(\perp, C_a^n), \perp)$, for n large enough, an instance of a variable of a left-hand side of an applicable rule should contain some subterm C_a^m , with $m > 0$. This variable should also occur in the right-hand side to preserve the balance of b symbols. But then a should occur in the right-hand side too, contradiction.

3 Hedge-Automata, Context-Free Hedge-Automata

We recall now the definition of hedge-automata [14] (denoted HA) and the less known class of context-free hedge automata (denoted CF-HA) introduced in

[18] and where they are shown to recognize the closure of regular (ranked) tree languages modulo associativity.

A *hedge automaton* (resp. *context-free hedge automaton*) is a tuple $\mathcal{A} = (Q, \Sigma, Q^f, \Delta)$ where Q is a finite set of states, Σ is an unranked alphabet, $Q^f \subseteq Q$ is a set of final states, and Δ is a set of transitions of the form $f(L) \rightarrow q$ where $f \in \Sigma, q \in Q$ and $L \subseteq Q^*$ is a regular word language (resp. a context-free word language). When Σ is clear from the context it is omitted in the tuple specifying \mathcal{A} .

We define the move relation between ground hedges in $\mathcal{T}(\Sigma \cup Q)$ as follows: for every terms t, t' we have $t \xrightarrow{\mathcal{A}} t'$ if there exists a context $C[x]$ and a transition $f(L) \rightarrow q$ in Δ such that $t = C[f(q_1 \dots q_n)]$, $q_1 \dots q_n \in L$ and $t' = C[q]$. The relation $\xrightarrow{\mathcal{A}^*}$ is the transitive closure of $\xrightarrow{\mathcal{A}}$. Following [23], we extend $\xrightarrow{\mathcal{A}}$ to terms of $\mathcal{T}(\Sigma \cup 2^{Q^*})$ as follows: $C[f(L_1 \dots L_n)] \xrightarrow{\mathcal{A}} C[q]$ if there exists a rule $f(L) \rightarrow q$ in \mathcal{A} such that $L_1 \dots L_n \subseteq L$ (in this definition, a lone state q is considered as a singleton set $\{q\}$).

The language denoted by $L(\mathcal{A}, q)$ is the set of ground terms $t \in \mathcal{T}(\Sigma)$ such that $t \xrightarrow{\mathcal{A}^*} q$. A term is accepted by \mathcal{A} if there is $q \in Q^f$ such that $t \in L(\mathcal{A}, q)$. The language denoted by $L(\mathcal{A})$ is the set of terms accepted by \mathcal{A} .

It is known that for both classes of automata [14, 18] membership and emptiness problems are decidable. Moreover Hedge-Automata are closed under Boolean operations.

We shall assume below *wlog* that every HA or CF-HA $\mathcal{A} = (Q, Q^f, \Delta)$ is *normalized*: for every $f \in \Sigma$ and every $q \in Q$, there is at most one transition rule $f(L_{f,q}) \rightarrow q$ in Δ . Otherwise, we can replace any two rules $f(L_1) \rightarrow q$ and $f(L_2) \rightarrow q$ by $f(L_1 \cup L_2) \rightarrow q$.

A HA $\mathcal{A} = (Q, Q^f, \Delta)$ is called *deterministic* (resp. *complete*) if for all $t \in \mathcal{T}(\Sigma)$, there exists at most (resp. at least) one state $q \in Q$ such that $t \in L(\mathcal{A}, q)$. It is known (see *e.g.* [4]) that for every HA \mathcal{A} , there exists a deterministic and complete \mathcal{A}_d recognizing the same language. A determinisation procedure (with a subset construction) which preserve completeness is described in Section 4.1.

3.1 Epsilon- and Collapsing Transitions

We can extend HA and CF-HA with ε -*transitions* of the form $q \rightarrow q'$, where q and q' are states, without augmenting the respective expressiveness of these classes. We also consider the extensions of HA (resp. CF-HA), with *collapsing transitions* of the form $L \rightarrow q$ where L is a regular (resp. CF) language and q is a state. The move relation for the extended set of transitions is defined as for HA and CF-HA for standard transition by $C[f(q_1 \dots q_n)] \xrightarrow{\mathcal{A}} C[f(q_1 \dots q_i q q_{i+k+1} \dots q_n)]$ if $L \rightarrow q$ is a collapsing transition of \mathcal{A} and $q_{i+1} \dots q_{i+k} \in L$.

Unlike ε -transitions, collapsing transitions strictly extend HA in expressiveness. However, we show that they can be eliminated for CF-HA.

Proposition 1. *For every extended HA with collapsing transitions \mathcal{A} , there exists a CF-HA \mathcal{A}' (without collapsing transitions) such that $L(\mathcal{A}') = L(\mathcal{A})$. There*

exists an extended HA with collapsing transitions whose language is not a HA language.

Proof. Assume that $L \rightarrow q$ is a collapsing transition of \mathcal{A} . Then we get a CF-HA \mathcal{A}' such that $L(\mathcal{A}') = L(\mathcal{A})$ by replacing every transition $f(L_1) \rightarrow q_1$ by the transition $f(L_2) \rightarrow q_1$ where L_2 is the context-free word language generated by the grammar G_2 as follows: we consider a context-free grammar G for L (resp. G_1 for L_1) with axiom X (resp. X_1). The axiom of G_2 is X_1 and the set of production in G_2 contains *i*) $G[q \leftarrow X_q] \cup G_1[q \leftarrow X_q]$ *i.e.* the terminal q is replaced by a non terminal X_q and *ii*) we add to these rules the production: $X_q := q|X$. We can iterate this construction to eliminate all collapsing transitions.

Consider now the HA $\mathcal{A} = (\{q, q_a, q_b, q_f\}, \{g, a, b, c\}, \{q_f\}, \Delta)$ where

$$\Delta = \{c \rightarrow q, a \rightarrow q_a, b \rightarrow q_b, g(q) \rightarrow q_f, q_a q q_b \rightarrow q\}$$

The language recognized by this extended HA is $\{g(a^n c b^n) \mid n \geq 0\}$ which is not a HA language. \square

Proposition 2. *For every extended CF-HA with collapsing transitions \mathcal{A} , there exists a CF-HA \mathcal{A}' (without collapsing transitions) such that $L(\mathcal{A}') = L(\mathcal{A})$.*

Proof. The proof amounts to construct context-free grammars as in the proof of Proposition 1. \square

3.2 Decision Problems

The problem of *ground reachability* and *ground joinability* are to decide that, given two ground terms $s, t \in \mathcal{T}(\Sigma)$ and a HRS \mathcal{R} , whether, $s \xrightarrow{\mathcal{R}}^* t$, respectively, $s \xrightarrow{\mathcal{R}}^* \circ \xleftarrow{\mathcal{R}}^* t$.

Regular hedge model checking is the problem to decide, given two HA languages L_{init} and L_{err} and a HRS \mathcal{R} whether $\mathcal{R}^*(L_{\text{init}})$ contains a term of L_{err} .

Ground reachability is reducible to regular hedge model-checking. Indeed, given s, t and \mathcal{R} , $s \xrightarrow{\mathcal{R}}^* t$ iff $\mathcal{R}^*(\{s\}) \cap \{t\} \neq \emptyset$. Note also that if ground-reachability (hence regular hedge model-checking) are undecidable for a class of HRS, then $\mathcal{R}^*(L)$ is not recursive in general for \mathcal{R} in this class and L HA or CF-HA.

4 Closure of Regular Hedge Automata Languages

In this section, we prove one result of preservation of HA language for a class of HRS, and give several counter example showing that the restrictions defining this class of HRS are necessary.

4.1 Inverse Context-Free Rewrite Rules

Theorem 1. *The closure $\mathcal{R}^*(L)$ of a HA language $L \subseteq \mathcal{T}(\Sigma)$ under rewriting by a HRS \mathcal{R} whose rules are all either inverse context-free or right-linear and variable-disjoint, is a HA language.*

Note that the right-linear and variable-disjoint case includes right-ground rules.

Proof. Let $\mathcal{A}_L = (Q_L, Q_L^f, \Delta_L)$ be a complete HA recognizing L . We shall construct below a finite sequence of HA $(\mathcal{A}_i)_{i \geq 0}$ whose last element recognizes $\mathcal{R}^*(L)$. Our construction uses elements of [16] and [23], but it is not a simple combination of both. Indeed, on one side we generalize [23] to an unbounded number of rewriting steps, and on the other side we generalize [16] to unranked tree languages. Both generalizations are non-trivial and require new constructions and new conditions.

First, let us associate to each $r \in rhs(\mathcal{R})$, where $rhs(\mathcal{R})$ is the set of subterms of rhs of variable-disjoint rules in \mathcal{R} , a HA $\mathcal{A}_r = (Q_r, Q_r^f, \Delta_r)$ recognizing the set of all ground instances of r . We have one state $q_u \in Q_r$ for each non-variable subterm u of r , and an universal state q_\forall . The transition set Δ_r contains one rule $f(q_1 \dots q_n) \rightarrow q_{f(u_1 \dots u_n)}$ for each subterm $f(u_1 \dots u_n)$, where q_i is q_\forall if u_i is a variable and $q_i = q_{u_i}$ otherwise. For q_\forall , we have one transition rule $f(q_\forall^*) \rightarrow q_\forall$ in Δ_r for each $f \in \Sigma$. We assume that Q_L and all Q_r are pairwise disjoint. Let

$$\mathcal{A} := (Q, Q_L^f, \Delta) \text{ with } Q := Q_L \uplus \biguplus_{r \in rhs(\mathcal{R})} Q_r, \Delta := \Delta_L \uplus \biguplus_{r \in rhs(\mathcal{R})} \Delta_r$$

For each $f \in \Sigma$, $q \in Q$, we note $L_{f,q}$ the language in the transition (assumed unique) $f(L_{f,q}) \rightarrow q \in \Delta$.

Next, we construct a deterministic HA $\mathcal{A}_d = (Q_d, Q_d^f, \Delta_d)$ recognizing $L(\mathcal{A})$. The HA \mathcal{A}_d is obtained by a subset construction, see *e.g.* [4], with $Q_d := 2^Q$, $Q_d^f := \{s \in Q_d \mid s \cap Q_L^f \neq \emptyset\}$ and $\Delta_d := \{f(L_{f,s}) \rightarrow s \mid f \in \Sigma, s \subseteq Q\}$ where $L_{f,s} := (\bigcap_{q \in s} S_{f,q}) \setminus (\bigcup_{q \notin s} S_{f,q})$ and $S_{f,q} = \{s_1 \dots s_n \in Q_d^* \mid \exists q_1 \in s_1, \dots, q_n \in s_n, q_1 \dots q_n \in L_{f,q}\}$ ³.

Now, following the approach of [23], we define first the set of languages of Q_d^* that will be used in the transitions of the \mathcal{A}_i 's constructed below. However, we must consider here a bigger set than [23] in order to deal with non linear variables in lhs of rules. Let \mathcal{L} be the smallest set of subsets of Q_d^* such that

- i. every $L_{f,s}$ (for $f \in \Sigma$ and $s \in Q_d$), every $\{s_1 \dots s_k \mid \exists \ell \rightarrow g(r_1 \dots r_k) \in \mathcal{R}, \text{ variable-disjoint, s.t. } \forall i \leq k, s_i \cap Q_{r_i}^f \neq \emptyset\}$ and Q_d^* are in \mathcal{L} ,
- ii. if $L \in \mathcal{L}$ and $u, v \in Q_d^*$, then $u^{-1} L v^{-1} \in \mathcal{L}$, where

$$u^{-1} L v^{-1} := \{w \in Q_d^* \mid u w v \in L\},$$

- iii. if $L_1, L_2 \in \mathcal{L}$ then $L_1 \cap L_2 \in \mathcal{L}$,
- iv. if $L_1, L_2 \in \mathcal{L}$ then $L_1 \setminus L_2 \in \mathcal{L}$.

³ Note that $S_{f,q}$ and $L_{f,s}$ are indeed regular languages, see [4].

Note that the condition $Q_d^* \in \mathcal{L}$ in *i* together with *iii* and *iv* imply that \mathcal{L} is also closed under union (if $L_1, L_2 \in \mathcal{L}$ then $L_1 \cup L_2 \in \mathcal{L}$), by De Morgan's Law.

Let us show that \mathcal{L} is finite and that all its members are regular languages. First, let us note that \mathcal{L}_1 , the smallest set satisfying *i* and *ii* above, is a finite set of regular languages of Q_d^* , since every $L_{f,q}$ is regular by hypothesis. The closure \mathcal{L}_2 of \mathcal{L}_1 under *iii* and then *iv* is also a finite set of regular languages. The following lemma shows that \mathcal{L}_2 fulfills *ii*, i.e. that $\mathcal{L}_2 = \mathcal{L}$.

Lemma 1. For all $L_1, L_2 \subseteq Q_d^*$, $u_1, u_2, v_1, v_2, u, v \in Q^*$,
 $u^{-1}(u_1^{-1} L_1 v_1^{-1} \cap u_2^{-1} L_2 v_2^{-1}) v^{-1} = (u_1 u)^{-1} L_1 (v v_1)^{-1} \cap (u_2 u)^{-1} L_2 (v v_2)^{-1}$,
 $u^{-1}(u_1^{-1} L_1 v_1^{-1} \setminus u_2^{-1} L_2 v_2^{-1}) v^{-1} = (u_1 u)^{-1} L_1 (v v_1)^{-1} \setminus (u_2 u)^{-1} L_2 (v v_2)^{-1}$.

Proof. The set in the left-hand-side of the first identity in Lemma 1 is $A = \{\ell \mid ulv \in \{\ell' \mid u_1 \ell' v_1 \in L_1 \text{ and } u_2 \ell' v_2 \in L_2\}\}$, and the set in its right hand side is $B = \{\ell \mid u_1 ulv v_1 \in L_1 \text{ and } u_2 ulv v_2 \in L_2\}$. If $\ell \in A$, then $u_1 ulv v_1 \in L_1$ and $u_2 ulv v_2 \in L_2$, hence $\ell \in B$. Conversely, if $\ell \in B$, then $ulv \in u_1^{-1} L_1 v_1^{-1} \cap u_2^{-1} L_2 v_2^{-1}$, hence $\ell \in A$. The proof is very similar for the identity with the complementation. \square

Let us now construct $\mathcal{A}_0, \mathcal{A}_1, \dots$ as announced. The set of states and final states of each of these HA are respectively Q_d and Q_d^f . We give below a recursive construction of Δ_i , $i \geq 0$ which preserves the determinism and completeness.

Let $\Delta_0 = \Delta_d$. Assume that Δ_i has been constructed and contains one transition $f(L_{f,s}^i) \rightarrow s$ for every $f \in \Sigma$, $s \in Q_d$; Δ_{i+1} is obtained from Δ_i according to one of the following cases (non deterministic choice):

(icf) there exists an inverse context-free rewrite rule $\ell \rightarrow g(x) \in \mathcal{R}$, and a substitution $\tau : \text{var}(\ell) \rightarrow \{L' \in \mathcal{L} \mid \forall s_1 \dots s_k \in L', \forall j \leq k, L(\mathcal{A}_i, s_j) \neq \emptyset\}$, such that $\ell \tau \xrightarrow{\Delta_i^*} s' \in Q_d$. In this case, let $L' = x\tau$, Δ_{i+1} is obtained by replacing, for each $s \in Q_d$, the rule $g(L_{g,s}^i) \rightarrow s$ by the rules:

$$g(L_{g,s}^i \cap L') \rightarrow s \cup s' \text{ and } g(L_{g,s}^i \setminus L') \rightarrow s.$$

(vd) there exists a variable-disjoint rule $\ell \rightarrow g(r_1 \dots r_k) \in \mathcal{R}$, and a substitution τ as above. In this case, we let $L' = \{s_1 \dots s_k \in Q_d^* \mid \forall i \leq k, s_i \cap Q_{r_i}^f \neq \emptyset\}$ and construct Δ_{i+1} as in the case (icf) above.

Moreover, we assume that after each construction step, the set of transitions obtained is normalized. Note that all the languages in the above transitions belong to \mathcal{L} , according to the closure properties of this set.

This construction terminates (because no new state is added and \mathcal{L} is finite) with a HA \mathcal{A}_j denoted \mathcal{A}^* . Moreover, for all $i \geq 0$, every \mathcal{A}_i is deterministic and complete. We show by induction on i that for every $g \in \Sigma$, the sets $L_{g,s}^i$ are disjoint (it implies determinism of \mathcal{A}_i) and form a partition of Q_d^* . For the base case $i = 0$, it follows from the completeness of the initial HA \mathcal{A}_L and the determinisation (this operation preserves completeness). For the induction step, replacing $g(L_{g,s}^i) \rightarrow s$ by $g(L_{g,s}^i \cap L') \rightarrow s \cup s'$ and $g(L_{g,s}^i \setminus L') \rightarrow s$ as above preserves this property. We show now that $L(\mathcal{A}^*) = \mathcal{R}^*(L)$.

The proof of the direction $L(\mathcal{A}^*) \subseteq \mathcal{R}^*(L)$ relies on the following lifting lemma.

Lemma 2. For all $i \geq 0$, $t \in \mathcal{T}(\Sigma, \mathcal{X})$, $\sigma : \text{var}(t) \rightarrow \mathcal{H}(\Sigma)$, $\theta : \text{var}(t) \rightarrow Q_d^*$, if $t\theta \xrightarrow[\mathcal{A}_k]{*} s_0 \in Q_d$, and for all $x \in \text{var}(t)$, all component $(x\theta)|_j$ of $x\theta$ (state of Q_d) and $q \in (x\theta)|_j$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}]{*} (x\sigma)|_j$, then for all $q' \in s_0$, there exists $v \in L(\mathcal{A}, q')$ s. t. $v \xrightarrow[\mathcal{R}]{*} t\sigma$.

Note that in Lemma 2, for all $x \in \text{var}(t)$, $x\sigma$ and $x\theta$ have the same length.

Proof. We make an induction on i .

Base case ($i = 0$). We have $\mathcal{A}_0 = \mathcal{A}_d$ and since $t\sigma \xrightarrow[\mathcal{A}_d]{*} s_0$, by construction of \mathcal{A}_d , for all $q_0 \in s_0$, $t\sigma \xrightarrow[\mathcal{A}]{*} q_0$.

Induction step ($i + 1$). We assume that the property is true for i and prove the property for $i + 1$ by induction on the number n' of applications of a rule of $\Delta_{i+1} \setminus \Delta_i$ in the reduction $t\theta \xrightarrow[\mathcal{A}_{i+1}]{*} s_0$.

Base case ($n' = 0$). If there are no rules of $\Delta_{i+1} \setminus \Delta_i$ in the above reduction, then $t\theta \xrightarrow[\mathcal{A}_i]{*} s_0$ and we apply the induction hypothesis (on i).

Induction step ($n' + 1$). Let ρ be a rule in $\Delta_{i+1} \setminus \Delta_i$ applied at the position p of $t\theta$ in the above reduction sequence. Let $z \notin \text{var}(t)$ be a fresh variable and let $\sigma' = \{z \mapsto t\sigma|_p\}$. With this definition, $t\sigma = t[z]_p\sigma \cup \sigma'$. We have the reduction $t\sigma \xrightarrow[\mathcal{A}_{i+1}]{*} s_0$ described in the top line of Figure 1.

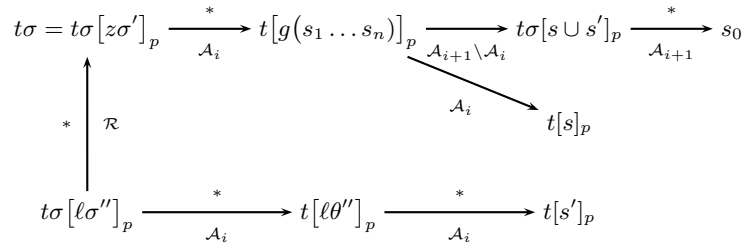


Fig. 1. Proof of Lemma 2, $L(\mathcal{A}^*) \subseteq \mathcal{R}^*(L)$.

Assume that $\rho = g(L_{g,s}^i \cap L') \rightarrow s \cup s'$ and that it has been added using the case (icf) of the construction with a inverse context-free rewrite rule $\ell \rightarrow g(z) \in \mathcal{R}$. Let $\tau : \text{var}(\ell) \rightarrow \mathcal{L}$ be the substitution used for the construction of the above rule, such that $\ell\tau \xrightarrow[\mathcal{A}_i]{*} s' \in Q_d$.

By hypothesis on τ , to each $y \in \text{var}(\ell)$ we can associate a hedge h which is reduced by \mathcal{A}_i to a sequence of states of Q_d in the language of $y\tau$. This permits to define two substitutions $\sigma'' : \text{var}(\ell) \rightarrow \mathcal{H}(\Sigma)$ and $\theta'' : \text{var}(\ell) \ni y \mapsto \bar{s} \in y\tau$, making possible the reduction (with \mathcal{A}_i) at the bottom line of Figure 1 and such that moreover $t[\ell\sigma'']_p \xrightarrow[\mathcal{R}]{*} t\sigma$. The later reduction is obtained by letting $z\sigma'' := z\sigma'|_1$.

By induction hypothesis, (on the number of applications of a rule of $\Delta_{i+1} \setminus \Delta_i$ in $\ell\theta'' \xrightarrow[\mathcal{A}_i]{*} s'$), for all $q' \in s'$, there exists $v \in L(\mathcal{A}, q')$ such that $v \xrightarrow[\mathcal{R}]{*} \ell\sigma''$. Moreover, by hypothesis on the rule ρ , the sequence of states $s_1 \dots s_n$ at position $p1$ in the reduction $t\sigma \xrightarrow[\mathcal{A}_{i+1}]{*} s_0$ (see Figure 1) belongs to $L_{g,s}^i$. Hence we have a reduction $z\sigma' \xrightarrow[\mathcal{A}_i]{*} s$ and by induction hypothesis, for all $q \in s$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}]{*} z\sigma'$. Altogether, for all $q \in s \cup s'$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}]{*} z\sigma'$. Letting $\theta' = \{z \mapsto s \cup s'\}$, we can apply the induction hypothesis to $t\sigma[z\theta']_p$, because there is at least one application of $\Delta_{i+1} \setminus \Delta_i$ less in $t\sigma[z\theta']_p \xrightarrow[\mathcal{A}_{i+1}]{*} s_0$ than in $t\theta_p \xrightarrow[\mathcal{A}_{i+1}]{*} s_0$. Hence, for all $q \in s'_0$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}]{*} t\sigma$.

The case $\rho = g(L_{f,s}^i \setminus L') \rightarrow s$ (added using (icf)) is simpler, because, in order to apply the induction hypothesis, we only need to show that for all $q \in s$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}]{*} z\sigma'$ in the first case. This can be done as before.

We use a similar reasoning for the case (vd).

(end of the proof of Lemma 2) \square

Now, for the particular case of Lemma 2 where $t \in \mathcal{T}(\Sigma)$, we have that if $t \xrightarrow[\mathcal{A}_i]{*} s_0$, for some i and $s_0 \in Q_d^f$, for all $q^f \in s_0$, where q^f is a final state of \mathcal{A} , there exists $u \in L(\mathcal{A}, q^f) \subseteq L(\mathcal{A})$ such that $u \xrightarrow[\mathcal{R}]{*} t$. This terminates the proof of the direction $L(\mathcal{A}^*) \subseteq \mathcal{R}^*(L)$.

For the direction $L(\mathcal{A}^*) \supseteq \mathcal{R}^*(L)$, assume that $t \in L(\mathcal{A})$ and that $t \xrightarrow[\mathcal{R}]{*} t'$. We show by induction on the length of this reduction sequence that $t' \in L(\mathcal{A}_i)$ for some i .

Base case ($t = t'$). It is immediate by construction of \mathcal{A}_0 .

Induction step. Assume that the last step in $t \xrightarrow[\mathcal{R}]{*} t'$ involves a inverse context-free rule $\ell \rightarrow g(x) \in \mathcal{R}$ at a position p and with a ground substitution $\sigma : \text{var}(\ell) \rightarrow \mathcal{H}(\Sigma)$, as described in the top line of the diagram in Figure 2. The case of a variable-disjoint rule is similar.

$$\begin{array}{ccc}
t \xrightarrow[\mathcal{R}]{*} t'[\ell\sigma]_p & \xrightarrow[\mathcal{R}]{} & t'[g(x\sigma)]_p = t' \\
* \downarrow \mathcal{A}_i \text{ (I.H.)} & & * \downarrow \mathcal{A}_i \\
t'[\ell\theta]_p & & t'[g(x\theta)]_p \\
\downarrow \mathcal{A}_i \text{ (I.H.)} & & \downarrow \mathcal{A}_{i+1} \setminus \mathcal{A}_i \\
t'[s']_p & & t'[s'']_p \\
\downarrow \mathcal{A}_i \text{ (I.H.)} & & * \downarrow \mathcal{A}_{i+1} \\
s^f & & s_1
\end{array}$$

Fig. 2. Proof of $\mathcal{R}^*(L) \subseteq L(\mathcal{A}^*)$.

By induction hypothesis, $t'[\ell\sigma]_p \in L(\mathcal{A}_i)$ for some i . Let us consider the reduction of this term to a final state s^f of \mathcal{A}_i depicted in the second column of the diagram in Figure 2. Since \mathcal{A}_i is deterministic, there exists a substitution $\theta : \text{var}(\ell) \rightarrow Q_d^*$ such that $t'[\ell\sigma]_p \xrightarrow{\mathcal{A}_i^*} t'[\ell\theta]_p$ and $t'[\ell\theta]_p \xrightarrow{\mathcal{A}_i} t'[s']_p \xrightarrow{\mathcal{A}_i^*} s^f$. Since \mathcal{A}_i is complete, there exists a substitution $\tau : \text{var}(\ell) \rightarrow \{L' \in \mathcal{L} \mid \forall s_1 \dots s_k \in L', \forall j \leq k, L(\mathcal{A}_i, s_j) \neq \emptyset\}$, such that for all $x \in \text{var}(\ell)$, $x\theta \in x\tau$. By definition of the relation $\xrightarrow{\mathcal{A}_i^*}$ extended to $\mathcal{T}(\Sigma, \mathcal{L})$, $\ell\tau \xrightarrow{\mathcal{A}_i^*} s'$.

Hence this τ is as in the case (icf) of the construction. It follows that a rule is added which permits the reduction $t'[g(x\theta)]_p \xrightarrow{\mathcal{A}_{i+1}^*} s''$ where $s'' = s \cup s'$ (the case $s'' = s$ is not possible because $x\theta \in x\tau$). We have a reduction from t' to a final state thank to the following technical lemma which state the monotonicity of the relation $\xrightarrow{\mathcal{A}_i^*}$ wrt state inclusion and context application.

Lemma 3. *For all $i \geq 0$, all context $C \in \mathcal{T}(\Sigma, \{x\})$ and $s_0, s'_0 \in Q_d$ such that $s_0 \subseteq s'_0$ if $C[s_0] \xrightarrow{\mathcal{A}_i^*} s \in Q_d$, then $C[s'_0] \xrightarrow{\mathcal{A}_i^*} s' \supseteq s$.*

Proof. The proof is an induction on the structure of C .

Base case ($C = x$). In this case, the result is immediate by hypothesis.

Induction step ($C = f(C_1 \dots C_n)$). We have $C[s_0] \xrightarrow{\mathcal{A}_i^*} f(s_1 \dots s_n) \xrightarrow{\mathcal{A}_i} s$. By induction hypothesis, for all $j \leq n$, $C_j[s'_0] \xrightarrow{\mathcal{A}_0^*} s'_j \supseteq s_j$. Since \mathcal{A}_i is deterministic and complete, there exists a unique s' such that $f(s'_1 \dots s'_n) \xrightarrow{\mathcal{A}_i} s'$. We show that $s' \subseteq s$ by induction on i .

Base case ($i = 0$). This case follows from the construction of \mathcal{A}_0 , with the determinisation procedure.

Induction step ($i = k + 1$). If $f(s'_1 \dots s'_n) \xrightarrow{\Delta_k} s'$, then we can conclude by induction hypothesis. If $f(s'_1 \dots s'_n) \xrightarrow{\Delta_{k+1} \setminus \Delta_k} s'$, then let s'' be the state of Q_d such that $s'_1 \dots s'_n \in L_{f, s''}^k$. This state exists and is unique because \mathcal{A}_k is deterministic and complete, and by induction hypothesis (on k), $s \subseteq s''$. An analyze of the two cases (icf) and (vd) which may have permit the construction the transition $f(L_{f, s'}^{k+1}) \rightarrow s' \in \Delta_{k+1} \setminus \Delta_k$ used above shows that this rule replaces $f(L_{f, s''}^k) \rightarrow s''$ and that, in both cases, $s'' \subseteq s'$. Hence $s \subseteq s'$. \square

With Lemma 3, we have $t' \xrightarrow{\mathcal{A}_{i+1}^*} s_1$ with $s_1 \supseteq s^f$. Hence $t' \in L(\mathcal{A}_{i+1})$.

(end of the proof of Theorem 1) \square

Corollary 1. *Ground reachability, ground joinability and regular hedge model-checking are decidable for HRS the rules of which are all either inverse context-free or right-linear and variable-disjoint.*

We present in the next subsections some counter examples showing that relaxing the assumption on \mathcal{R} in Theorem 1 invalidate the result.

4.2 Collapsing Rewrite Rules

Collapsing rules preserve regularity of term languages [16] when the function symbols are ranked. Indeed, in this case, if \mathcal{R} is left-linear and collapsing, a tree automaton (TA) recognizing L can be completed into a TA recognizing $\mathcal{R}^*(L)$ just by the iterated addition of ε -transitions of the form $x\theta \rightarrow q$ when there is $\ell \rightarrow x \in \mathcal{R}$ and a substitution $\theta : \text{var}(\ell) \rightarrow Q$ such that $\ell\theta \xrightarrow{\mathcal{A}}^* q$. When \mathcal{R} is just collapsing (not left-linear), the construction requires determinism and hence is more complicated but the idea is the same [16].

In the case of unranked terms and HA, if we want to follow the principles of the construction of Section 4.1, we need to add *collapsing transitions* and not just ε -transitions. But the addition of collapsing transitions does not preserve HA languages (Proposition 1). The following proposition shows that the above construction is actually not possible for collapsing rewrite rules.

Proposition 3. *$\mathcal{R}^*(L)$ is not a HA language in general when L is a HA language and \mathcal{R} is a left-linear HRS.*

Proof. We use the principle of the construction in the proof of Proposition 1. Let $\Sigma = \{f, g, a, b, c\}$, let L be the language of the HA

$$\mathcal{A} = (\{q, q_a, q_b, q_f\}, \{q_f\}, \{c \rightarrow q, a \rightarrow q_a, b \rightarrow q_b, g(q_a q_b) \rightarrow q, f(q) \rightarrow q_f\})$$

and let $\mathcal{R} = \{g(x) \rightarrow x\}$. Assume that $\mathcal{R}^*(L)$ is a HA language. Its intersection with the HA language $\{f(a^*cb^*)\}$ is $\{f(a^n cb^n) \mid n \geq 0\}$, which is not a HA language. This contradicts the fact that HA languages are closed under intersection. \square

Note that the completion of the above \mathcal{A} , following the procedure in the proof of Theorem 1, would add the collapsing transition $q_a q_b \rightarrow q$.

4.3 Flat Linear Rewrite Rules

In the case of ranked terms, it is known [16] that regularity of tree languages is preserved under rewriting with systems of right-linear rules of the form $\ell \rightarrow f(u_1, \dots, u_n)$ where f has arity n and each u_i ($i \leq n$) is either a ground term or a variable of $\text{var}(\ell)$. We call such a rule *flat* if its *lhs* and *rhs* both have depth one. Note that this class of TRS is not captured by the HRS of Theorem 1 (when restricted to ranked terms). The above regularity preservation result is non-longer true for unranked terms.

Proposition 4. *$\mathcal{R}^*(L)$ is not a HA language in general when L is a HA language and \mathcal{R} a context-free, linear and flat HRS. Moreover, it can be assumed that all the rules of \mathcal{R} are prefix or postfix.*

Proof. Let us consider the context-free HRS $\mathcal{R} = \{g(x) \rightarrow g(axb)\}$ of Example 1, and the HA language $L = \{g(c)\}$. $\mathcal{R}^*(L) = \{g(a^n cb^n) \mid n \geq 0\}$ and this language is not HA. We can transform the above \mathcal{R} into $\mathcal{R}' = \{g(x) \rightarrow g'(ax), g'(y) \rightarrow g'(yb)\}$ whose rules are prefix or postfix (and linear) and which is such that $\mathcal{R}'^*(L) \cap \mathcal{T}(\{g, a, b\}) = \mathcal{R}^*(L)$. \square

Note that the language in the above proof is recognized by a CF-HA. We shall show below (Theorem 2 in Section 5) that context-free HRS like the \mathcal{R} above preserve CF-HA languages.

We show now the stronger result that the closure of a HA language under rewriting with a flat HRS, even linear, is neither HA, nor CF-HA and actually not even recursive.

Proposition 5. *$\mathcal{R}^*(L)$ is not recursive in general when L is a HA language and \mathcal{R} a linear and flat HRS whose rules contain at most two variables.*

Proof. We reduce the blank accepting problem for TM to ground reachability for an HRS as in Proposition 5. Let \mathcal{M} be a TM with a tape alphabet Γ and a state set S and let $\Sigma = \Gamma \cup S \cup \{g\}$. A configuration of \mathcal{M} is represented by a term $g(w)$ where w is a word of $\Gamma^*Q\Gamma^*$ (the position of the state symbol indicates the position of the head of \mathcal{M} and the rest represents the contents of the tape). We assume, wlog unique blank initial and final configurations, respectively c_i and c_f . We consider a HRS \mathcal{R} containing one rule for each transition of \mathcal{M} . For instance, \mathcal{R} contains a rule $f(xaqy) \rightarrow f(xq'a'y)$ corresponding to a transition $s, a \rightarrow L, s', a'$ (with $s, s' \in S$ and $a, a' \in \Gamma$) and $f(xaqby) \rightarrow f(xa'by)$ to the transition $q, a \rightarrow R, q'$. The blank tape is accepted by \mathcal{M} iff $c_i \xrightarrow[\mathcal{R}]{} c_f$. \square

As a consequence, regular hedge model checking is undecidable for the HRS of Proposition 5, according to the remarks in Section 3.2.

4.4 Rewrite Rules with Flat and One-Variable or Ground Right-Hand-Sides

If we relax the inverse context-free condition, with only one variable allowed in the *rhs* of rules, but possibly with two occurrences, both at depth 1, then the result of Theorem 1, again, is not valid anymore.

Proposition 6. *$\mathcal{R}^*(L)$ is not recursive in general when L is a HA language and \mathcal{R} an HRS whose rhs of rules are ground or of the form $d(xx)$.*

Proof. We reduce the blank accepting problem for a TM \mathcal{M} with a tape alphabet Γ and a state set S . Let us consider an alphabet containing all the symbols in Γ , S and $f, g, d, d', 0, 1, 2$ and $\#$. Like in the proof of Proposition 5, we represent a configuration of \mathcal{M} by a term $g(w)$ with $w \in \Gamma^*Q\Gamma^*$. We write $c \vdash_{\mathcal{M}} c'$ if the configuration c' is a successor of c following the transition table of \mathcal{M} . It is folklore knowledge that every such pair of configurations has the form $ua_1a_2a_3v \vdash_{\mathcal{M}} ua'_1a'_2a'_3v$ for some $u, v \in \Gamma^*$ and $(a_1, a_2, a_3, a'_1, a'_2, a'_3) \in D_{\mathcal{M}}$, where $D_{\mathcal{M}}$ is a subset of $(\Gamma \cup S)^6$ which depends only on \mathcal{M} .

A run of \mathcal{M} is a sequence of configuration $c_0 \vdash_{\mathcal{M}} \dots \vdash_{\mathcal{M}} c_n$, starting with $c_0 = g(q_i b)$ (q_i is an initial state and $b \in \Gamma$ is the blank symbol), and ending with a final configuration $c_n = g(uq_f v)$ where q_f is a final state and $u, v \in \Gamma^*$. We assume wlog that the length of every run is even. A sequence as above is represented as a right comb $f(c_0, f(c_1, \dots f(c_n, \#)))$ (here f is used as a binary symbol).

The following right-ground rewrite rules reduce a run both to 0 and 1. The rules reducing to 0 check, for each $1 \leq i \leq \frac{n}{2}$ that $c_{2i-1} \vdash_{\mathcal{M}} c_{2i}$, and the rules reducing to 1 check, for each $1 \leq i \leq \frac{n}{2}$ that $c_{2i} \vdash_{\mathcal{M}} c_{2i+1}$, and moreover they check the initial and final configurations c_0 and c_n .

$$\begin{aligned} f(g(xa_1a_2a_3y)f(g(xa'_1a'_2a'_3y)\#)) &\rightarrow 0 \\ f(g(xa_1a_2a_3y)f(g(xa'_1a'_2a'_3y)0)) &\rightarrow 0 & f(g(xq^f y)\#) &\rightarrow 1 \\ f(g(xa_1a_2a_3y)f(g(xa'_1a'_2a'_3y)1)) &\rightarrow 1 & f(g(q_{\text{init}}b)1) &\rightarrow 2 \end{aligned}$$

We consider also right-ground rewrite rules corresponding to the production rules of a regular tree grammar \mathcal{G} , with axiom (initial non-terminal) I , which generates right-combs of the above form which are expected to be a run, and a copy \mathcal{G}' of \mathcal{G} , with a disjoint set of non-terminals, and axiom I' . Let \mathcal{R} be the set of all the above right-ground rules and the one-variable rule $d(xx) \rightarrow d'(xx)$. We have that $d(II') \xrightarrow{\mathcal{R}^*} d'(02)$ iff there exists a run of \mathcal{M} starting with c_0 . \square

5 Closure of Context-Free Hedge Automata Languages

It has been observed [9] that in several cases, one class of word rewrite system preserve regularity and its symmetric class preserve context-free languages. We show in this section that a restricted case of context-free HRS, *i.e.* of the symmetric version of the systems considered in Section 4, preserve CF-HA languages. We give next some counter examples showing that the restrictions are necessary for this result.

5.1 Linear Restricted Context-Free Rewrite Rules

We call a HRS \mathcal{R} *restricted context-free* if it is context-free, and moreover, for all $f(x) \rightarrow r \in \mathcal{R}$, either x does not occur in r or x occurs in r at depth 1.

Theorem 2. *The closure $\mathcal{R}^*(L)$ of a CF-HA language L under rewriting by a linear restricted context-free HRS \mathcal{R} is a CF-HA language.*

Proof. Let $\mathcal{A}_L = (Q_L, Q_L^f, \Delta_L)$ be a CF-HA recognizing L .

First, let us construct for each rule $f(x) \rightarrow g(r_1 \dots r_n x s_1 \dots s_m) \in \mathcal{R}$ and every subterm r amongst $r_1, \dots, r_n, s_1, \dots, s_m$ a HA $\mathcal{A}_r = (Q_r, Q_r^f, \Delta_r)$ recognizing the set of ground instances of r . The construction of \mathcal{A}_r is the same as in the proof of Theorem 1. The states sets Q_r and Q_L are assumed pairwise disjoint. Like in the proof of Theorem 1, we let $\mathcal{A} := (Q, Q_L^f, \Delta)$ with $Q := Q_L \uplus \biguplus_{r \in \text{rhs}(\mathcal{R})} Q_r$ and $\Delta := \Delta_L \uplus \biguplus_{r \in \text{rhs}(\mathcal{R})} \Delta_r$.

For each $f \in \Sigma$, $q \in Q$, let $L_{f,q}$ be the context-free language in the transition (assumed unique) $f(L_{f,q}) \rightarrow q \in \Delta$, and let $\mathcal{G}_{f,q} = (Q, N_{f,q}, I_{f,q}, P_{f,q})$ be a CF grammar generating $L_{f,q}$, with alphabet (set of terminal symbols) Q , set of non terminal symbols $N_{f,q}$, axiom $I_{f,q}$, and set of production rules $P_{f,q}$. The sets of non-terminals $N_{f,q}$ are assumed pairwise disjoint.

We complete the grammars $\mathcal{G}_{f,q}$ with new non-terminals $I'_{f,q}$ and some sets $P'_{f,q}$ of new production rules containing:

- i. $I'_{f,q} := I_{f,q}$ for all $f \in \Sigma, q \in Q$,
- ii. $I'_{g,q} := q_{r_1} \dots q_{r_n} I'_{f,q} q_{s_1} \dots q_{s_m}$ for each rule $f(x) \rightarrow g(r_1 \dots r_n x s_1 \dots s_m) \in \mathcal{R}$, with $n, m \geq 0$, and
- iii. $I'_{g,q} := q_{r_1} \dots q_{r_n}$ for each rule $f(x) \rightarrow g(r_1 \dots r_n) \in \mathcal{R}$ with $x \notin \text{var}(r_1, \dots, r_n)$, or $I'_{g,q} := \varepsilon$ if $n = 0$, if $L(\mathcal{A}, q) \cap f(\mathcal{H}(\Sigma)) \neq \emptyset$.

Note that in the case ii, $x \notin \text{var}(r_1, \dots, r_n, s_1, \dots, s_m)$ because \mathcal{R} is linear.

Let $N = \bigcup_{f \in \Sigma, q \in Q} (N_{f,q} \cup \{I'_{f,q}\})$ and $P = \bigcup_{f \in \Sigma, q \in Q} (P_{f,q} \cup P'_{f,q})$.

Let us clean up these sets: if the language generated by a CF grammar $(Q, N, I'_{f,q}, P)$ is empty then we remove $I'_{f,q}$ from N and all the productions of P which contain $I'_{f,q}$. We iterate this operation, until there is no remaining non-terminals generating an empty language in N (note that the construction stops since we only remove non-terminals and productions). Let us note N' and P' the sets of non-terminals and productions obtained.

For each $f \in \Sigma, q \in Q$, let $\mathcal{G}'_{f,q} = (Q, N', I'_{f,q}, P')$, and let $L'_{f,q}$ be its language. Let $\mathcal{A}' = (Q, Q^f, \Delta')$ with $\Delta' = \{f(L'_{f,q}) \rightarrow q \mid f \in \Sigma, q \in Q, L'_{f,q} \neq \emptyset\}$. We show that $L(\mathcal{A}') = \mathcal{R}^*(L(\mathcal{A}))$.

Direction $L(\mathcal{A}') \subseteq \mathcal{R}^*(L(\mathcal{A}))$. We show more generally that for all $t \in L(\mathcal{A}', q)$, $q \in Q$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow{\mathcal{R}^*} t$. The proof is by induction on the number of occurrences of non-terminals of the form $I'_{g,q}$ in the derivation of the sequence $q_1 \dots q_n \in Q^*$ involved in the last step of $t = f(t_1 \dots t_n) \xrightarrow{\mathcal{A}'} f(q_1 \dots q_n) \xrightarrow{\mathcal{A}'} yq \in Q$. Note that this last step cannot be the application of a collapsing transition. Intuitively every such $I'_{g,q}$ corresponds to a rewrite step in $u \xrightarrow{\mathcal{R}^*} t$.

Let us note \vdash the relation of derivation using the rules of P' , and \vdash^* its transitive closure. Let $t = f(t_1 \dots t_k)$, $k \geq 0$, such that $t \xrightarrow{\mathcal{A}'} f(q_1 \dots q_k) \xrightarrow{\mathcal{A}'} q$ with $q_1 \dots q_k \in L'_{f,q}$. It means that there is a derivation $I'_{f,q} \vdash^* q_1 \dots q_k$. Let us make an induction on the number n' of non-terminals of $N' \setminus N_0$ in this derivation, where $N_0 := \bigcup_{f \in \Sigma, q \in Q} N_{f,q}$.

Base case ($n' = 1$). In this case, the only non-terminal of $N' \setminus N_0$ in $I'_{f,q} \vdash^* q_1 \dots q_k$ is the $I'_{f,q}$ at the beginning. There are only two possibilities.

Case i. $I'_{f,q} \vdash I_{f,q} \vdash^* q_1 \dots q_k$. In this case, $q_1 \dots q_k \in L_{f,q}$, hence $t = f(t_1 \dots t_k) \in L(\mathcal{A}, q)$ and we let $u = t$.

Case iii. $I'_{f,q} \vdash q_1 \dots q_k$ by the case iii of the construction of P' . It means that there is a rewrite rule $g(x) \rightarrow f(r_1 \dots r_k)$, with $x \notin \text{var}(r_1, \dots, r_k)$, and for all $i \leq k$, $q_i = q_{r_i}$. Hence, for all $i \leq k$, t_i is an instance of r_i by construction. Because of the condition in the case iii of the construction, there exists $h \in \mathcal{H}(\Sigma)$ such that $u = f(h) \in L(\mathcal{A}, q)$, and $f(h) \xrightarrow{\mathcal{R}^*} t$.

Induction step ($n' + 1$). We have $I'_{f,q} \vdash q_{r_1} \dots q_{r_n} I'_{g,q} q_{s_1} \dots q_{s_m} \vdash^* q_1 \dots q_k$, and the first production rule used in this derivation was added by the case *ii* of the construction, because there exists a rewrite rule $g(x) \rightarrow f(r_1 \dots r_n x s_1 \dots s_m) \in \mathcal{R}$. It follows that $I'_{g,q} \vdash^* q_{n+1} \dots q_{n+p}$, with $p = k - m - n$, and $g(t_{n+1} \dots t_{n+p}) \xrightarrow{\mathcal{A}^*} g(q_{n+1} \dots q_{n+p}) \xrightarrow{\mathcal{A}^*} q$.

By the induction hypothesis applied to the above derivation $I'_{g,q} \vdash^* q_{n+1} \dots q_{n+p}$, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow{\mathcal{R}^*} g(t_{n+1} \dots t_{n+p})$.

Moreover, $q_1 = q_{r_1}, \dots, q_n = q_{r_n}$ and $q_{n+p+1} = q_{s_1}, \dots, q_k = q_{s_m}$. Therefore, t_1 is an instance of r_1, \dots, t_n is an instance of r_n, t_{n+p+1} is an instance of s_1, \dots, t_k is an instance of s_m , by construction, and $g(t_{n+1} \dots t_{n+p}) \xrightarrow{\mathcal{R}} t$.

Direction $L(\mathcal{A}') \supseteq \mathcal{R}^*(L(\mathcal{A}))$. We show that for all $u \in L(\mathcal{A}, q), q \in Q$, if $u \xrightarrow{\mathcal{R}^*} t$, then $t \in L(\mathcal{A}', q)$, by induction on the length of the rewrite sequence.

Base case (0 rewrite steps). In this case, $u = t \in L(\mathcal{A}, q)$. We can note that $L(\mathcal{A}, q) \subseteq L(\mathcal{A}', q)$, because of the productions $I'_{f,q} := I_{f,q}$ added by the case *i* of the construction. Hence, $t \in L(\mathcal{A}', q)$.

Induction step ($k + 1$ rewrite steps). We have two cases.

Case ii. The last rewrite step of the sequence involves a rewrite rule $f(x) \rightarrow g(r_1 \dots r_n x s_1 \dots s_m) \in \mathcal{R}, n, m \geq 0$:

$$u \xrightarrow{\mathcal{R}^*} f(h) \xrightarrow{\mathcal{R}} g(r_1 \dots r_n h s_1 \dots s_m) \sigma = t.$$

By induction hypothesis, $f(h) \in L(\mathcal{A}', q)$. Let $f(h) \xrightarrow{\mathcal{A}^*} f(q_1 \dots q_p) \xrightarrow{\mathcal{A}^*} q$ with $q_1 \dots q_p \in L'_{f,q}$, *i.e.* $I'_{f,q} \vdash^* q_1 \dots q_p$. By the case *ii* of the construction, we have $I'_{g,q} := q_{r_1} \dots q_{r_n} I'_{f,q} q_{s_1} \dots q_{s_m} \in P'$ (note that both $I'_{g,q}$ and $I'_{f,q}$ are clean, *i.e.* members of N'), hence $q_{r_1} \dots q_{r_n} q_1 \dots q_p q_{s_1} \dots q_{s_m} \in L'_{g,q}$. It follows that $t \xrightarrow{\mathcal{A}^*} g(q_{r_1} \dots q_{r_n} q_1 \dots q_p q_{s_1} \dots q_{s_m}) \xrightarrow{\mathcal{A}^*} q$, *i.e.* $t \in L(\mathcal{A}', q)$.

Case iii. The last rewrite step of the sequence involves a rewrite rule $f(x) \rightarrow g(r_1 \dots r_n) \in \mathcal{R}$: $u \xrightarrow{\mathcal{R}^*} f(h) \xrightarrow{\mathcal{R}} g(r_1 \dots r_n) \sigma = t$. With the case *ii* of the construction, $I'_{g,q} := q_{r_1} \dots q_{r_n} \in P'$, hence $q_{r_1} \dots q_{r_n} \in L'_{g,q}$. Since for all $i \leq n$, $r_i \sigma \xrightarrow{\mathcal{A}^*} q_{r_i}, t \in L(\mathcal{A}', q)$.

(end of the proof of Theorem 2) \square

Corollary 2. *Reachability and regular hedge model-checking are decidable for linear restricted context-free HRS.*

Proof. The intersection of an CF-HA language and a HA languages is a CF-HA language, and emptiness of CF-HA is decidable. \square

It is shown in [18] that the languages in CF-HA are closures of regular tree languages modulo associativity of one or several binary function symbols. Therefore, the above results are also valid for these languages.

5.2 Linear Context-Free Rewrite Rules

Context-free HRS are named after context-free tree grammars, whose production rules have the form $N(x_1, \dots, x_n) \rightarrow r$ where N is a non-terminal of arity n (from a finite set \mathcal{N}), $x_1, \dots, x_n \in \mathcal{X}$ and $r \in \mathcal{T}(\Sigma \cup \mathcal{N}, \mathcal{X})$. Note that our definition of context-free HRS is restricted to unary non-terminals. However, even for this case of unary non-terminals and right-linear rewrite rules, the result of Theorem 2 cannot be generalized to context-free HRS.

Proposition 7. *$\mathcal{R}^*(L)$ is not a CF-HA language in general when L is a CF-HA language and \mathcal{R} a linear context-free HRS.*

Proof. Let us consider the context-free HRS: $\mathcal{R} = \{f(x) \rightarrow g(f(ax))\}$ and let $L = \{f(c)\}$. The set $\mathcal{R}^*(L)$ is $\{g(\underbrace{g(\dots g}_{n}(f(a^n c)))) \mid n \in \mathbb{N}\}$.

Using a pumping argument, we can show that it is not a CF-HA language. Assume indeed that it is recognized by a CF-HA \mathcal{A} with state set Q . In a term of $\mathcal{R}^*(L)$ with $n > |Q|$, there will two subterms $u = g^i(f(a^n c))$ and $v = g^j(u)$, with $i, j > 0$, both in $L(\mathcal{A}, q)$ for some $q \in Q$. The term $g^{n-j}(f(a^n c))$ is recognized by \mathcal{A} , and is not in $\mathcal{R}^*(L)$. \square

The above counter-example shows the importance in Theorem 2 of the condition, in the definition of restricted context-free HRS, that the variable x in *lhs* of rules occurs at a *shallow* position in *rhs*.

5.3 Restricted Context-Free Rewrite Rules

If we keep the restricted context-free condition (the variable x in the *lhs* of a rule occurs at a shallow position in the corresponding *rhs*) but we drop the linearity condition, we also lose the CF-HA preservation result of Theorem 2.

Proposition 8. *$\mathcal{R}^*(L)$ is not a CF-HA language in general when L is a CF-HA language and \mathcal{R} a restricted context-free HRS.*

Proof. Let $\mathcal{R} = \{f(x) \rightarrow f(xx)\}$ and $L = \{f(a)\}$. We have that $\mathcal{R}^* = \{f(a^n) \mid n = 2^k, k \geq 0\}$ which is not a CF-HA language. Assume indeed that this language is recognized by a CF-HA (Q, Q^f, Δ) . It means that Δ contains a transition $f(L) \rightarrow q$ where L is a context-free language of Q^* of words of Q^* of length 2^k , $k \geq 0$. The image of L under the strictly alphabetic homomorphism which translate every state $q \in Q$ into a is context-free. As it is a one letter language, it is also regular. But it is well known that this language $\{a^n \mid n = 2^k, k \geq 0\}$ is actually not regular. \square

5.4 Mixing Inverse CF and Restricted CF Rewrite Rules

We show now that the results of Theorems 1 and 2 cannot be combined. In other terms, for some HRS containing both linear inverse context-free and restricted context-free rules, the set of descendants of a HA language is not a HA language, and neither a CF-HA language or even recursive.

Proposition 9. $\mathcal{R}^*(L)$ is not recursive in general when L is a HA language and \mathcal{R} an HRS whose rules are either inverse context-free or restricted context-free and contain only one variable.

Proof. We reduce the Post Correspondence Problem (PCP). Let us consider an instance $\mathcal{P} = \{\langle u_i, v_i \rangle \mid i \leq n, u_i, v_i \in \Sigma^*\}$ of PCP on an alphabet Γ . The problem is to find a sequence $i_1, \dots, i_k \leq n$ such that $u_{i_1} \dots u_{i_k} = v_{i_1} \dots v_{i_k}$.

Let \mathcal{R} be an HRS containing a rule $f_0(x) \rightarrow f_0(\tilde{u}_i x v_i)$ for each pair $\langle u_i, v_i \rangle \in \mathcal{P}$ (\tilde{u}_i is the mirror image of u_i), a rule $f_0(x) \rightarrow f_1(x)$ and two rules $f_1(axa) \rightarrow f_2(x)$ and $f_2(axa) \rightarrow f_2(x)$ for each $a \in \Gamma$. We assume that f_0, f_1, f_2 and c are symbols not in Γ . We can show that $f_0(c) \xrightarrow{\mathcal{R}^*} f_2(c)$ iff \mathcal{P} has a solution. \square

Moreover, as we have shown that context-free HRS do not preserve HA languages (Proposition 4), the symmetric also holds for inverse-context-free HRS and CF-HA languages.

Proposition 10. $\mathcal{R}^*(L)$ is not recursive in general when L is a CF-HA language and \mathcal{R} an inverse context-free HRS.

Proof. Let \mathcal{R}_1 be the subset of the context-free rewrite rules of the HRS of the above proof of Proposition 9, and \mathcal{R}_2 be the subset of the other rules. Note that \mathcal{R}_2 is an inverse HRS.

By Theorem 2, $L = \mathcal{R}_1^*(\{f_0(c)\})$ is a CF-HA language. Like in the proof of Proposition 9, we have that $f_2(c) \in \mathcal{R}_2^*(L)$ iff the PCP has a solution. Hence, because of the decidability of the membership problem for CF-HA, $\mathcal{R}_2^*(L)$ cannot be a CF-HA language. \square

6 Conclusion

We have shown that HA and CF-HA languages are preserved by rewrite closure for an interesting class of non ground hedge rewriting rules. These rules allow us for instance to modify the structure of XML documents when processing them. We plan to extend our results to non ordered unranked trees by considering sheaves automata as in [5] or commutative hedge automata (see [3] for application to process rewrite systems).

Regularity preservation has been studied for transducing term rewriting system, *i.e.* rewrite rules corresponding to transducers rules [21]. A generalization of such classes of TRS to hedge rewriting seems conceptually close to XML transformations [13] and we plan to study the preservation of HA or CF-HA languages w.r.t. to such HRS.

References

1. P. A. Abdulla, B. Jonsson, M. Nilsson, and M. Saksena. A survey of regular model checking. In *Proc. of the 15th Int. Conf. on Concurrency Theory (CONCUR'04)*, vol. 3170 of *LNCS*, pages 35–48. Springer, 2004.

2. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Proc. of the 12th Int. Conf. on Computer Aided Verification (CAV'00)*, vol. 1855 of *LNCS*, pages 403–418, 2000.
3. A. Bouajjani and T. Touili. On computing reachability sets of process rewrite systems. In *Proc. 16th Int. Conf. Term Rewriting and Applications (RTA'05)*, vol. 3467 of *LNCS*, pages 484–499. Springer, 2005.
4. 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>. Last release October, 12th 2007.
5. S. Dal-Zilio and D. Lugiez. XML schema, tree logic and sheaves automata. In *Proc. 14th Int. Conf. Rewriting Techniques and Applications (RTA'03)*, vol. 2706 of *LNCS*, pages 246–263. Springer, 2003.
6. I. Durand and G. Sénizergues. Bottom-up rewriting is inverse recognizability preserving. In *Proc. 18th Int. Conf. Term Rewriting and Applications (RTA'07)*, vol. 4533 of *LNCS*, pages 107–121. Springer, 2007.
7. J. d'Orso and T. Touili. Regular hedge model checking. In *In Proc. of the 4th IFIP Int. Conf. on Theoretical Computer Science (TCS'06)*. IFIP, 2006.
8. R. Gilleron and S. Tison. Regular tree languages and rewrite systems. *Fundamenta Informaticae*, 24(1/2):157–176, 1995.
9. D. Hofbauer and J. Waldmann. Deleting string rewriting systems preserve regularity. *Theor. Comput. Sci.*, 327(3):301–317, 2004.
10. F. Jacquemard. Decidable approximations of term rewriting systems. In *Proc. of the 7th Int. Conf. on Rewriting Techniques and Applications (RTA'96)*, vol. 1103 of *LNCS*, pages 362–376. Springer Verlag, 1996.
11. F. Jacquemard and M. Rusinowitch. Rewrite closure of hedge-automata languages. Research Report LSV-08-05, Laboratoire Spécification et Vérification, ENS Cachan, France, 2007. Available on: <http://www.lsv.ens-cachan.fr/Publis>
12. C. Löding and A. Spelten. Transition graphs of rewriting systems over unranked trees. In *Proc. 32nd Int. Symposium on Mathematical Foundations of Computer Science (MFCS'07)*, vol. 4708 of *LNCS*, pages 67–77, 2007.
13. W. Martens and F. Neven. On the complexity of typechecking top-down XML transformations. *Theor. Comput. Sci.* Vol 336, N. 1, 2005, pages 153–180.
14. M. Murata. “Hedge Automata: a Formal Model for XML Schemata”. http://www.horobi.com/Projects/RELAX/Archive/hedge_nice.html, 2000.
15. M. Murata, D. Lee, and M. Mani. Taxonomy of xml schema languages using formal language theory. In *In Extreme Markup Languages*, 2001.
16. T. Nagaya and Y. Toyama. Decidability for left-linear growing term rewriting systems. In *Proc. 10th Int. Conf. on Rewriting Techniques and Applications (RTA'99)*, vol. 1631 of *LNCS*, pages 256–270. Springer Verlag, 1999.
17. H. Ohsaki. Beyond the regularity: Equational tree automata for associative and commutative theories. In *Proc. of CSL'01*, vol. 2142 of *LNCS*. Springer, 2001.
18. H. Ohsaki, H. Seki, and T. Takai. Recognizing boolean closed A-tree languages with membership conditional rewriting mechanism. In *Proc. of the 14th Int. Conf. on Rewriting Techniques and Applications (RTA'03)*, vol. 2706 of *LNCS*, pages 483–498. Springer Verlag, 2003.
19. J. d'Orso and T. Touili. Regular Hedge Model Checking. In *Proc. of the 4th IFIP Int. Conf. on Theoretical Computer Science (TCS'06)*. 2006, IFIP.
20. K. Salomaa. Deterministic Tree Pushdown Automata and Monadic Tree Rewriting Systems. *J. of Comp. and System Sci.*, vol. 37, pages 367–394, 1988.

21. H. Seki, T. Takai, Y. Fujinaka and Y. Kaji. Layered Transducing Term Rewriting System and Its Recognizability Preserving Property. In Proc. of 13th Int. Conf. on Rewriting Techniques and Applications (RTA'02), vol. 2378 of *LNCS*, pages 98-113, 2002.
22. T. Takai, Y. Kaji, and H. Seki. Right-linear finite path overlapping term rewriting systems effectively preserve recognizability. In *Proc. of 11th Int. Conf. on Rewriting Techniques and Applications (RTA'00)*, vol. 1833 of *LNCS*, pages 246–260, 2000.
23. T. Touili. Computing transitive closures of hedge transformations. In *In Proc. 1st Int. Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS'07)*, eWIC Series. British Computer Society, 2007.