Rewrite based Verification of XML Updates

Florent Jacquemard, Michael Rusinowitch

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Rewrite based Verification of XML Updates

Florent Jacquemard and Michael Rusinowitch

N° 7007

July 2009

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7007--FR+ENG

Rewrite based Verification of XML Updates

Florent Jacquemard and Michael Rusinowitch

Thème : Knowledge and Data Representation and Management
Équipes-Projets DAHU and CASSIS

Rapport de recherche n° 7007 — July 2009 — 36 pages

Abstract: We consider problems of access control for update of XML document. In the context of XML programming, types can be viewed as hedge automata, and static type checking amounts to verify that a program always converts valid source documents into also valid output documents. Given a set of update operations we are particularly interested by checking safety properties such as preservation of document types along any sequence of updates. We are also interested by the related policy consistency problem, that is detecting whether a sequence of authorized operations can simulate a forbidden one. We reduce these questions to type checking problems, solved by computing variants of hedge automata characterizing the set of ancestors and descendants of the initial document type for the closure of parameterized rewrite rules.

Key-words: XML transformations, Typing, Software Verification, Tree Automata, Term Rewriting.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Rewrite based Verification of XML Updates

Résumé : We consider problems of access control for update of XML document. In the context of XML programming, types can be viewed as hedge automata, and static type checking amounts to verify that a program always converts valid source documents into also valid output documents. Given a set of update operations we are particularly interested by checking safety properties such as preservation of document types along any sequence of updates. We are also interested by the related policy consistency problem, that is detecting whether a sequence of authorized operations can simulate a forbidden one. We reduce these questions to type checking problems, solved by computing variants of hedge automata characterizing the set of ancestors and descendants of the initial document type for the closure of parameterized rewrite rules.

Mots-clés : XML transformations, Typing, Software Verification, Tree Automata, Term Rewriting.

1 Introduction

XML has developed into the de facto standard for the exchange and manipulation of data on the Web [1]. XML documents are textual presentations of data stored in a tree structure, and are commonly represented as finite labeled unranked trees. In general, they are constrained by typing restrictions such as XML schemas expressing structural constraints on the organisation of the markups. Most of the typing formalisms currently used for XML are based on finite tree automata. Several formalisms exist for the specification of transformation functions for XML documents, e.g. for converting data from one source into a format suitable to a destination, for the automatic update of documents or the deletion of confidential data, e.g. for the enforcement of an access control policy (wrapping or anonymization). Among these formalisms, the W3C XQuery Update Facility [4] defines some operations for document updates.

Applying transformation functions in the context of documents following type constraints defined by schemas raises several compatibility problems. Static Type Checking in the context of XML document processing amounts to verify at compile time that every XML document which is the result of a specified query or transformation of a document with a valid input type produces an output document with a valid output type. Static Type Checking decidability is clearly dependant of the expressive power of the types and transformations that are employed. A standard approach to XML type checking is forward (resp. backward) *type inference*, that is the computation of an output (resp. input) XML type from given input (resp. output) type and a tree transformation. Then the type checking itself can be reduced to the verification of inclusion of the computed type in the given output or input type.

In this paper, motivated by XML access control problems, we consider document transformations that are arbitrary sequences of atomic update operations, and we address the problem of their type inference. Since update operations, beside relabeling document nodes, can create and delete entire XML fragments, modifying document's structure, it is not obvious to check whether they preserve the types of documents.

We propose a redefinition in term of rewrite rules (Section 3.1) of the update operations of XACU [8], a formal model for specifying access control on XML data based on the W3C XQuery Update Facility draft [4]. For these operations, and some proposed extensions, we derive type inference algorithms that can also be employed to check access control policy local consistency (i.e. to determine whether no sequence of allowed updates starting from a given document can achieve an explicitly forbidden update). Such situations may lead to serious security breaches and that are challenging to detect according to [8]. Our results are obtained through the analysis of reachability sets of term rewriting systems for unranked trees, parametrized by hedge automata, and through the computation of an extension of hedge automata called context-free hedge automata. Therefore they may give more insight on these notions that have not been investigated before.

Related work: When considering general purpose transformation languages (e.g. XDuce, CDuce) for writing transformations, typechecking is generally undecidable and approximations must be applied. In order to obtain exact algorithms, several approaches define conveniently abstract formalisms for representing transformations. Let us cite for instance TL (the transformation language) [15] whose programs can be translated in macro tree transducers [21], and k-pebble tree transducers [17], a powerful model defined so as to cover relevant fragments of XSLT [12] and other XML transformation languages. Some restrictions on schema languages and on top down tree transducers (on which transformations are based) have also been studied [16] in order to obtain PTIME type checking procedures.

The results based on tree transducers are difficult to compare to ours. On one hand, we consider a small class of atomic update operations whose expressiveness cannot be compared to general purpose transformation languages, on the other hand, the application of updates is not restricted by strategies like e.g. top-down transformations in [16]. One can note that the works on typechecking generally focus on the expressiveness of transformation languages, and are restricted to XML types modeled as regular tree languages (languages of tree automata) or DTDs (a strict subclass of regular tree languages). In our work we need to consider XML types that generalize regular tree languages and are recognized by context-free hedge automata [11].

The first access control model for XML was proposed by [6] and was extended to secure updates in [3]. In [9], the authors propose a solution to secure XUpdate queries. Static analysis has been applied to XML Access Control in [19] to determine if a query expression is guaranteed not to access to elements that are forbidden by the policy. In [8] the authors propose the XACU language. They study policy consistency and show that it is undecidable in their setting. On the positive side [2] consider policies defined in term of annotated non recursive XML DTDs and give a polynomial algorithm for checking consistency.

Several recent works have considered the application of rewriting to reason about access control policies. These works do not adress XML access control.

Organization of the paper: We introduce the needed formal background about terms, hedge automata and rewriting systems in Section 2. Then we present XML update as parameterized rewriting rules in Section 3. Finally we give application to Access Control Policies in Section 4.

2 Definitions

2.1 Unranked Ordered Trees

Terms and Hedges. We consider a finite alphabet Σ and an infinite set of variables $\mathcal{X}$. The symbols of Σ are generally denoted $a, b, c\ldots$ and the variables of $\mathcal{X}$ x, $y\ldots$ The set $\mathcal{H}(\Sigma, \mathcal{X})$ of *hedges* over Σ and $\mathcal{X}$ is the set of finite (possibly empty) sequences of terms where the set of *terms* over Σ and $\mathcal{X}$ is $\mathcal{T}(\Sigma, \mathcal{X}) := \mathcal{X} \cup \big\{a(h) \;\big|\; a \in \Sigma, h \in \mathcal{H}(\Sigma, \mathcal{X})\big\}$. The empty sequence is denoted $()$ and when h is empty, the term $a(h)$ will be simply denoted by a. We will sometimes consider a term as a hedge of length one, *i.e.* consider that $\mathcal{T}(\Sigma, \mathcal{X}) \subset \mathcal{H}(\Sigma, \mathcal{X})$. A leaf of a hedge $(t_1 \ldots t_n)$ is a leaf of one of the terms $t_1, ..., t_n$.

The sets of ground terms (terms without variables) and ground hedges are respectively denoted $\mathcal{T}(\Sigma)$ and $\mathcal{H}(\Sigma)$. The set of variables occurring in a hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$ is denoted $var(h)$. 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 root node of a term is denoted by Λ.

Substitutions. A *substitution* σ is a mapping of finite domain from $\mathcal{X}$ into $\mathcal{H}(\Sigma, \mathcal{X})$. The application of a substitution σ to terms and hedges (written with postfix notation) is defined recursively by $x\sigma := \sigma(x)$ when $x \in dom(\sigma)$, $y\sigma := \sigma(x)$ when $y \in \mathcal{X} \setminus dom(\sigma)$, $(t_1 \dots t_n)\sigma := (t_1\sigma \dots t_n\sigma)$ for $n \geq 0$, $f(h)\sigma := f(h\sigma)$.

Contexts. A *context* is a hedge $u \in \mathcal{H}(\Sigma, \mathcal{X})$ with a distinguished variable x_u linear (with exactly one occurrence) in u. The application of a context u to a hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$ is defined by $u[h] := u\{x_u \mapsto h\}$: it consists in inserting h into an hedge in u at the position of x_u. Sometimes, we write $t[s]$ in order to emphasis that s is a subterm (or subhedge) of t.

2.2 Hedge Automata

We consider two typing formalisms for XML documents, defined as two classes of unranked tree automata. The first class is the hedge-automata [18], denoted HA. Most popular XML typing schemas like W3C XML Schemas or Relax NG are equivalent in expressiveness to HA. The second and perhaps lesser known class is the context-free hedge automata, denoted CF-HA and introduced in [20]. CF-HA are strictly more expressive than HA and we shall see that they are of interest for the typing of certain update operations.

Definition 1 *A* hedge automaton *(resp.* context-free hedge automaton*) is a tuple $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ where Σ is an finite unranked alphabet, Q is a finite set of states disjoint from Σ, $Q^{\mathsf{f}} \subseteq Q$ is a set of final states, and Δ is a set of transitions of the form $a(L) \to q$ where $a \in \Sigma$, $q \in Q$ 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 $h, h' \in \mathcal{H}(\Sigma \cup Q)$ as follows: $h \xrightarrow[\mathcal{A}]{} h'$ iff there exists a context $u \in \mathcal{H}(\Sigma, \{x_C\})$ and a transition $a(L) \to q \in \Delta$ such that $h = u[a(q_1 \dots q_n)]$, with $q_1 \dots q_n \in L$ and $h' = u[q]$. The relation $\xrightarrow[\mathcal{A}]{*}$ is the transitive closure of $\xrightarrow[\mathcal{A}]{}$.

Collapsing Transitions. We consider the extension of HA and CF-HA with so called with *collapsing transitions* which are special transitions of the form $L \to q$ where $L \subseteq Q^*$ is a CF language and q is a state. The move relation for the extended set of transitions generalizes the above definition with the case $u[q_1 \dots q_n] \xrightarrow[\mathcal{A}]{} u[q]$ if $L \to q$ is a collapsing transition of $\mathcal{A}$ and $q_1 \dots q_n \in L$. Note that we do not exclude the case $n = 0$ in this definition, i.e. L may contain the empty word in $L \to q$. Collapsing transitions with a singleton language L containing a length one word (i.e. transitions of the form $q \to q'$, where q and q' are states) correspond to *ε-transitions* for tree automata.

Languages. The language of a HA or CF-HA $\mathcal{A}$ in one of its states q, denoted by $L(\mathcal{A}, q)$, is the set of ground hedges $h \in \mathcal{H}(\Sigma)$ such that $h \xrightarrow[\mathcal{A}]{*} q$. We say sometimes that an hedge of $L(\mathcal{A}, q)$ has type q (when $\mathcal{A}$ is clear from context). A hedge is accepted by $\mathcal{A}$ if there exists $q \in Q^{\mathsf{f}}$ such that $h \in L(\mathcal{A}, q)$. The language of $\mathcal{A}$, denoted by $L(\mathcal{A})$ is the set of hedge accepted by $\mathcal{A}$.

Note that without collapsing transitions, all the hedges of $L(\mathcal{A}, q)$ are terms. Indeed, by applying standard transitions of the form $a(L) \to a$, one can only reduce length-one hedges into states. But collapsing transitions permit to reduce a ground hedge of length more than one into a single state.

The ε-transitions of the form $q \to q'$ do not increase the expressiveness HA or CF-HA (see [5] for HA and the proof for CF-HA is similar). But the situation is not the same in general for collapsing transitions: collapsing transitions strictly extend HA in expressiveness, and even collapsing transitions of the form $L \to q$ where the left member L is a finite (hence regular) word language.

Example 1 [11]. The extended HA $\mathcal{A} = \big(\{q, q_a, q_b, q_{\mathsf{f}}\}, \{g, a, b\}, \{q_{\mathsf{f}}\}, \{a \to q_a, b \to q_b, g(q) \to q_{\mathsf{f}}, q_a q_b \to q\}\big)$ recognizes $\{g(a^n b^n) \,|\, n \geq 1\}$ which is not a HA language.

However, collapsing transitions can be eliminated from CF-HA, when restricting to the recognition of terms.

Lemma 1 ([11]) *For every extended CF-HA over Σ with collapsing transitions $\mathcal{A}$, there exists a CF-HA $\mathcal{A}'$ without collapsing transitions such that $L(\mathcal{A}') = L(\mathcal{A}) \cap \mathcal{T}(\Sigma)$.*

Properties. It is known that for both classes of HA and CF-HA membership and emptiness problems are decidable in PTIME [18, 20]. Moreover HA languages are closed under Boolean operations, but CF-HA are not closed under intersection and complementation. The intersection of a CF-HA language and a HA language is a CF-HA language. All these results are effective, with PTIME constructions of automata of polynomial sizes for the closures under union and intersection.

We call a HA or CF-HA $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ *normalized* if for every $a \in \Sigma$ and every $q \in Q$, there is at most one transition rule $a(L_{a,q}) \to q$ in Δ. Every HA (resp. CF-HA) can be transformed into a normalized HA (resp. CF-HA) in polynomial time by replacing every two rules $a(L_1) \to q$ and $a(L_2) \to q$ by $a(L_1 \cup L_2) \to q$.

2.3 Infinite Term Rewrite Systems

We use term rewriting as a formalism for modeling XML update operations. For this purpose, we propose a non-standard definition of term rewriting, extending the classical one in two ways: the application of rewrite rules is extended from ranked terms to unranked terms and second, the rules are parametrized by HA languages (i.e. each parametrized rule can represent an infinite number of unparametrized rules).

Term Rewriting Systems. A term rewriting system $\mathcal{R}$ over a finite unranked alphabet Σ (TRS) is a set of *rewrite rules* of the form $\ell \to r$ where $\ell \in \mathcal{H}(\Sigma, \mathcal{X}) \setminus \mathcal{X}$ and $r \in \mathcal{H}(\Sigma, \mathcal{X})$; ℓ and r are respectively called left- and right-hand-side (*lhs* and *rhs*) of the rule. Note that we do note assume the cardinality of $\mathcal{R}$ to be finite.

The rewrite relation $\xrightarrow[\mathcal{R}]{}$ of an TRS $\mathcal{R}$ 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 context u, a rule $\ell \to r \in \mathcal{R}$ and a substitution σ such that $h = u[\ell\sigma]$ and $h' = u[r\sigma]$. The reflexive and transitive closure of $\xrightarrow[\mathcal{R}]{}$ is denoted $\xrightarrow[\mathcal{R}]{*}$.

Example 2 With $\mathcal{R} = \{g(x) \to x\}$, we have $g(h) \xrightarrow[\mathcal{R}]{} h$ for all $h \in \mathcal{H}(\Sigma, \mathcal{X})$ (the term is reduced to the hedge h of its arguments). With $\mathcal{R} = \{g(x) \to g(axb)\}$, $g(c) \xrightarrow[\mathcal{R}]{*} g(a^n cb^n)$ for every $n \geq 0$.

Parametrized Term Rewriting Systems. Let $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ be a HA. A term rewriting system over Σ and parametrized by $\mathcal{A}$ (PTRS) (see [10]is given by a finite set, denoted $\mathcal{R}/\mathcal{A}$, of rewrite rules $\ell \to r$ where $\ell \in \mathcal{H}(\Sigma, \mathcal{X})$ and $r \in \mathcal{H}(\Sigma \uplus Q, \mathcal{X})$ and symbols of Q can only label leaves of r. In this notation, $\mathcal{A}$ may be omitted when it is clear from context or not necessary. The rewrite relation $\xrightarrow[\mathcal{R}/\mathcal{A}]{}$ associated to a PTRS $\mathcal{R}/\mathcal{A}$ is defined as the rewrite relation $\xrightarrow[\mathcal{R}[\mathcal{A}]]{}$ where the TRS $\mathcal{R}[\mathcal{A}]$ is the (possibly infinite) set of all rewrite rules obtained from rules $\ell \to r$ in $\mathcal{R}/\mathcal{A}$ by replacing in r every state $q \in Q$ by a ground hedge of $L(\mathcal{A}, q)$. Several example of rewrite rules can be found in Figure 1 below.

Properties. Given a set $L \subseteq \mathcal{H}(\Sigma, \mathcal{X})$ and a PTRS $\mathcal{R}/\mathcal{A}$, we denote by $post^*_{\mathcal{R}/\mathcal{A}}(L) = \{h \in \mathcal{H}(\Sigma, \mathcal{X}) \mid \exists h' \in L, h' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h\}$ and $pre^*_{\mathcal{R}/\mathcal{A}}(L) = \{h \in \mathcal{H}(\Sigma, \mathcal{X}) \mid \exists h' \in L, h \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h'\}$.
Ground reachability is the problem to decide, given two hedges $h, h' \in \mathcal{H}(\Sigma)$ and a PTRS $\mathcal{R}/\mathcal{A}$ whether $h \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h'$. Reachability problems for ground ranked tree rewriting have been investigated in e.g. [10]. C. Löding [13] has obtained results in a more general setting where rules of type $L \to R$ specify the replacement of any element of a regular language L by any element of a regular language R. Then [14] has extended some of these works to unranked tree rewriting for the case of *subtree and flat prefix rewriting* which is a combination of standard ground tree rewriting and prefix word rewriting on the ordered leaves of subtrees of height 1.
Typechecking is the problem to decide, given two sets of terms τ_{in} and τ_{out} called input and output types (generally presented as HA) and a PTRS $\mathcal{R}/\mathcal{A}$ whether $post^*(\tau_{in}) \subseteq \tau_{out}$ or equivalently $\tau_{in} \subseteq pre^*(\tau_{out})$ [17].

Note that reachability is a special case of model checking, when both τ_{in} and τ_{out} are singleton sets. Hence typechecking is undecidable as soon as reachability is.
One related problem, called *type inference*, is, given a of PTRS $\mathcal{R}/A$ and a HA or CF-HA language L, to construct a HA or CF-HA recognizing $post^*_{\mathcal{R}}(L)$ or $pre^*_{\mathcal{R}}(L)$.

3 Type Inference for Update Operations

In this section, we address the problem of type inference for arbitrary finite sequence of update operations. More precisely, we propose a redefinition in term of PTRS rules (Section 3.1) of the update operations of XACU [8] and some extensions. Then, we show how to construct HA and CF-HA recognizing respectively $post^*_{\mathcal{R}}(L)$ and $pre^*_{\mathcal{R}}(L)$ given a HA or CF-HA language L and a

XACU				XACU+			
$a(x)$	$\rightarrow$	$b(x)$	REN				
$a(x)$	$\rightarrow$	$a(p\,x)$	$\mathsf{INS}_{\mathsf{first}}$	$a(x)$	$\rightarrow$	$b(p\,x)$	$\mathsf{INS}'_{\mathsf{first}}$
$a(x)$	$\rightarrow$	$a(x\,p)$	$\mathsf{INS}_{\mathsf{last}}$	$a(x)$	$\rightarrow$	$b(x\,p)$	$\mathsf{INS}'_{\mathsf{last}}$
$a(xy)$	$\rightarrow$	$a(x\,p\,y)$	$\mathsf{INS}_{\mathsf{into}}$				
$a(x)$	$\rightarrow$	$p\,a(x)$	$\mathsf{INS}_{\mathsf{left}}$				
$a(x)$	$\rightarrow$	$a(x)\,p$	$\mathsf{INS}_{\mathsf{right}}$				
$a(x)$	$\rightarrow$	p	RPL	$a(x)$	$\rightarrow$	$p_1 \ldots p_n$	RPL'
$a(x)$	$\rightarrow$	$()$	DEL	$a(x)$	$\rightarrow$	x	$\mathsf{DEL}_{\mathsf{s}}$

Figure 1: PTRS rules for XACU and extension

PTRS $\mathcal{R}$ representing XACU operations (Sections 3.2) or extended updates (Section 3.3).

The motivation for showing these results are twofold. First, these constructions permit to address the problems of reachability and typechecking. Second, they also permit the synthesis of missing input or output types. Imagine that a PTRS $\mathcal{R}$ is given, as well as an input type τ_{in}, defined as an HA, but that the output type (for the application of rules of $\mathcal{R}$ to terms of τ_{in}) is not known. The result of Theorem 1 ensures that we can build a CF-HA recognizing $post^*_{\mathcal{R}}(\tau_{in})$ and which can be use as a definition of a synthesized output type for $\mathcal{R}$. Similarly, the result of Theorem 3 can be used to synthesis an input type, defined by the HA constructed for $pre^*_{\mathcal{R}}(\tau_{out})$, given an output type τ_{out} and a PTRS $\mathcal{R}/\mathcal{A}$.

3.1 Update Operations

Figure 1 displays PTRS rules corresponding to the rules of XACU as defined in [8] (in the first column) and to some extensions (in the second column). We call XACU the class of all PTRS containing rules of the kind presented in the first column of Figure 1, and XACU+ the class of all PTRS containing any rule presented in Figure 1.

In this section we assume given an unranked alphabet Σ and a HA $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$. The rewrite rules are parametrized by states p, p_1,..,p_n of $\mathcal{A}$.

XACU rules. Let us first describe the update operations of XACU (see also [8]). REN renames a node: it changes it label from a into b. Such a rule leaves the structure of the term unchanged. $\mathsf{INS}_{\mathsf{first}}$ inserts a term of type p at the first position below a node labeled by a. $\mathsf{INS}_{\mathsf{last}}$ inserts at the last position and $\mathsf{INS}_{\mathsf{into}}$ at an arbitrary position below a node labeled by a. $\mathsf{INS}_{\mathsf{left}}$ (resp. $\mathsf{INS}_{\mathsf{right}}$) insert a term of type p at the left (resp. right) sibling position to a node labeled by a. DEL deletes a whole subterm whose root node is labeled by a and RPL replaces such a subterm by a term of type p.

Example 3 The patient data in a hospital are stored in an XML document whose DTD type can be recognized by an HA $\mathcal{A}$ with rules:

$$
\begin{array}{rclrclrcl}
\mathsf{hospital}(p_{\mathsf{p}}^*) & \rightarrow & p_{\mathsf{h}} & \mathsf{name}(p_c^*) & \rightarrow & p_{\mathsf{n}} & \mathsf{a} & \rightarrow & p_c \\
\mathsf{patient}(p_{\mathsf{n}}\, p_{\mathsf{t}}) & \rightarrow & p_{\mathsf{pa}} & \mathsf{drug}(p_c^*) & \rightarrow & p_{\mathsf{dr}} & \mathsf{b} & \rightarrow & p_c \\
\mathsf{patient}(p_{\mathsf{n}}) & \rightarrow & p_{\mathsf{epa}} & \mathsf{diagnosis}(p_c^*) & \rightarrow & p_{\mathsf{dia}} & \mathsf{c} & \rightarrow & p_c \\
\mathsf{treatment}(p_{\mathsf{dr}}\, p_{\mathsf{dia}}\, p_{\mathsf{da}}) & \rightarrow & p_{\mathsf{t}} & \mathsf{date}(p_c^*) & \rightarrow & p_{\mathsf{da}} & \vdots & &
\end{array}
$$

For instance we can use a DEL rule $\mathsf{patient}(x) \rightarrow ()$ for deleting a patient, and a $\mathsf{INS}_{\mathsf{last}}$ rule $\mathsf{hospital}(x) \rightarrow \mathsf{hospital}(x\, p_{\mathsf{pa}})$ to insert a new patient, at the last position below the root node hospital. We can ensure that the patient newly added has an empty treatments list (to be completed later) using the rule $\mathsf{hospital}(x) \rightarrow \mathsf{hospital}(x\, p_{\mathsf{epa}})$. The $\mathsf{INS}_{\mathsf{right}}$ rule $\mathsf{name}(x) \rightarrow \mathsf{name}(x)\, p_{\mathsf{t}}$ can be used to insert later a treatment next to the patient's name.

Extended rules. In XACU+ we introduce several extensions of the rules of XACU. We shall see in Section 3.3 that the typing of these extended operations is different from the typing of the operation of XACU: while the type of terms obtained by XACU operations can be described by HA, CF-HA must be used in order to describe the type of terms obtained by XACU+. A restriction of the insertion rules of XACU (the rules called INS_*), following the definitions in [8], is that the label of the node at the top of the lhs of the rules is left unchanged. Only the rule REN permits to change the label of a node in a term, while preserving the other nodes. The rules INS'_* combine the application of the corresponding insert operation INS_* and of a node renaming REN. We will see in Section 3.3 that allowing such combinations has important consequences wrt type inference.
The rule $\mathsf{DEL}_{\mathsf{s}}$ deletes a single node n whose arguments inherit the position. It can be employed to build a user view as in [7].

Example 4 Assume that some patients of the hospital of Example 3 are grouped into one category like in $\mathsf{hospital}(\ldots \mathsf{priority}(p_{\mathsf{pa}}^*) \ldots)$, and that we want to delete the category priority while keeping the patients information. This can be done with the $\mathsf{DEL}_{\mathsf{s}}$ rule $\mathsf{priority}(x) \rightarrow x$.

Finally, with RPL' we slightly generalize the rule RPL by allowing a subterm whose root node is labeled by a to be replaced by a sequence of n terms of respective types $p_1, \ldots, p_n$.
Note that RPL and DEL are special cases of RPL, with $n = 1$ and $n = 0$ respectively.

3.2 Forward Type Inference for XACU Rules

In this section and the following, we want to characterize the sets of terms which can be obtained, from terms of a given type, by arbitrary application of updates operations as PTRS rules. For this purpose, we shall study the recognizability (by HA and CF-HA), of the forward closure (*post**) of automata languages under the above rewrite rules.

Theorem 1 *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU, for all HA language L, $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of an HA of size polynomial and which can be constructed in PTIME on the size of $\mathcal{R}$, $\mathcal{A}$ and an HA of language L.*

Proof. (sketch, see Appendix B for a complete proof). We consider a normalised HA $\mathcal{A}_L$ recognizing L and add transitions (but no states) to the NFAs defining its horizontal languages in transitions $a(L_{a,q}) \to q$. For instance, if $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$ we add one transition $(i_{a,q}, p, i_{a,q})$ looping on the initial state $i_{a,q}$ of the NFA for $L_{a,q}$. If $a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}$, and there exists a transition $(s, q, s')'$ in some NFA, we add one transition (s', p, s'). □

Let us come back to our motivations. A first consequence of Theorem 1 concerns to the typechecking problem.

Corollary 1 *The typechecking is decidable in PTIME for XACU.*

Proof. Let τ_{in} and τ_{out} be two HA languages (resp. input and output types), and let $\mathcal{R}/\mathcal{A}$ by a PTRS. We want to know whether $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in}) \subseteq \tau_{out}$. Following Theorem 1, $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in})$ is a HA language. Hence $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in}) \cap \overline{\tau_{out}}$ is a HA language, and testing its emptiness solves the problem. □

Regarding the problem of type synthesis, if we are given $\mathcal{R}/\mathcal{A}$ and an input type τ_{in}, Theorem 1 provides an output type presented as a HA.

3.3 Forward and Backward Type Inference for XACU+ Rules

Theorem 1 is no longer true for the rules of the extension XACU+: the examples below show that the rules of XACU+ \ XACU do not preserve HA languages in general. However, we prove in Theorem 2 that the rules of XACU+ preserve the larger class of CF-HA language.

Example 5 Let $\Sigma = \{a, b, c, c'\}$ and let $\mathcal{R}$ be the finite TRS containing the two $\mathsf{INS}'_{\mathsf{first}}$ and $\mathsf{INS}'_{\mathsf{last}}$ rules $c(x) \to c'(ax), c'(x) \to c(xb)$. We have $post^*_{\mathcal{R}}\big(\{c\}\big) \cap \mathcal{H}(\Sigma) = \{c(a^n b^n) \mid n \geq 0\}$, and this set is not a HA language. It follows that $post^*_{\mathcal{R}}\big(\{c\}\big)$ is not a HA language. ◇

Example 6 Let $\Sigma = \{a, b, c\}$, let $\mathcal{R}$ be the finite TRS with one $\mathsf{DEL_s}$ rule $c(x) \to x$ and let L be the HA language containing exactly the terms $c(ac(a \ldots c \ldots b)b)$; it is recognized by the HA with the set of transition rules $\big\{a \to q_a, b \to q_b, c\big(\{(), q_a\,q\,q_b\}\big) \to q\big\}$. We have $post^*_{\mathcal{R}}(L) \cap c\big(\{a,b\}^*\big) = \{c(a^n b^n) \mid n \geq 0\}$, hence $post^*_{\mathcal{R}}(L)$ is not a HA language. ◇

Theorem 2 *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU+, for all CF-HA term language L, $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of an CF-HA of size polynomial and which can be constructed in PTIME on the size of $\mathcal{R}$, $\mathcal{A}$ and an CF-HA recognizing L.*

Proof. (sketch, see Appendix C for a complete proof). We consider a normalised HA $\mathcal{A}_L$ recognizing L and, very roughly, we define new CFG $\mathcal{G}_{a,q}$ for the horizontal languages as the union of CFG of transitions of A_L with a new initial non-terminal $I'_{a,q}$ and new production rules according to $\mathcal{R}/\mathcal{A}$. For instance, if $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$, we add a production rule $I'_{b,q} := I'_{a,q}$ and for

$a(x) \to b(p\,x)$, we add $I'_{b,q} := pI'_{a,q}$. Moreover, we also add collapsing transitions like $p_1 \dots p_n \to q$ if $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$. □

Corollary 2 *The typechecking is decidable in PTIME for* $\mathsf{XACU}+$.

Proof. The proof is the same as for Corollary 1, because the intersection of a CF-HA and a HA language is a CF-HA language (and there is an effective PTIME construction of an CF-HA of polynomial size) and emptiness of CF-HA is decidable in PTIME. □

Theorem 3 *Given a HA* $\mathcal{A}$ *on* Σ *and a PTRS* $\mathcal{R}/\mathcal{A} \in \mathsf{XACU}+$, *for all HA language* L, $pre^*_{\mathcal{R}/\mathcal{A}}(L)$ *is a HA the language.*

Regarding the problem of type synthesis for a $\mathcal{R}/\mathcal{A} \in \mathsf{XACU}+$, if only an output type τ_{out} is given, then Theorem 2 provides an input type for $\mathcal{R}/\mathcal{A}$ presented as a HA, and if only an input type τ_{in} is given, then Theorem 2 provides an output type presented as a CF-HA. Unlike HA, CF-HA are not popular type schemas, but HA solely do not permit to extend the results of Theorem 1 as shown by the above examples.

4 Access Control Policies for Updates

In this last section we study some models of Access Control Policies (ACP) for the update operations defined in Section 3, and verification problems for these ACP.

4.1 Term Rewrite Systems with Global Membership Constraints

The ACP language $\mathsf{XACU}_{\mathsf{annot}}$ introduced in [8] follows the approach of DTD with security annotations of [7] to specify the read and write access authorizations for XML documents in the presence of a DTD. Annotated DTDs offer an elegant formalism for ACP specification, which is especially convenient for developing techniques of type analysis. However, it imposes the strong restriction that every document t to which we want to apply an update operation (under the given ACP) must comply to the DTD D used for the ACP specification.

In our rewrite based formalism, this condition may be expressed by adding global constraints to the parametrized rewrite rules of Section 2.3. These global constraints restrict the whole term to be rewritten (not only the redex) to belong to a given regular language. Theorem 4 below shows that, unfortunately, adding such constraints to ground rules (which are a very special kind of RPL rules) makes the reachability undecidable.

Given a HA $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$, a term rewriting system over Σ, parametrized by $\mathcal{A}$ and with global constraints (PGTRS) is given by a finite set, denoted $\mathcal{R}/\mathcal{A}$, of constrained rewrite rules $L :: \ell \to r$ where ℓ and r satisfy the conditions of the rewrite rules of Section 2.3 and $L \subseteq \mathcal{T}(\Sigma)$ is a HA language. A PGTRS is called *uniform* if the language L is the same for every rule. The rewrite relation for PGTRS is defined as the restriction of the relation defined in Section 2.3 to ground terms: for the application of a rule $L :: \ell \to r$ to a term t, we require that $t \in L$.

XACU$_2$				XACU$_2$+			
$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,p\,a(x)\,z)$	$\mathsf{INS}_{2,\mathsf{left}}$				
$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,a(x)\,p\,z)$	$\mathsf{INS}_{2,\mathsf{right}}$				
$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,p\,z)$	RPL_2	$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,p_1\ldots p_n\,z)$	RPL'_2
$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,z)$	DEL_2	$b(y\,a(x)\,z)$	$\rightarrow$	$b(y\,x\,z)$	$\mathsf{DEL}_{2,\mathsf{s}}$

Figure 2: PTRS rules for XACU with context control

Theorem 4 *Reachability is undecidable for uniform PGTRS without variables and parameters.*

The result can be contrasted with some decidability results on ground rewriting [10]. It is also a refinement of [8] where XPath queries are used filter out nodes where the updates apply. As a corollary, reachability, hence inconsistency (see Section 4.3), are undecidable for XACU_{annot} ACP based on annotated recursive DTDs.

4.2 XACU$_2$+: Rewrite Rules with Context Control

The PTRS rewrite rules of Section 3 permit to define a minimal control for the application of the updates operations. Indeed, all the lhs of rules have the form $a(x)$ (or $a(xy)$ for $\mathsf{INS}_{\mathsf{into}}$), meaning that the application to such rules is restricted to nodes labeled with a (i.e. to nodes of DTD element type a if the document conforms to a given fixed DTD).

For the rules with an hedge at rhs (like $\mathsf{INS}_{\mathsf{left}}$, $\mathsf{INS}_{\mathsf{right}}$, RPL, DEL, $\mathsf{DEL}_{\mathsf{s}}$...) we can extend this idea by furthermore constraining the label of the node at the parent node of the performed update. The generalized rules are defined in Figure 2.

Example 7 The DEL_2 rule $\mathsf{hospital}(y\,\mathsf{patient}(x)\,z) \rightarrow \mathsf{hospital}(y\,z)$ can be used to delete a patient only if it is located under a hospital node.

This approach can be compared to the annotated DTD of [7]. The security annotations of [7] are indeed mappings *ann* from pairs of DTD elements types (b, a) into values of Y, N or $[q]$ (for resp. read access allowed, denied or conditionally allowed, where q is an XPath qualifier). An annotation $ann(b, a) = Y$ or N or $[q]$ indicates that the a children of b elements (in an instantiation of the given DTD D) are accessible, inaccessible or conditionally accessible respectively. This approach is limited to the case of unambiguous DTDs, where the element type a can have at most one element b as parent.

Let us call XACU$_2$+ the class of all PTRS containing rules of XACU+ or rules of the kind described in Figure 2. The construction of Theorem 3 for backward type inference can be straightforwardly extended from XACU+ to XACU$_2$+.

Theorem 5 *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU$_2$+, for all HA language L, $pre^*_{\mathcal{R}/\mathcal{A}}(L)$ is a HA language.*

4.3 Local Inconsistency of ACP

Following e.g. [2], an ACP for XML updates can be defined by a pair $(\mathcal{R}_a/\mathcal{A}, \mathcal{R}_f/\mathcal{A})$ of PTRS, where $\mathcal{R}_a$ contains allowed operations and $\mathcal{R}_f$ contains forbidden operations. Such an ACP is called *inconsistent* [8, 2] if some forbidden operation can be simulated through a sequence of allowed operations.

Example 8 Assume that in the $\mathsf{hospital}$ document of example 3, it is forbidden to rename a $\mathsf{patient}$, that is the following update of RPL_2 is forbidden: $\mathsf{patient}(y\, \mathsf{name}(x)\, z) \to \mathsf{patient}(y\, p_{\mathsf{n}}\, z)$.
If the following updates are allowed: $\mathsf{patient}(x) \to ()$ for deleting a $\mathsf{patient}$, and $\mathsf{hospital}(x) \to \mathsf{hospital}(x\, p_{\mathsf{pa}})$ to insert a $\mathsf{patient}$, then we have an inconsistency in the sense of [2] since the effect of the forbidden update can be obtained by a combination of allowed updates.

Using the results of Section 3, we can decide the above problems individually for terms of D. More precisely, we solve the following problem called *local inconsistency*: given a HA $\mathcal{A}$ over Σ, an ACP $(\mathcal{R}_a/\mathcal{A}, \mathcal{R}_f/\mathcal{A})$ and a term $t \in \mathcal{T}(\Sigma)$, does there exists $u \in \mathcal{T}(\Sigma)$ such that $t \xrightarrow[\mathcal{R}_f/\mathcal{A}]{} u$ and $t \xrightarrow[\mathcal{R}_a/\mathcal{A}]{*} u$?

Theorem 6 *Local inconsistency is decidable in PTIME for* $\mathsf{XACU}+$.

Proof. It can be easily shown that the set $\{u \in \mathcal{T}(\Sigma) \mid t \xrightarrow[\mathcal{R}_f/\mathcal{A}]{} u\}$ is the language of a HA of size polynomial and constructed in PTIME on the sizes of $\mathcal{A}$, $\mathcal{R}_f$ and t. By Theorem 2, $post^*_{\mathcal{R}_a/\mathcal{A}}(\{t\})$ is the language of a CF-HA of polynomial size and constructed in polynomial time on the sizes of $\mathcal{A}$, $\mathcal{R}_a$ and t. The ACP is locally inconsistent wrt t iff the intersection of the two above language is non empty, and this property can be tested in polynomial time. □

Conclusion

We have proposed a model for XML updates based on term rewriting, and shown that type inference is possible and the problems of reachability and typecheking are decidable for the arbitrary application of XACU update rules, as well as some extensions, when the application is only controlled by the label of the node at the update position and also at its parent node. We have also shown that these problems become undecidable when restricting the application of update operations to documents conforming to a fixed given DTD.

As further works, we could study restrictions on the regular tree languages in the constraints of PGTRS enabling the decidability of typechecking for XACU rules with global constraints. Another interesting topic, w.r.t. the verification ACP for updates based on annotated DTDs is the access conditioned with XPath queries. We could model this with rewrite rules constrained by XPath qualifiers. Reachability is undecidable for such a formalism, even when the rules are ground (a consequence of a result of [8][1]). However, the construction of [8] involves upward navigation; some fragments of downward Core XPath could permit to obtain decidability.

[1] Actually in [8], the undecidability of the inconsistency problem is stated but the construction in this paper proves the undecidability of reachability as well.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML.* Morgan Kaufmann, 1999.

[2] L. Bravo, J. Cheney, and I. Fundulaki. ACCOn: checking consistency of XML write-access control policies. In *Proceedings of 11th Int. Conf. on Extending Database Technology (EDBT)*, volume 261 of *ACM Int. Conf. Proceeding Series*, pages 715–719. ACM, 2008.

[3] S. C. Lim and S. H. Son. Access control of XML documents considering update operations. In *Proc. of ACM Workshop on XML Security*, 2003.

[4] D. Chamberlin, M. Dyck, D. Florescu, J. Melton, J. Robie, and J. Siméon. Xquery update facility. W3C, 2009.

[5] 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`, 1997. release October, 12th 2007.

[6] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Securing XML Documents. In *Proceedings of the 7th Int. Conf. on Extending Database Technology (EDBT)*, volume 1777 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 2000.

[7] W. Fan, C.-Y. Chan, and M. Garofalakis. Secure XML querying with security views. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data (SIGMOD'04)*, pages 587–598, ACM, 2004.

[8] I. Fundulaki and S. Maneth. Formalizing xml access control for update operations. In *Proceedings of the 12th ACM symposium on Access control models and technologies (SACMAT)*, pages 169–174, ACM, 2007.

[9] A. Gabillon. A formal access control model for XML databases. In *Proceedings Second VLDB Workshop on Secure Data Management (SDM)*, volume 3674 of *Lecture Notes in Computer Science*, pages 86–103. Springer, 2005.

[10] R. Gilleron. Decision problems for term rewrite systems and recognizable tree languages. In *$8^t h$ Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 480 of *Lecture Notes in Computer Science*, pages 148–159, Springer, 1991.

[11] F. Jacquemard and M. Rusinowitch. Closure of Hedge-automata languages by Hedge rewriting. In *Proceedings of the 19th Int. Conf. on Rewriting Techniques and Applications (RTA)*, volume 5117 of *Lecture Notes in Computer Science*, pages 157–171, Springer, 2008.

[12] M. Kay. Xsl transformations (xslt) 2.0. W3c working draft, World Wide Web Consortium, 2003. Available at `http://www.w3.org/TR/xslt20`.

[13] C. Löding. Ground tree rewriting graphs of bounded tree width. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2285 of *Lecture Notes in Computer Science*, pages 559–570, Springer, 2002.

[14] C. Löding and A. Spelten. Transition graphs of rewriting systems over unranked trees. In *Proceedings 32nd International Symposium on Mathematical Foundations of Computer Science 2007 (MFCS)*, volume 4708 of *LNCS*, pages 67–77, Springer, 2007.

[15] S. Maneth, A. Berlea, T. Perst, and H. Seidl. XML type checking with macro tree transducers. In *24th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS)*, pages 283–294, 2005.

[16] W. Martens and F. Neven. Frontiers of tractability for typechecking simple xml transformations. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 23–34. ACM, 2004.

[17] T. Milo, D. Suciu, and V. Vianu. Typechecking for XML transformers. *J. Comput. Syst. Sci.*, 66(1):66–97, 2003.

[18] M. Murata. "Hedge Automata: a Formal Model for XML Schemata". Web page, 2000.

[19] M. Murata, A. Tozawa, M. Kudo, and S. Hada. Xml access control using static analysis. *ACM Trans. Inf. Syst. Secur.*, 9(3):292–324, 2006.

[20] 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)*, volume 2706 of *LNCS*, pages 483–498, Springer, 2003.

[21] T. Perst and H. Seidl. Macro forest transducers. *Information Processing Letters*, 89:141–149, 2004.

A Appendix: proof of Lemma 1

In this proof and the following, we describe the *CF grammars* used for defining the horizontal languages of CF-HA transitions as tuples $\mathcal{G} = (\Sigma, \mathcal{N}, I, \Gamma)$, where Σ is a finite alphabet (set of terminal symbols), N is a set of non terminal symbols, $I \in \mathcal{N}$ is the initial non-terminal, and $\Gamma \in \mathcal{N} \times (\mathcal{N} \cup \Sigma)^*$ is a set of production rules.

Lemma 1 [11]. *For every extended CF-HA over Σ with collapsing transitions $\mathcal{A}$, there exists a CF-HA $\mathcal{A}'$ without collapsing transitions such that $L(\mathcal{A}') \cap \mathcal{T}(\Sigma) = L(\mathcal{A}) \cap \mathcal{T}(\Sigma)$.*

Proof. Let $\mathcal{G} = (Q, N, I, \Gamma)$ and $\mathcal{G}_1 = (Q, N_1, I_1, \Gamma_1)$ be two CF grammars over the same finite alphabet Q. Below, $\mathcal{G}$ and $\mathcal{G}_1$ are respectively meant to generate the languages L and L_1 of CF HA transitions $L \to q$ and $a(L_1) \to p$. We assume *wlog* that the sets of non terminals N and N_1 of $\mathcal{G}$ and $\mathcal{G}_1$ respectively are disjoint. Let $q \in Q$ be a terminal symbol and let X_q be a fresh non terminal symbol. We consider below the CF grammar

$$\mathcal{G}_1{\downarrow}_q^{\mathcal{G}} := \big(Q, N_1 \uplus N \uplus \{X_q\}, I_1, \Gamma_1[q \leftarrow X_q] \cup \Gamma[q \leftarrow X_q] \cup \{X_q := q, X_q := I\}\big)$$

where $\Gamma[q \leftarrow X_q]$ denotes the set of production rules of Γ where every occurrence of the terminal symbol q is replaced by the non-terminal X_q. Using this construction, we can get rid of collapsing transitions in CF HA.

We assume that $\mathcal{A}$ is normalized with state set Q and for each $a \in \Sigma$ and $p \in Q$, we let $\mathcal{G}_{a,p}$ by the CF grammar generating the language $L_{a,p}$ in the transition (assumed unique) $a(L_{a,p}) \to p$ of $\mathcal{A}$. In order to construct $\mathcal{A}'$ out of $\mathcal{A}$, we perform the following operation for every collapsing transition $L \to q$ of $\mathcal{A}$: (i.) delete $L \to q$ (ii.) for each $a \in \Sigma$ and $p \in Q$, replace $\mathcal{G}_{a,p}$ by $\mathcal{G}_{a,p}{\downarrow}_q^{\mathcal{G}}$ where $\mathcal{G}$ is a CF grammar generating L. $\Box$

B Appendix: proof of Theorem 1

In this proof and the following, we describe *finite automata* for the horizontal languages of HA transitions as tuples $B = (\Sigma, S, i, F, \Gamma)$, where Σ is the finite input alphabet, S is a finite set of states, i is the initial state, $F \subseteq S$ is the set of final states and $\Gamma \subseteq S \times (\Sigma \cup \{\varepsilon\}) \times S$ is the set of transitions and ε-transitions. For $s, s' \in S$, we write $s \xrightarrow[B]{\varepsilon} s'$ to express that s' can be reached from s by a sequence of ε-transitions of B, and $s \xrightarrow{a_1 \ldots a_n} s'$, for $a_1, \ldots, a_n \in \Sigma$, if there exists $2(n+1)$ states $s_0, s'_0, \ldots, s_n, s'_n \in S$ with $s_0 = s$, $s_n \xrightarrow[B]{\varepsilon} s'$ and $0 \leq i < n$, $s_i \xrightarrow[B]{\varepsilon} s'_i$ and $(s'_i, \sigma_{i+1}, s_{i+1}) \in \Gamma$.

Theorem 1. *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU, for all HA language L, $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of an HA of size polynomial and which can be constructed in PTIME on the size of $\mathcal{R}$, $\mathcal{A}$ and an HA of language L.*

Proof. Let $\mathcal{A} = (\Sigma, P, P^{\mathsf{f}}, \Theta)$ and let $\mathcal{A}_L = (\Sigma, Q_L, Q_L^{\mathsf{f}}, \Delta_L)$ recognize L. We assume that both $\mathcal{A}$ and $\mathcal{A}_L$ are normalized and that their state sets P and Q_L are disjoint. We construct a HA $\mathcal{A}' = (P \uplus Q_L, Q_L^{\mathsf{f}}, \Delta')$ recognizing $post^*_{\mathcal{R}/\mathcal{A}}(L)$. For each $a \in \Sigma$, $q \in Q_L$, let $L_{a,q}$ be the regular language in the transition (assumed unique) $a(L_{a,q}) \to q \in \Delta_L$, and let $B_{a,q} = \big(Q_L, S_{a,q}, i_{a,q}, \{f_{a,q}\}, \Gamma_{a,q}\big)$ be finite automaton recognizing $L_{a,q}$. It has input alphabet Q_L, set of states $S_{a,q}$, initial state $i_{a,q} \in S_{a,q}$, final state $f_{a,q} \in S_{a,q}$ (that we assume unique wlog) and set of transition rules $\Gamma_{a,q} \subseteq S_{a,q} \times Q_L \times S_{a,q}$. The sets of states $S_{a,q}$ are assumed pairwise disjoint. Let S be the disjoint union of all $S_{a,q}$ for all $a \in \Sigma$ and $q \in Q_L$.

For the construction of Δ', we develop a set of transition rules $\Gamma' \subseteq S \times (P \cup Q_L) \times S$. Initially, we let Γ' be the union Γ_0 of all $\Gamma_{a,q}$ for $a \in \Sigma$, $q \in Q_L$, and we complete Γ' iteratively by analyzing the different cases of update rules of $\mathcal{R}/\mathcal{A}$. At each step, for each $a \in \Sigma$ and $q \in Q_L$, we let $B'_{a,q}$ be the automaton $(P \cup Q_L, S, i_{a,q}, \{f_{a,q}\}, \Gamma')$. For the sake of conciseness we make no distinction between an automaton $B'_{a,q}$ and its language $L(B'_{a,q})$.

REN: for every $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$ and $q \in Q_L$, we add two ε-transitions $(i_{b,q}, \varepsilon, i_{a,q})$ and $(f_{a,q}, \varepsilon, f_{b,q})$ to Γ'.

$\mathsf{INS}_{\mathsf{first}}$: for every $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$ and $q \in Q_L$, we add one looping transition $(i_{a,q}, p, i_{a,q})$ to Γ'.

$\mathsf{INS}_{\mathsf{last}}$: for every $a(x) \to a(x\,p) \in \mathcal{R}/\mathcal{A}$ and $q \in Q_L$, we add one looping transition rule $(f_{a,q}, p, f_{a,q})$ to Γ'.

$\mathsf{INS}_{\mathsf{into}}$: for every $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$ and $s \in S$ reachable from $i_{a,q}$ using the transitions of Γ', we add one looping transition rule (s, p, s) to Γ'.

$\mathsf{INS}_{\mathsf{left}}$: for every $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$ and state $s \in S$ such that $L(B'_{a,q}) \neq \emptyset$ and there exists a transition $(s, q, s') \in \Gamma'$, we add one looping transition (s, p, s) to Γ'.

$\mathsf{INS}_{\mathsf{right}}$: for every $a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$ and $s' \in S$ such that $L(B'_{a,q}) \neq \emptyset$ and there exists a transition $(s, q, s') \in \Gamma'$, we add one looping transition (s', p, s') to Γ'.

RPL: for every $a(x) \to p \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$, and $s, s' \in S$ such that $L(B'_{a,q}) \neq \emptyset$, and there exists a transition $(s, q, s') \in \Gamma'$, we add one transition (s, p, s') to Γ'.

DEL: for every $a(x) \to () \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$, and $s, s' \in S$ such that $L(B'_{a,q}) \neq \emptyset$, and there exists a transition $(s, q, s') \in \Gamma'$, we add one ε-transition (s, ε, s') to Γ'.

We iterate the above operations until a fixpoint is reached (only a finite number of transition can be added to Γ' this way). Finally, we let $\Delta' := \Theta \cup \big\{a\big(B'_{a,q}\big) \to q \;\big|\; a \in \Sigma, q \in Q, L(B'_{a,q}) \neq \emptyset\big\}$. Let us show now that $L(\mathcal{A}') = post^*_{\mathcal{R}/\mathcal{A}}(L)$.

Lemma 2 *$L(\mathcal{A}') \subseteq post^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show more generally that for all $t \in L(\mathcal{A}', q)$, $q \in Q_L$, there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}]{*} t$. The proof is by induction on the multiset $\mathcal{M}$ of the applications of horizontal transitions of Γ' not in Γ_0 in a run of $\mathcal{A}'$ on t leading to state q.

Base case. If all the horizontal transitions are in Γ_0, then by construction $t \in L(\mathcal{A}_L, q)$ and we are done.

Induction step. We analyse the cases causing the addition of a transition of $\Gamma' \setminus \Gamma_0$.

REN : let $t \in L(\mathcal{A}', q)$ ($q \in Q_L$), and assume that an ε-transition $(i_{b,q}, \varepsilon, i_{a,q})$ is used in a run of $\mathcal{A}'$ on t, and that this ε-transition was added to Γ' because $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$. Let

$$t = t[b(h)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

be a reduction of $\mathcal{A}'$ such that the above ε-transition is involved in the step $t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0]$, where the the transition $b(B_{b,q_0}) \to q_0$ is applied. Hence $q_1 \dots q_n \in L(B_{b,q_0})$, with $i_{b,q} \xrightarrow[B_{b,q_0}]{q_1 \dots q_n} f_{b,q}$, and the first step in this computation is $(i_{b,q}, \varepsilon, i_{a,q})$. The last step must be $(f_{a,q}, \varepsilon, f_{b,q})$, using an ε-transition added to Γ' in the same step as $(i_{b,q}, \varepsilon, i_{a,q})$. By deleting these first and last steps, we get $i_{a,q} \xrightarrow[B_{a,q_0}]{q_1 \dots q_n} f_{a,q}$, hence $q_1 \dots q_n \in L(B_{a,q_0})$. Therefore, we have a reduction $t' = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ (hence $t' \in L(\mathcal{A}', q)$) with a measure $\mathcal{M}$ strictly smaller than the above reduction for the recognition of t. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Since $t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h)] = t$, with $a(x) \to b(x)$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

$\mathsf{INS_{first}}$: let $t \in L(\mathcal{A}', q)$ ($q \in Q_L$), and assume that an transition $(i_{a,q}, p, i_{a,q})$ is used in a run of $\mathcal{A}'$ on t, and that this transition was added to Γ' because $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$. Let

$$t = t[a(t_p h)] \xrightarrow[\mathcal{A}']{*} t[a(p\, q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

be a reduction of $\mathcal{A}'$, with $t_p \in L(\mathcal{A}, p)$, such that the above transition is involved in the step $t[a(p\, q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0]$, where the the transition $b(B_{a,q_0}) \to q_0$ is applied. Hence $p\, q_1 \dots q_n \in L(B_{a,q_0})$, with $i_{a,q} \xrightarrow[B_{a,q_0}]{p\, q_1 \dots q_n} f_{a,q}$, and the first step in this computation is $(i_{a,q}, p, i_{a,q})$. By deleting this first step, we get $i_{a,q} \xrightarrow[B_{a,q_0}]{q_1 \dots q_n} f_{a,q}$, hence $q_1 \dots q_n \in L(B_{a,q_0})$. Therefore, we have a reduction $t' = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ (hence $t' \in L(\mathcal{A}', q)$) with a measure $\mathcal{M}$ strictly smaller than the above reduction for the recognition of t. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}]{*} t'$. Since $t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(t_p h)] = t$, with $a(x) \to b(x)$, we conclude $u \xrightarrow[\mathcal{R}]{*} t$.

$\mathsf{INS_{last}}$: this case is similar to the previous one.

$\mathsf{INS_{into}}$: let $t \in L(\mathcal{A}', q)$ ($q \in Q_L$), and assume that an transition (s, p, s) is used in a run of $\mathcal{A}'$ on t, and that this transition was added to Γ' because $a(xy) \to a(xpy) \in \mathcal{R}/\mathcal{A}$. Let

$$t = t[a(h\, t_p\, \ell)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n\, p\, q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

be a reduction of $\mathcal{A}'$, with $t_p \in L(\mathcal{A}, p)$, such that the above transition (s, p, s) is involved in the step $t[a(q_1 \ldots q_n\, p\, q'_1 \ldots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0]$, where the transition $b(B_{a,q_0}) \rightarrow q_0$ is applied. More precisely, assume that $q_1 \ldots q_n\, p\, q'_1 \ldots q'_m \in L(B_{a,q_0})$, because $i_{a,q} \xrightarrow[B_{a,q_0}]{q_1 \ldots q_n} s \xrightarrow[B_{a,q_0}]{p} s \xrightarrow[B_{a,q_0}]{q'_1 \ldots q'_m} f_{a,q}$. By deleting the middle step (s, p, s), we get $i_{a,q} \xrightarrow[B_{a,q_0}]{q_1 \ldots q_n\, q'_1 \ldots q'_m} f_{a,q}$, hence $q_1 \ldots q_n\, q'_1 \ldots q'_m \in L(B_{a,q_0})$. Therefore, we have a reduction $t' = t[a(h\ell)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \ldots q_n\, q'_1 \ldots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ (hence $t' \in L(\mathcal{A}', q)$) with a measure $\mathcal{M}$ strictly smaller than the above reduction for the recognition of t. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Since $t' = t[a(h\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(h\, t_p\, \ell)] = t$, with $a(xy) \rightarrow b(xpy)$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

$\mathsf{INS}_{\mathsf{left}}$: let $t \in L(\mathcal{A}', q)$ $(q \in Q_L)$, and assume that an transition (s, p, s) is used in a run of $\mathcal{A}'$ on t, and that this transition was added to Γ' because $a(x) \rightarrow p\, a(x) \in \mathcal{R}/\mathcal{A}$ and because there exists $(s, q_0, s') \in \Gamma'$ for some $q_0 \in Q_L$ with $L(B_{a,q_0}) \neq \emptyset$. Let

$$t = t[t_p\, a(h)] \xrightarrow[\mathcal{A}']{*} t[pq_0] \xrightarrow[\mathcal{A}']{*} q$$

be a reduction of $\mathcal{A}'$, with $t_p \in L(\mathcal{A}, p)$, involving the transition (s, p, s) in $s \xrightarrow[B_{b,q'}]{pq_0} s'$, for some b. Removing the transition (s, p, s), we have $s \xrightarrow[B_{b,q'}]{q_0} s'$ and a reduction $t' = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ (meaning $t' \in L(\mathcal{A}', q)$) with a measure $\mathcal{M}$ strictly smaller than the above reduction for the recognition of t. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Since $t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[t_p\, a(h)] = t$, with $a(x) \rightarrow p, a(x)$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

$\mathsf{INS}_{\mathsf{right}}$: this case is similar to the previous one.

RPL : let $t \in L(\mathcal{A}', q)$ $(q \in Q_L)$, and assume that a horizontal transition (s, p, s') is used in a run of $\mathcal{A}'$ on t, and that this transition was added to Γ' because $a(x) \rightarrow p \in \mathcal{R}/\mathcal{A}$ and because there exists $(s, q_0, s') \in \Gamma'$ for some $q_0 \in Q_L$ such that $L(B'_{a,q_0}) \neq \emptyset$. Let

$$t = t[t_p] \xrightarrow[\mathcal{A}']{*} t[p] \xrightarrow[\mathcal{A}']{*} q$$

be a reduction of $\mathcal{A}'$, with $t_p \in L(\mathcal{A}, p)$, involving the added transition (s, p, s') in $s \xrightarrow[B_{b,q'}]{p} s'$, for some b and some $q' \in Q_L$. Replacing the transition (s, p, s') with (s, q_0, s'), we obtain $s \xrightarrow[B_{b,q'}]{q_0} s'$ and a reduction $t' = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ (meaning $t' \in L(\mathcal{A}', q)$). The measure $\mathcal{M}$ of this later reduction is strictly smaller than the above reduction for the recognition of t, because the transition (s, q_0, s') belongs to Γ_0 (no such transition can be added by the above procedure). By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Since $t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[t_p] = t$, with $a(x) \rightarrow p$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

DEL : let $t \in L(\mathcal{A}', q)$ $(q \in Q_L)$, and assume that a horizontal transition (s, ε, s') is used in a run of $\mathcal{A}'$ on t, and that this transition was added to Γ'

because $a(x) \to () \in \mathcal{R}/\mathcal{A}$ and because there exists $(s, q_0, s') \in \Gamma'$ for some $q_0 \in Q_L$ such that $L(B'_{a,q_0}) \neq \emptyset$. Let us replace this ε-transition (s, ε, s') with (s, q_0, s') in a reduction $t \xrightarrow[\mathcal{A}']{*} q$, we obtain a reduction

$$t' = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q.$$

It means that $t' \in L(\mathcal{A}', q)$. The measure $\mathcal{M}$ of this later reduction is strictly smaller than the above reduction for the recognition of t, because the transition (s, q_0, s') belongs to Γ_0 (no such transition can be added by the above procedure). By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Since $t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t$, with $a(x) \to ()$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

(end Lemma direction $\subseteq$) □

Lemma 3 *$L(\mathcal{A}') \supseteq post^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show that for all $t \in L$, if $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$, then $u \in L(\mathcal{A}')$, by induction on the length of the rewrite sequence.

Base case (0 rewrite steps). In this case, $u = t \in L$ and we are done since $L = L(\mathcal{A}_L) \subseteq L(\mathcal{A}')$ by construction.

Induction step. Assume that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{+} u$ with $t \in L$. We analyse the type of rewrite rule used in the last rewrite step.

REN**.** The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h)] = t.$$

By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L(B'_{a,q_0})$, i.e. $i_{a,q_0} \xrightarrow[B'_{a,q_0}]{q_1 \dots q_n} f_{a,q_0}$. By construction, the ε-transitions $(i_{b,q_0}, \varepsilon, i_{a,q_0})$ and $(f_{a,q_0}, \varepsilon, f_{b,q_0})$ have been added to Γ'. Hence $i_{b,q_0} \xrightarrow[B'_{b,q_0}]{q_1 \dots q_n} f_{b,q_0}$ and $q_1 \dots q_n \in L(B'_{b,q_0})$. Therefore there exists a reduction sequence: $t = t[b(h)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ and $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{first}}$**.** The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to a(p\, x) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(t_p h)] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L(B'_{a,q_0})$, i.e. $i_{a,q_0} \xrightarrow[B'_{a,q_0}]{q_1 \dots q_n} f_{a,q_0}$. By construction, the transition $(i_{a,q_0}, p, i_{a,q_0})$ has been added to Γ'. Hence $i_{a,q_0} \xrightarrow[B'_{a,q_0}]{p} i_{a,q_0} \xrightarrow[B'_{a,q_0}]{q_1 \dots q_n} f_{b,q_0}$, i.e. $p\, q_1 \dots q_n \in L(B'_{a,q_0})$ and there exists a reduction sequence

$$t = t[a(t_p\, h)] \xrightarrow[\mathcal{A}']{*} t[a(p\, q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L.$$

It follows that $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{last}}$. The case where the last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to a(x\,p) \in \mathcal{R}/\mathcal{A}$, with $p \in P$ is similar to the previous one.

$\mathsf{INS}_{\mathsf{into}}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(h\,t_p\,\ell)] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h\ell)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h\ell)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n\, q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n\, q'_1 \dots q'_m \in L(B'_{a,q_0})$, i.e. $i_{a,q_0} \xrightarrow[B'_{a,q_0}]{q_1 \dots q_n} s \xrightarrow[B'_{a,q_0}]{q'_1 \dots q'_m} f_{a,q_0}$ for some state $s \in S$. By construction, the looping transition (s, p, s) has been added to Γ'. Hence $i_{a,q_0} \xrightarrow[B'_{a,q_0}]{q_1 \dots q_n} s \xrightarrow[B'_{a,q_0}]{p} s \xrightarrow[B'_{a,q_0}]{q'_1 \dots q'_m} f_{a,q_0}$, i.e. $q_1 \dots q_n\, p\, q'_1 \dots q'_m \in L(B'_{a,q_0})$ and there exists a reduction sequence

$$t = t[a(h\,t_p\,\ell)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n\, p\, q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L.$$

It follows that $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{left}}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to p\, a(x) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[t_p\, a(h)] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$. Hence $L(B'_{a,q_0}) \neq \emptyset$ and at some point of the reduction, a transition $(s, q_0, s') \in \Gamma'$ is involved. By construction, the transition (s, p, s) has been added to Γ'. Hence there exists a reduction sequence $t = t[t_p\, a(h)] \xrightarrow[\mathcal{A}']{*} t[p\, q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$. It follows that $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{right}}$. The case where the last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to a(x)\, p \in \mathcal{R}/\mathcal{A}$, with $p \in P$ is similar to the previous one.

RPL. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to p \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[t_p] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$. Hence $L(B'_{a,q_0}) \neq \emptyset$ and at some point of the reduction, a transition $(s, q_0, s') \in \Gamma'$ is applied. By construction, the transition (s, p, s') has been added to Γ', and there exists a reduction sequence $t = t[t_p] \xrightarrow[\mathcal{A}']{*} t[p] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$. It follows that $t \in L(\mathcal{A}')$.

DEL. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to () \in \mathcal{R}/\mathcal{A}$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[()] = t.$$

By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$. Hence $L(B'_{a,q_0}) \neq \emptyset$ and at some point of the reduction, a transition $(s, q_0, s') \in \Gamma'$ is applied. By construction, the ε-transition (s, ε, s') has been added to Γ', and there exists a reduction sequence $t \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$, hence $t \in L(\mathcal{A}')$.

(end Lemma direction $\supseteq$) □

(end of the proof of Theorem 1) □

C Appendix: proof of Theorem 2

Theorem 2. *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU+, for all CF-HA term language L, $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of an CF-HA of size polynomial and which can be constructed in PTIME on the size of $\mathcal{R}$, $\mathcal{A}$ and an CF-HA recognizing L.*

Proof. Let $\mathcal{A} = (P, P^\mathsf{f}, \Theta)$ and let us assumed that it is normalized. Let $\mathcal{A}_L = (Q_L, Q^\mathsf{f}_L, \Delta_L)$ be a CF-HA recognizing L, normalized and without collapsing transitions (this can be assumed thanks to Lemma 1) The state sets P and Q_L are assumed disjoint. We shall construct a CF-HA extended with collapsing transitions $\mathcal{A}' = (P \uplus Q_L, Q^\mathsf{f}_L, \Delta')$ recognizing $post^*_{\mathcal{R}/\mathcal{A}}(L)$. The set of transitions Δ' is constructed starting from $\Delta_L \cup \Theta$ and analysing the different cases of update rules.

For each $a \in \Sigma$, $q \in Q_L$, let $L_{a,q}$ be the context-free language in the transition (assumed unique) $a(L_{a,q}) \to q \in \Delta_L$, and let $\mathcal{G}_{a,q} = (Q_L, N_{a,q}, I_{a,q}, \Gamma_{a,q})$ be a CF grammar in Chomski normal form generating $L_{a,q}$. It has alphabet (set of terminal symbols) Q_L, set of non terminal symbols $N_{a,q}$, initial non-terminal $I_{a,q} \in N_{a,q}$, and set of production rules $\Gamma_{a,q}$. The sets of non-terminals $N_{a,q}$ are assumed pairwise disjoint.

Let us consider one new non-terminal $I'_{a,q}$ for each $a \in \Sigma$ and $q \in Q_L$. Each of these non terminals aims at becoming the initial non terminal of the CF grammar in the transition associated to a and q in Δ'. For technical convenience, we also add one new non terminal X_p for each $p \in P$. For the construction of Δ', we shall construct below a set C' of collapsing transitions, initially empty, and a set Γ' of production rules of CF grammar over the set of terminal symbols in $P \cup Q_L$ and the set of non terminals

$$\mathcal{N} = \bigcup_{a \in \Sigma, q \in Q} \big(N_{a,q} \cup \{I'_{a,q}\}\big) \cup \{X_p \mid p \in P\}.$$

Initially, we let $\Gamma' = \Gamma'_0 := \bigcup_{a \in \Sigma, q \in Q} \big(P_{a,q} \cup \{I'_{a,q} := I_{a,q}\}\big) \cup \{X_p := p \mid p \in P\}.$

We now proceed by analysis of the rewrite rules of $\mathcal{R}/\mathcal{A}$ for the completion of Γ' and C'. At each step, for each $a \in \Sigma$ and $q \in Q_L$, we let $\mathcal{G}'_{a,q}$ be the CF grammar $(P \cup Q_L, \mathcal{N}, I'_{a,q}, \Gamma')$, and let $L'_{a,q} = L(\mathcal{G}'_{a,q})$. The production rules of Γ' remain in Chomski normal form after each completion step.

REN: for every $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$, we add one production rule $I'_{b,q} := I'_{a,q}$ to Γ'.

$\mathsf{INS}'_{\mathsf{first}}$: for every $a(x) \to b(p\,x) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$, we add one production rule $I'_{b,q} := X_p I'_{a,q}$ to Γ'.

$\mathsf{INS}'_{\mathsf{last}}$: for every $a(x) \to b(x\,p) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$, we add one production rule $I'_{b,q} := I'_{a,q} X_p$ to Γ'.

$\mathsf{INS}_{\mathsf{into}}$: for every $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, $q \in Q_L$ and every $N \in \mathcal{N}$ reachable from $I'_{a,q}$ using the rules of Γ', we add two production rules $N := N X_p$ and $N := X_p N$.

$\mathsf{INS}_{\mathsf{left}}$: for every $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, and $q \in Q_L$ such that $L'_{a,q} \neq \emptyset$, we add one collapsing transition $p\,q \to q$ to C'.

$\mathsf{INS}_{\mathsf{right}}$: for every $a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}$, and $q \in Q_L$ such that $L'_{a,q} \neq \emptyset$, we add one collapsing transition $q\,p \to q$ to C'.

RPL': for every $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$, with $n \geq 0$, and $q \in Q_L$ such that $L'_{a,q} \neq \emptyset$, we add one collapsing transition $p_1 \dots p_n \to q$ to C'.

DEL: for every $a(x) \to () \in \mathcal{R}/\mathcal{A}$ and $q \in Q_L$ such that $L'_{a,q} \neq \emptyset$, we add one collapsing transition $() \to q$ to C'.

Note that $\mathsf{INS}_{\mathsf{first}}$, $\mathsf{INS}_{\mathsf{last}}$, RPL are special cases of respectively $\mathsf{INS}'_{\mathsf{first}}$, $\mathsf{INS}'_{\mathsf{last}}$, RPL'.

We iterate the above operations until a fixpoint is reached. Indeed, only a finite number of production and collapsing rules can be added. Finally, we let

$$\Delta' := \Theta \cup \big\{ a(L'_{a,q}) \to q \;\big|\; a \in \Sigma, q \in Q, L'_{a,q} \neq \emptyset \big\} \cup C' \cup \{ L'_{a,q} \to q \mid a(x) \to x \in \mathcal{R}/\mathcal{A}, L'_{a,q} \neq \emptyset \}.$$

We show that $L(\mathcal{A}') = post^*_{\mathcal{R}/\mathcal{A}}(L)$. It follows that $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is a CF-HA language by Lemma 1.

Lemma 4 *$L(\mathcal{A}') \subseteq post^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show more generally that for all $t \in L(\mathcal{A}', q)$, $q \in Q_L$, there exists $u \in L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. The proof is by induction on the number of applications of collapsing transitions in the reduction $t \xrightarrow[\mathcal{A}']{*} q$.

Base case. For the base case (no collapsing transition applied), we make a second induction on the number of application of production rules of $\Gamma' \setminus \Gamma_0$ in the derivations, by the grammars $\mathcal{G}'_{a,q_0}$, for the generations of the sequences of states $q_1 \dots q_n \in Q^*$ used in moves of $\mathcal{A}'$ of the form $u[a(q_1 \dots q_n)] \to u[q_0]$ in the reduction $t \xrightarrow[\mathcal{A}']{*} q$. Let us note $\vdash$ the relation of derivation using the production rules of Γ', and $\vdash^*$ its transitive closure.

Intuitively every application of a production rule of $\Gamma' \setminus \Gamma_0$ corresponds to a rewrite step with a rule of $\mathcal{R}/\mathcal{A}$ in the rewrite sequence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$, according to the above construction cases.

Base case (second induction). For the base case, no production rule of $\Gamma' \setminus \Gamma_0$ is applied. It means that $t \xrightarrow[\mathcal{A}_L]{*} q$ (every CF grammar derivation in the reduction $t \xrightarrow[\mathcal{A}']{*} q$ starts with $I'_{a,q} \vdash I_{a,q}$) and we let $u = t$.

Induction step (second induction). Assume that the reduction $t \xrightarrow[\mathcal{A}']{*} q$ has the form

$$t = t[a(t_1 \dots t_n)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

where $t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0]$ is one transition such that the derivation of $I'_{a,q_0} \vdash^* q_1 \dots q_n$ by $\mathcal{G}'_{a,q_0}$ involves one production rule of $\Gamma' \setminus \Gamma_0$. We shall analyse below the different cases of rewrite rules of $\mathcal{R}/\mathcal{A}$ (rules of type XACU_1) which permitted the addition of this production rule of $\Gamma' \setminus \Gamma_0$. Let us first note before that we can assume that for every $i \leq n$, $t_i \xrightarrow[\mathcal{A}']{*} q_i$ because no collapsing transition are used, by hypothesis. Hence, together with the above hypothesis, it follows that $t_i \in L(\mathcal{A}, q_i)$ for all $i \leq n$.

Case REN. We have $I'_{a,q_0} \vdash I'_{b,q_0} \vdash^* q_1 \dots q_n$, and the first production rule used in this derivation, $I'_{a,q_0} := I'_{b,q_0}$, was added because there exists a rule $b(x) \to a(x) \in \mathcal{R}/\mathcal{A}$. It follows that $I'_{b,q_0} \vdash^* q_1 \dots q_n$ and then that

$$s = t[b(t_1 \dots t_n)] \xrightarrow[\mathcal{A}']{*} t\big[b(q_1 \dots q_n)\big] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q.$$

Hence, by induction hypothesis, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s = t[b(t_1 \dots t_n)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t = t[a(t_1 \dots t_n)]$ using $b(x) \to a(x) \in \mathcal{R}/\mathcal{A}$. Hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Case $\mathsf{INS}'_{\mathsf{first}}$. We have $I'_{b,q_0} \vdash X_p I'_{a,q_0} \vdash^* q_1 \dots q_n$, and the first production rule used in this derivation, $I'_{b,q_0} := X_p I'_{a,q_0}$ was added because there exists a rule $a(x) \to b(px) \in \mathcal{R}/\mathcal{A}$. By construction, it follows that $q_1 = p$ and $I'_{a,q_0} \vdash^* q_2 \dots q_n$, and

$$s = t[a(t_2 \dots t_n)] \xrightarrow[\mathcal{A}']{*} t\big[a(q_2 \dots q_n)\big] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q.$$

By induction hypothesis, applied to the above reduction, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s = t[a(t_2 \dots t_n)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t = t[b(t_1 \dots t_n)]$ using $a(x) \to a(px) \in \mathcal{R}/\mathcal{A}$, because $t_1 \in L(\mathcal{A}, p)$. Hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Case $\mathsf{INS}'_{\mathsf{last}}$. This case is similar to the previous one.

Case $\mathsf{INS}_{\mathsf{into}}$. We have $I'_{a,q_0} \vdash^* \alpha N \beta \vdash \alpha N X_p \beta \vdash \alpha N p \beta \vdash^* q_1 \dots q_n$, and the production $N := N X_p$ was added because there exists a rule $a(xy) \to a(xpy) \in \mathcal{R}/\mathcal{A}$, and N is reachable from $I'_{a,q}$ using Γ'. It follows that there exists two integers $k < \ell \leq n$ such that $\alpha \vdash^* q_1 \dots q_k$ and $N X_p \vdash^* q_{k+1} \dots q_\ell$ (hence $q_\ell = p$) and $\beta \vdash^* q_{\ell+1} \dots q_n$ (if $\ell = n$ then this latter sequence is empty), and

$$s = t[a(t_1 \dots t_{\ell-1}\, t_{\ell+1} \dots t_n)] \xrightarrow[\mathcal{A}']{*} t\big[a(q_1 \dots q_{\ell-1}\, q_{\ell+1} \dots q_n)\big] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q.$$

By induction hypothesis, applied to the above reduction, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s = t[a(t_1 \ldots t_{\ell-1}\, t_{\ell+1} \ldots t_n)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t = t[a(t_1 \ldots t_n)]$ using the rewrite rule $a(xy) \to a(xpy)$, because $t_n \in L(\mathcal{A}, p)$. Hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Induction step (first induction). Assume that the reduction $t \xrightarrow[\mathcal{A}']{*} q$ has the form

$$t = t[t_1 \ldots t_n] \xrightarrow[\mathcal{A}']{*} t[q_1 \ldots q_n] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q \qquad (1)$$

such that there exists a collapsing transition $L' \to q \in \Delta'$ with $q_1 \ldots q_n \in L'$ and the first part of the reduction, $t \xrightarrow[\mathcal{A}']{*} t[q_1 \ldots q_n]$, involves no collapsing transition. It implies in particular that $t_i \in L(\mathcal{A}', q_i)$ for all $i \leq n$.

The collapsing transition $L' \to q$ belongs to C' (by hypothesis $\mathcal{A}_L$ and $\mathcal{A}$ do not contain collapsing transitions) and was added because of a rewrite rule of $\mathcal{R}/\mathcal{A}$ in $\mathsf{XACU+}$. We consider below the different possible cases for this addition.

Case $\mathsf{INS}_{\mathsf{left}}$. We have $n = 2$, $q_1 = p \in P$, $q_2 = q_0$ and the collapsing transition $pq_0 \to q_0$ has been added because there exists a rule $a(x) \to pa(x) \in \mathcal{R}/\mathcal{A}$. In this case, the reduction (1) is

$$t = t[t_1 t_2] \xrightarrow[\mathcal{A}']{*} t[pq_0] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

and we have $s = t[t_2] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$ because the first part of the reduction uses no collapsing transition. By induction hypothesis, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s \xrightarrow[\mathcal{R}/\mathcal{A}]{} t$ using the rewrite rule $a(x) \to pa(x)$, because $t_1 \in L(\mathcal{A}, p)$. Hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Case $\mathsf{INS}_{\mathsf{right}}$. This case is similar to the previous one.

Case RPL'. In this case, for all $i \leq n$, $q_i = p_i \in P$ and the collapsing transition $p_1 \ldots p_n \to q_0$ was added because there exists a rewrite rule $a(x) \to p_1 \ldots p_n \in \mathcal{R}/\mathcal{A}$ and $L'_{a,q_0} \neq \emptyset$. Hence there exists a term $a(h) \in L(\mathcal{A}', q_0)$, and

$$s = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

By induction hypothesis, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, using the rewrite rule $a(x) \to p_1 \ldots p_n$, $s \xrightarrow[\mathcal{R}/\mathcal{A}]{} t$ because $t_i \in L(\mathcal{A}, p_i)$ for all $i \leq n$. Hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Case DEL. In this case, $n = 0$ and the collapsing transition $() \to q_0$ was added to C' because there exists a rewrite rule $a(x) \to () \in \mathcal{R}/\mathcal{A}$ and $L'_{a,q_0} \neq \emptyset$. Let $a(h) \in L(\mathcal{A}', q_0)$, we have $s = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$. By induction hypothesis, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s \xrightarrow[\mathcal{R}/\mathcal{A}]{} t$ using the rewrite rule $a(x) \to ()$, and $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

Case $\mathsf{DEL_s}$. In this last case, the collapsing transition $L'_{a,q_0} \to q_0$ was added to Δ' because there exists a rewrite rule $a(x) \to x \in \mathcal{R}/\mathcal{A}$ and $L'_{a,q_0} \neq \emptyset$. We have

$$s = t[a(t_1 \dots t_n)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

because $q_1 \dots q_n \in L'_{a,q_0}$. By induction hypothesis, there exists $u \in L(\mathcal{A}, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} s$. Moreover, $s \xrightarrow[\mathcal{R}/\mathcal{A}]{} t$ using the rewrite rule $a(x) \to x$, and $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

(end Lemma direction $\subseteq$) $\square$

Lemma 5 *$L(\mathcal{A}') \supseteq post^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show that for all $u \in L$, if $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$, then $t \in L(\mathcal{A}')$, by induction on the length of the rewrite sequence.

Base case (0 rewrite steps). In this case, $u = t \in L$. We can note that $L \subseteq L(\mathcal{A}')$ because Γ' contains the production rule $I'_{a,q} := I_{a,q}$ for all $a \in \Sigma$, $q \in Q_L$. Hence, $t \in L(\mathcal{A}')$.

Induction step ($k+1$ rewrite steps). We analyse the type of rewrite rule used in the last rewrite step of $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

REN. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(h)] = t.$$

By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$, i.e. $q_1 \dots q_n$ can be generated by $\mathcal{G}'_{a,q_0}$, starting from I'_{a,q_0} and using the production rules of Γ'.
By construction, Γ' contains the production rule $I'_{b,q_0} := I'_{a,q_0}$. Hence $q_1 \dots q_n \in L'_{b,q_0}$: it can be generated by $\mathcal{G}'_{b,q_0}$, starting from I'_{b,q_0} and using the production rules of Γ'.

Hence $t = u[b(h)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$, i.e. $t \in L(\mathcal{A}')$.

$\mathsf{INS}'_\mathsf{first}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to b(p\,x) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(t_p h)] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$, i.e. $q_1 \dots q_n$ can be generated by $\mathcal{G}'_{a,q_0}$, starting from I'_{a,q_0} and using the production rules of Γ'.
By construction, Γ' contains the production rule $I'_{b,q_0} := X_p I'_{a,q_0}$. Hence $pq_1 \dots q_n$ is in L'_{b,q_0}. Hence $t = u[b(t_p h)] \xrightarrow[\mathcal{A}']{*} u[b(pq_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$, i.e. $t \in L(\mathcal{A}')$.

$\mathsf{INS}'_{\mathsf{last}}$. This case is similar to the above one.

$\mathsf{INS}_{\mathsf{into}}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(xy) \to a(xpy) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[a(h\, t_p\, \ell)] = t$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[a(h\ell)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$, i.e. $q_1 \dots q_n$ can be generated by $\mathcal{G}'_{a,q_0}$, starting from I'_{a,q_0} and using the production rules of Γ'.
By construction, Γ' contains the production rules $N := NX_p$ and $N := X_pN$ for all non terminal N reachable from I'_{q,q_0} using Γ'. Using one of these production rules, it is possible to generate $q_1 \dots q_j\, p\, q_{j+1} \dots q_n$ with $\mathcal{G}'_{a,q_0}$, starting from I'_{a,q_0} and using the production rules of Γ', where j is the length of h. Hence $t = u[a(h\, t_p\, \ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \dots q_j\, p\, q_{j+1} \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$, and $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{left}}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to pa(x) \in \mathcal{R}/\mathcal{A}$, with $p \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[t_p\, a(h)] = t.$$

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$.
By construction, $\mathcal{A}'$ contains a collapsing transition rule $pq_0 \to q_0$. Hence $t = u[t_p a(h)] \xrightarrow[\mathcal{A}']{*} u[pq_0] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$, *i.e.* $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{right}}$. This case is similar to the above one.

RPL'. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$, with $p_1, \dots, p_n \in P$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[t_1 \dots t_n] = t.$$

with $t_i \in L(\mathcal{A}, p_i)$ for all $i \leq n$. By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$.
Therefore, by construction, $\mathcal{A}'$ contains a collapsing transition rule $p_1 \dots p_n \to q_0$. Hence $t = u[t_1 \dots t_n] \xrightarrow[\mathcal{A}']{*} u[p_1 \dots p_n] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$, *i.e.* $t \in L(\mathcal{A}')$.

$\mathsf{DEL}_{\mathsf{s}}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to () \in \mathcal{R}/\mathcal{A}$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[()] = t.$$

By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$ with $q_1 \dots q_n \in L'_{a,q_0}$.
By construction, $\mathcal{A}'$ contains a collapsing transition rule $() \to q_0$. Hence $t = u[()] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$, *i.e.* $t \in L(\mathcal{A}')$.

$\mathsf{DEL_s}$. The last rewrite step of the sequence involves a rewrite rule of the form $a(x) \to x \in \mathcal{R}/\mathcal{A}$:

$$u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[h] = t.$$

By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence: $u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$ with $q_1 \ldots q_n \in L'_{a,q_0}$.
By construction, $\mathcal{A}'$ contains a collapsing transition rule $L'_{a,q_0} \to q_0$. Hence $t = u[h] \xrightarrow[\mathcal{A}']{*} u[q_1 \ldots q_n] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$, *i.e.* $t \in L(\mathcal{A}')$.

(end Lemma direction $\supseteq$) □

(end of the proof of Theorem 1) □

D Appendix: proof of Theorem 3

Theorem 3. *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ XACU+, for all HA language L, $pre^*_{\mathcal{R}/\mathcal{A}}(L)$ is a HA the language.*

Proof. Let $\mathcal{A} = (P, P^{\mathsf{f}}, \Theta)$, and let $\mathcal{A}_L = (Q_L, Q_L^{\mathsf{f}}, \Delta_L)$ be a HA recognizing L; both are assumed normalized. We also assume wlog that $\mathcal{A}_L$ is complete: for all term t, there exists a state q such that $t \in L(\mathcal{A}', q)$. Like in the proof of Theorem 1, we assume given, for each $a \in \Sigma$, $q \in Q_L$, a finite automaton $B_{a,q} = (Q_L, S_{a,q}, i_{a,q}, \{f_{a,q}\}, \Gamma_{a,q})$ recognizing the regular language $L_{a,q}$ in the transition $a(L_{a,q}) \to q \in \Delta_L$ (assumed unique).

We shall construct a finite sequence sequence of HA $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ whose final element's language is $pre^*_{\mathcal{R}/\mathcal{A}}(L)$, where for all $i \leq n$, $\mathcal{A}_i = (\Sigma, Q_L, Q_L^{\mathsf{f}}, \Delta_i)$. For the construction of the transition sets Δ_i, we consider a set $\mathcal{C}$ of finite automata over Q_L defined as the smallest set such that:

- $\mathcal{C}$ contains every $B_{a,q}$ for $a \in \Sigma$, $q \in Q_L$,
- for all $B \in \mathcal{C}$, $B = (Q_L, S, i, F, \Gamma) \in \mathcal{C}$ and all states $s, s' \in S$, the automaton $B_{s,s'} := (Q_L, S, s, \{s'\}, \Gamma)$ is in $\mathcal{C}$,
- for all $B \in \mathcal{C}$, $B = (Q_L, S, i, F, \Gamma) \in \mathcal{C}$, $q \in Q_L$ and all states $s, s' \in S$, the automata $(Q_L, S, i, F, \Gamma \cup \{\langle s, q, s'\rangle\})$ and $(Q_L, S, i, F, \Gamma \cup \{\langle s, \varepsilon, s'\rangle\})$, respectively denoted by $B + \langle s, q, s'\rangle$ and $B + \langle s, \varepsilon, s'\rangle$ also belong to $\mathcal{C}$.

Note that $\mathcal{C}$ is finite with this definition. For the sake of conciseness, we make no distinction below between a NFA $B \in \mathcal{C}$ and the language $L(B)$ recognized by B. Moreover, we assume that every $B \in \mathcal{C}$ has a unique final state denoted f_B and an initial state denoted i_B.

First, we let $\Delta_0 = \Delta_L$. The other Δ_i are constructed recursively by iteration of the following case analysis until a fixpoint is reached (only a finite number of transition can be added in the construction). In the construction we use an extension of the move relation of HA, from states to set of states (single states are considered as singleton sets): $a(L_1, \ldots, L_n) \hookrightarrow_{\Delta_i} q$ (where $L_1, \ldots, L_n \subseteq Q_L$ and $q \in Q_L$) iff there exists a transition $a(L) \to q \in \Delta_i$ such that $L_1 \ldots L_n \subseteq L$.

REN: if $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$ and $q \in Q_L$, such that $b(B) \hookrightarrow q$, then let $\Delta_{i+1} := \Delta_i \cup \{a(B) \to q\}$.

$\mathsf{INS}'_{\mathsf{first}}$: if $a(x) \to b(p\,x) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$ and $q, q_p \in Q_L$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$ and $b(q_p B) \hookrightarrow_{\Delta_i} q$, then $\Delta_{i+1} := \Delta_i \cup \{a(B) \to q\}$.

$\mathsf{INS}'_{\mathsf{last}}$: if $a(x) \to b(x\,p) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$ and $q, q_p \in Q_L$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$ and $b(B\,q_p) \hookrightarrow_{\Delta_i} q$, then $\Delta_{i+1} := \Delta_i \cup \{a(B) \to q\}$.

$\mathsf{INS}_{\mathsf{into}}$: if $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$, s, s' are states of B, and $q, q_p \in Q_L$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q_p} s'$, and $a(B) \hookrightarrow_{\Delta_i} q$ then $\Delta_{i+1} := \Delta_i \cup \big\{a(B + \langle s, \varepsilon, s'\rangle) \to q\big\}$.

$\mathsf{INS}_{\mathsf{left}}$: if $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q_p, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q_p q'} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$.

$\mathsf{INS}_{\mathsf{right}}$: if $a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q_p, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q' q_p} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$.

RPL': if $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q', q_1, \dots, q_n \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_j) \cap L(\mathcal{A}, p_j) \neq \emptyset$ for all $1 \leq j \leq n$, $s \xrightarrow[B]{q_1 \dots q_n} s'$ then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle)) \to q\big\}$.

DEL: if $a(x) \to () \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $B, B' \in \mathcal{C}$, s is a state of B, $q, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s\rangle) \to q\big\}$.

$\mathsf{DEL_s}$: if $a(x) \to x \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $B \in \mathcal{C}$, s, s' are states of B, $q, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B_{s,s'}) \hookrightarrow_{\Delta_i} q'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$.

Note that $\mathsf{INS}_{\mathsf{first}}$, $\mathsf{INS}_{\mathsf{last}}$, RPL are special cases of respectively $\mathsf{INS}'_{\mathsf{first}}$, $\mathsf{INS}'_{\mathsf{last}}$, RPL'. Since no state is added to the original automaton $\mathcal{A}_L$ and all the transition added involve horizontal languages of the set $\mathcal{C}$, which is finite, the iteration of the above operations terminates with an automaton $\mathcal{A}'$. Let us show that $L(\mathcal{A}') = pre^*_{\mathcal{R}/\mathcal{A}}(L)$.

Lemma 6 *$L(\mathcal{A}') \subseteq pre^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show more generally that for all $t \in L(\mathcal{A}', q)$, $q \in Q_L$, there exists $u \in L(\mathcal{A}_L, q)$ such that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. The proof is by induction on the measure $\mathcal{M}$ associating to a reduction $t \xrightarrow[\mathcal{A}']{*} q$ the multiset containing, for each transition rule $\rho \in \Delta_i$ with $i > 0$ used in the reduction, the index $min(j > 0 \mid \rho \in \Delta_j)$.

Base case. If $\mathcal{M}$ is empty, all the transition are in Δ_0. It means that $t \in L(\mathcal{A}_L, q)$ and we let $u = t$.

Induction step. Assume that we have a reduction by $\mathcal{A}'$ of the form

$$t = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q \tag{2}$$

(with $q_0 \in Q_L$, $q_1 \dots q_n \in L(B)$) and that the step $t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0]$ applies a transition $b(B) \to q_0$ added to Δ_{i+1} for some $i \geq 0$. We analyse the cases which permitted the addition of this transition to Δ_{i+1}.

REN: the transition $a(B) \to q_0$ was added to Δ_{i+1} because $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$ and $b(B) \hookrightarrow_{\Delta_i} q_0$. Hence, there exists a reduction

$$t' = t[b(h)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (2), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h)] = t'$, we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

$\mathsf{INS}'_{\mathsf{first}}$: the transition $a(B) \to q_0$ was added to Δ_{i+1} because $a(x) \to b(p\,x) \in \mathcal{R}/\mathcal{A}$, with $q_0, q_p \in Q_L$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$ and $b(q_p B) \hookrightarrow_{\Delta_i} q_0$. Hence, there exists a reduction

$$t' = t[b(t_p\, h)] \xrightarrow[\mathcal{A}']{*} t[b(q_p q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (2), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(t_p\, h)] = t'$, we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

$\mathsf{INS}'_{\mathsf{last}}$: this case is similar to the previous one.

$\mathsf{INS}_{\mathsf{into}}$: the transition is $a(B') \to q_0$ and was added to Δ_{i+1} because $a(xy) \to b(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$, s, s' are states of B, $q_0, q_p \in Q_L$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q_p} s'$, $b(B) \hookrightarrow_{\Delta_i} q_0$ and $B' = B + \langle s, \varepsilon, s' \rangle$. In this case, let $t = a(h\ell)$, and assume that the reduction (2) has the form

$$t = t[a(h\ell)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n\, q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q_1 \dots q_n\, q'_1 \dots q'_m \in L(B')$ by $i_{B'} \xrightarrow[B']{q_1 \dots q_n} s \xrightarrow[B']{\varepsilon} s' \xrightarrow[B']{q'_1 \dots q'_m} f_{B'}$ ($i_{B'}$ and $f_{B'}$ are resp. initial and final states of B'). Hence, by construction, we have $i_B \xrightarrow[B]{q_1 \dots q_n} s \xrightarrow[B]{q_p} s' \xrightarrow[B']{q'_1 \dots q'_m} f_B$ ($i_{B'} = i_B$ and $f_{B'} = f_B$) and there exists a reduction

$$t' = t[b(h\, t_p\, \ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n\, q_p\, q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (2), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\,\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\, t_p\, \ell)] = t'$, we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

From now on we assume that the reduction of t by $\mathcal{A}'$ has the form

$$t = t[b(h)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q \qquad (3)$$

with $q_1 \dots q_n \in L(B'')$, $q_0 \in Q_L$, and that the step $t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0]$ applies a transition $b(B'') \to q_0$ added to Δ_{i+1} for some $i \geq 0$ in one of the five cases.

$\mathsf{INS_{left}}$: the transition $b(B'') \to q_0$ was added to Δ_{i+1} because $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s, s' are states of B, $q_0, q_p, q'_0 \in Q_L$, such that $b(B) \to q_0 \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'_0$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q_p\, q'_0} s'$, and $B'' = B + \langle s, q'_0, s'\rangle$. In this case, let $t = b(ha(v)\ell)$, and assume that the above reduction (3) has the form

$$t = t[b(h\,a(v)\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_n\, q'_0\, q'_1 \ldots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q_1 \ldots q_n\, q'_1 \ldots q'_m \in L(B'')$ by $i_{B''} \xrightarrow[B'']{q_1 \ldots q_n} s \xrightarrow[B'']{q'_0} s' \xrightarrow[B'']{q'_1 \ldots q'_m} f_{B''}$ ($i_{B''}$ and $f_{B''}$ are resp. the initial and final states of B''). Hence, by construction, we have $i_B \xrightarrow[B]{q_1 \ldots q_n} s \xrightarrow[B]{q_p\, q'_0} s' \xrightarrow[B]{q'_1 \ldots q'_m} f_B$ ($i_{B''} = i_B$ and $f_{B''} = f_B$) and there exists a reduction

$$t' = t[b(h\, t_p\, a(v)\,\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_n\, q_p\, q'_0 q'_1 \ldots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (3), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\,t_p\,a(v)\ell)] = t'$, we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

$\mathsf{INS_{right}}$: this case is similar to the previous one.

$\mathsf{RPL'}$: the transition $b(B'') \to q_0$ has been added to Δ_{i+1} because $a(x) \to p_1 \ldots p_n \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s, s' are states of B, $q_0, q'_0, q_{p_1}, \ldots, q_{p_n} \in Q_L$, such that $b(B) \to q_0 \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'_0$, $L(\mathcal{A}_i, q_{p_j}) \cap L(\mathcal{A}, p_j) \neq \emptyset$ for all $j \leq n$, $s \xrightarrow[B]{q_{p_1} \ldots q_{p_n}} s'$, and $B'' = B + \langle s, q'_0, s'\rangle$. In this case, let $t = b(ha(v)\ell)$, and assume that the above reduction (3) has the form

$$t = t[b(h\,a(v)\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, q'_0\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q_1 \ldots q_m\, q'_1 \ldots q'_{m'} \in L(B'')$ by $i_{B''} \xrightarrow[B'']{q_1 \ldots q_m} s \xrightarrow[B'']{q'_0} s' \xrightarrow[B'']{q'_1 \ldots q'_{m'}} f_{B''}$ ($i_{B''}$ and $f_{B''}$ are resp. initial and final states of B''). Hence, by construction, we have $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{q_{p_1} \ldots q_{p_n}} s' \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$ ($i_{B''} = i_B$ and $f_{B''} = f_B$) and there exists a reduction, with for all $j \leq n$, $t_j \in L(\mathcal{A}_i, q_{p_j}) \cap L(\mathcal{A}, p_j)$,

$$t' = t[b(h\,t_1 \ldots t_n\,\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, q_{p_1} \ldots q_{p_n}\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (3), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\,t_1 \ldots t_n\,\ell)] = t'$, using the rule $a(x) \to p_1 \ldots p_n$, and we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

DEL: the transition $b(B'') \to q_0$ has been added to Δ_{i+1} because $a(x) \to () \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s is a state of B, $q_0, q'_0 \in Q_L$, such that $b(B) \to q_0 \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'_0$, and $B'' = B + \langle s, q'_0, s\rangle$. In this case, let $t = b(h\,a(v)\ell)$, and assume that the above reduction (3) has the form

$$t = t[b(h\,a(v)\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, q'_0\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q_1 \ldots q_m\, q'_1 \ldots q'_{m'} \in L(B'')$ by $i_{B''} \xrightarrow[B'']{q_1 \ldots q_m} s \xrightarrow[B'']{q'_0} s \xrightarrow[B'']{q'_1 \ldots q'_{m'}} f_{B''}$ ($i_{B''}$ and $f_{B''}$ are resp. initial and final states of B''). Hence, by construction, we have $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$ ($i_{B''} = i_B$ and $f_{B''} = f_B$) and there exists a reduction

$$t' = t[b(h\,\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (3), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\,\ell)] = t'$, and we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

DEL_s**:** the transition $b(B'') \to q_0$ has been added to Δ_{i+1} because $a(x) \to x \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$, s, s' are states of B, $q_0, q'_0 \in Q_L$, such that $b(B) \to q_0 \in \Delta_i$, $a(B_{s,s'}) \hookrightarrow_{\Delta_i} q'_0$, and $B'' = B + \langle s, q'_0, s' \rangle$. In this case, let $t = b(ha(v)\ell)$, and assume that the above reduction (3) has the form

$$t = t[b(h\,a(v)\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, a(v_1 \ldots v_k)\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, q'_0\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q_1 \ldots q_m\, q'_0, q'_1 \ldots q'_{m'} \in L(B'')$ by $i_{B''} \xrightarrow[B'']{q_1 \ldots q_m} s \xrightarrow[B'']{q'_0} s' \xrightarrow[B'']{q'_1 \ldots q'_{m'}} f_{B''}$ ($i_{B''}$ and $f_{B''}$ are resp. initial and final states of B'') and $s \xrightarrow[B_{s,s'}]{v_1 \ldots v_k} s'$.

Hence, by construction, we have $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{v_1 \ldots v_k} s' \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$ ($i_{B''} = i_B$ and $f_{B''} = f_B$) and there exists a reduction

$$t' = t[b(h\,v\,\ell)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \ldots q_m\, v_1 \ldots v_k\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}$ strictly smaller than for (3), by hypothesis. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\,v\,\ell)] = t'$, we conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

Note that $\mathsf{INS}_\mathsf{first}$, $\mathsf{INS}_\mathsf{last}$, RPL, were not considered above because they are special cases of respectively $\mathsf{INS}'_\mathsf{first}$, $\mathsf{INS}'_\mathsf{last}$, RPL'.

(end Lemma direction $\subseteq$) $\square$

Lemma 7 *$L(\mathcal{A}') \supseteq pre^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show that for all $t \in L$, if $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$, then $u \in L(\mathcal{A}')$, by induction on the length of the rewrite sequence.

Base case (0 rewrite steps). In this case, $u = t \in L$ and we are done since $L = L(\mathcal{A}_L) \subseteq L(\mathcal{A}')$ by construction.

Induction step. Assume that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{+} t$, we analyse the type of rewrite rule used in the first rewrite step.

REN**.** Assume that $u = u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. By induction hypothesis, $u_1 = u[b(h)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence $u_1 = u[b(h)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \ldots q_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f}$ where $q, q_1, \ldots, q_n \in Q_L$, $q^\mathsf{f} \in Q^\mathsf{f}_L$, and a transition $a(B) \to q$ has been added to $\mathcal{A}'$, with $q_1 \ldots q_n \in B$. It follows that $u = u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f}$, hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}'_{\mathsf{first}}$. Assume that $u = u[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(t_p\,h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$ for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u_1 = u[b(t_p\,h)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(t_p\,h)] \xrightarrow[\mathcal{A}']{*} u[b(q_p q_1 \ldots q_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_p, q_1, \ldots, q_n \in Q_L$, $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$. Hence $L(\mathcal{A}', q_p) \cap L(\mathcal{A}, p)$ is not empty because it contains t_p, and a transition $a(B) \to q$ has been added to $\mathcal{A}'$, with $q_1 \ldots q_n \in B$. It follows that $u = u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}'_{\mathsf{last}}$. This case is similar to the previous one.

$\mathsf{INS}_{\mathsf{into}}$. Assume that $u = u[a(h\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[a(h\,t_p\,\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$ for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u_1 = u[a(h\,t_p\,\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u_1 = u[a(h\,t_p\,\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q_p q'_1 \ldots q'_n)] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_p, q_1, \ldots, q_m, q'_1, \ldots, q'_n \in Q_L$ and $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$. Hence $L(\mathcal{A}', q_p) \cap L(\mathcal{A}, p)$ is not empty because it contains t_p, and the transition rule denoted ρ in the above sequence has the form $b(B) \to q$, where $q_1 \ldots q_m q_p q'_1 \ldots q'_n$ is recognized by B, with a sequence $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{q_p} s' \xrightarrow[B]{q'_1 \ldots q'_n} f_B$ for some states s, s' of B. Therefore, a transition $a(B + \langle s, \varepsilon, s' \rangle) \to q$ has been added to $\mathcal{A}'$, and $q_1 \ldots q_m q'_1 \ldots q'_n$ is recognized by $B + \langle s, \varepsilon, s' \rangle$. It follows that $u = u[a(h\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q'_1 \ldots q'_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{left}}$. Assume that $u = u[b(h\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(h\,t_p\,a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$ for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u_1 = u[b(h\,t_p\,a(v)\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h\,t_p\,a(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \ldots q_m\,q_p\,q' q'_1 \ldots q'_n)] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q', q_p, q_1, \ldots, q_m, q'_1, \ldots, q'_n \in Q_L$, $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$, and $a(v) \xrightarrow[\mathcal{A}']{*} q'$. Hence $L(\mathcal{A}', q_p) \cap L(\mathcal{A}, p)$ is not empty because it contains t_p, and the transition rule denoted ρ in the above sequence has the form $b(B) \to q$ with $q_1 \ldots q_m q_p q' q'_1 \ldots q'_n$ is recognized by B, with a sequence, $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow{q_p q'} s' \xrightarrow[B]{q'_1 \ldots q'_n} f_B$ for some of states s and s' of B. Hence, a transition $b(B + \langle s, q', s' \rangle) \to q$ has been added to $\mathcal{A}'$, and $q_1 \ldots q_m q' q'_1 \ldots q'_n$ is recognized by $B + \langle s, q', s' \rangle$. It follows that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q' q'_1 \ldots q'_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{right}}$. This case is similar to the previous one.

RPL'. Assume that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(ht_1 \ldots t_n\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$ for some $t_1, \ldots, t_n$ respectively in $L(\mathcal{A}, p_1), \ldots, L(\mathcal{A}, p_n)$. By induction hypothesis, $u_1 = u[b(ht_1 \ldots t_n\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h\,t_1 \ldots t_n\,\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \ldots q_m\,q_{p_1} \ldots q_{p_n}\,q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_{p_1}, \ldots, q_{p_n}, q_1, \ldots, q_m, q'_1, \ldots, q'_{m'} \in Q_L$, $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$, and for all $j \le n$, $L(\mathcal{A}', q_{p_j}) \cap L(\mathcal{A}, p_j)$ contains t_j, and the transition rule denoted ρ in the above sequence has the form $b(B) \to q$ with $q_1 \ldots q_m\, q_{p_1} \ldots q_{p_n}\, q'_1 \ldots q'_{m'} \in L(B)$, with a sequence $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow{q_{p_1} \ldots q_{p_n}} s' \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$, for some states s and s' of B. Let $q' \in Q_L$ be such that $a(v) \xrightarrow[\mathcal{A}']{*} q'$. By construction, a transition $b(B + \langle s, q', s' \rangle) \to q$ has been added to $\mathcal{A}'$, and $q_1 \ldots q_m\, q'\, q'_1 \ldots q'_{m'}$ is recognized by $B + \langle s, q', s' \rangle$. It follows that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q' q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

DEL. Assume that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(h\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. By induction hypothesis, $u_1 = u[b(h\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \ldots q_m\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_1, \ldots, q_m, q'_1, \ldots, q'_{m'} \in Q_L$ and $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$. The transition rule denoted ρ in the above sequence has the form $b(B) \to q$ and $q_1 \ldots q_m\, q'_1 \ldots q'_{m'}$ is recognized by B with a sequence $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$, where s is a state of B. Let $q' \in Q_L$ be such that $a(v) \xrightarrow[\mathcal{A}']{*} q'$. By construction, a transition $b(B + \langle s, q', s \rangle) \to q$ has been added to $\mathcal{A}'$, and $q_1 \ldots q_m\, q'\, q'_1 \ldots q'_{m'}$ is recognized by $B + \langle s, q', s \rangle$. It follows that $u = u[b(h\, a(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q' q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

$\mathsf{DEL_s}$. Assume that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} u[b(hv\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. By induction hypothesis, $u_1 = u[b(hv\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(hv\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \ldots q_m\, q''_1 \ldots q''_n\, q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_1, \ldots, q_m, q''_1, \ldots, q''_n, q'_1, \ldots, q'_{m'} \in Q_L$ and $q^{\mathsf{f}} \in Q^{\mathsf{f}}_L$. The transition rule denoted ρ in the above sequence has the form $b(B) \to q$ and $q_1 \ldots q_m\, q''_1, \ldots, q''_n\, q'_1 \ldots q'_{m'}$ is recognized by B, with a sequence $i_B \xrightarrow[B]{q_1 \ldots q_m} s \xrightarrow[B]{q''_1 \ldots q''_n} s' \xrightarrow[B]{q'_1 \ldots q'_{m'}} f_B$, where s, s' are two states of B. By completeness of $\mathcal{A}_L$, given s, s', there exists q' such that $a(B_{s,s'}) \hookrightarrow_{\Delta_i} q'$. It follows in particular that $a(v) \xrightarrow[\mathcal{A}']{*} q'$. By construction, a transition $b(B + \langle s, q', s \rangle) \to q$ has been added to $\mathcal{A}'$, and $q_1 \ldots q_m\, q'\, q'_1 \ldots q'_{m'}$ is recognized by $B + \langle s, q', s \rangle$. It follows that $u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \ldots q_m q' q'_1 \ldots q'_{m'})] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$, hence that $u \in L(\mathcal{A}')$.

(end Lemma direction $\subseteq$) □

(end of the proof of Theorem 3) □

E Appendix: proof of Theorem 4

Theorem 4. *Reachability is undecidable for uniform PGTRS without variables and parameters.*

Proof. We will reduce the halting problem of Deterministic Turing Machines (TM) that work on half a tape (unbounded on the right). We consider the following unary symbols to represent the tape alphabet $\Sigma = \{0, 1, \sharp, \flat\}$. We need a copy of the alphabet $\Sigma' = \{0', 1', \sharp', \flat'\}$. We only use $\sharp$ to mark the left endpoint of the tape and $\flat$ is the blank symbol, e.g. representing the rightmost part of the tape.

The state symbols are constants in a finite set $Q \cup Q'$ where $Q = \{q_1, q_2, \ldots, q_n\}$ and $Q' = \{q'_1, q'_2, \ldots, q'_n\}$. Hence each state of the TM has two representations.

In order to represent a Turing machine configuration as a ground term we shall introduce a binary symbol $+$ and a nullary symbol $\perp$. Now the TM configuration with tape $abccde\flat\flat\ldots$, symbol under head d, state q will be represented by:

$$\sharp(\perp) + (a(\perp) + (b(\perp) + (c(\perp) + (c(\perp) + (d(q) + (c(\perp) + \flat(\perp))))))).$$

We denote by $\mathcal{T}_0$ (resp. $\mathcal{T}_1$) the set of terms on signature $\Sigma \cup \{\perp, +\}$ with no occurrence of $\sharp$ (resp. with a unique occurrence of $\sharp$ at position 1). Given a term in $t \in \mathcal{T}_0$ and a term $s \in \mathcal{T}(\Sigma)$ we write $t[\perp \leftarrow s]$ the term obtained from t by replacing its rightmost $\perp$ symbol by s.

For each TM transition we introduce some rewrite rules that simulate it on the term representation. We introduce now some tree regular languages: $L_{s,a}$ is the subset of $t \in \mathcal{T}(\Sigma)$ such that t admits a single occurrence of a state symbol and this state symbol is s, and it occurs right below a symbol a.
"In state q reading a go to state r and write b". This is translated to the ground rewrite rule:

$$L_{q,a} :: a(q) \to b(r)$$

"In state q reading a go to state r and move right". This can be simulated by some application of rules:

$$L_{q,a} :: u(\perp) \quad \to \quad u(r') \;\text{ for all }\; u \in \{0, 1, \sharp\} \tag{4}$$
$$L_{q,a} :: \flat(\perp) \quad \to \quad \flat(r') + \flat(\perp) \tag{5}$$

Note that one of these rule application may create a pattern $a(q) + (b(r') + x)$ at the location where we had a pattern $a(q) + (b(\perp) + x)$ in the configuration. Let $L_{q,a,r,R}$ be the set of term of type $U[\perp \leftarrow (a(q) + (b(r') + V)]$ where $U \in \mathcal{T}_1$, $V \in \mathcal{T}_0$. This is clearly a regular language. Then we add the rules:

$$L_{q,a,r,R} :: a(q) \quad \to \quad a(\perp) \tag{6}$$
$$L_{r',u} :: u(r') \quad \to \quad u(r) \;\text{ for all }\; u \in \{0, 1, \sharp\} \tag{7}$$

"In state q reading a go to state r and move left". This can be simulated by some application of rules:

$$L_{q,a} :: u(\perp) \quad \to \quad u(r') \;\text{ for all }\; u \in \{0, 1, \sharp\} \tag{8}$$

This rule application may create a pattern $b(r') + (a(q) + x)$ at the location where we had a pattern $b(\perp) + (a(q) + x)$ in the configuration. Let $L_{q,a,r,L}$ be the set of term of type $U[\perp \leftarrow ((b(r') + a(q)) + V)]$ where $U \in \mathcal{T}_1$, $V \in \mathcal{T}_0$. This is clearly a regular language. Then we add the rules:

$$L_{q,a,r,L} :: a(q) \quad \to \quad a(\perp) \tag{9}$$
$$L_{r',u} :: u(r') \quad \to \quad u(r) \;\text{ for all }\; u \in \{0, 1, \sharp\} \tag{10}$$

Let us denote $\mathcal{R} = \{L_i :: \ell_i \to r_i \mid 1 \leq i \leq n\}$ the set of rules we obtain by the above construction. Note that the languages L_i are pairwise disjoint. By case inspection we can show that for any couple of TM configurations T_1, T_2 and their respective term encodings t_1, t_2, there is a sequence of transitions from T_1 to T_2 iff $t_1 \xrightarrow[\mathcal{R}]{*} t_2$. If we replace in every rule the regular language L_i by the disjoint union $\biguplus_{1\leq i\leq n} L_i$, the result still holds. The theorem follows. □

F Appendix: proof of Theorem 5

Theorem 5. *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in \mathsf{XACU}_2{+}$, for all HA language L, $pre^*_{\mathcal{R}/\mathcal{A}}(L)$ is a HA the language.*

Proof. The proof is very close to the one of Theorem 3. Indeed, in the above construction for Theorem 3, we consider the applications of rules $\mathsf{INS}_{\mathsf{left}}$, $\mathsf{INS}_{\mathsf{right}}$, RPL', DEL and $\mathsf{DEL}_{\mathsf{s}}$ under any symbol $b \in \Sigma$. Here instead, we can restrict the construction to the application under the symbol specified in the lhs of the rewrite rules. More precisely, let us just detail below the cases of the construction which are modified. The rest of the prof is the same as for Theorem 3.

$\mathsf{INS}_{2,\mathsf{left}}$: if $b(y\,a(x)\,z) \to b(y\,p\,a(x)\,z) \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q_p, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q_p q'} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$.

$\mathsf{INS}_{2,\mathsf{right}}$: if $b(y\,a(x)\,z) \to b(y\,a(x)\,p\,z) \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q_p, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[B]{q' q_p} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$.

RPL'_2: if $b(y\,a(x)\,z) \to b(y\,p_1 \ldots p_n\,z) \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s, s' are states of B, and $q, q', q_1, \ldots, q_n \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_j) \cap L(\mathcal{A}, p_j) \neq \emptyset$ for all $1 \leq j \leq n$, $s \xrightarrow[B]{q_1 \ldots q_n} s'$ then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle)) \to q\big\}$.

DEL_2: if $b(y\,a(x)\,z) \to b(yz) \in \mathcal{R}/\mathcal{A}$, $B, B' \in \mathcal{C}$, s is a state of B, $q, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B') \hookrightarrow_{\Delta_i} q'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s\rangle) \to q\big\}$.

$\mathsf{DEL}_{2,\mathsf{s}}$: if $b(y\,a(x)\,z) \to b(yxz) \in \mathcal{R}/\mathcal{A}$, $B \in \mathcal{C}$, s, s' are states of B, $q, q' \in Q_L$ such that $b(B) \to q \in \Delta_i$, $a(B_{s,s'}) \hookrightarrow_{\Delta_i} q'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(B + \langle s, q', s'\rangle) \to q\big\}$. □

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399