Formal Verification of XML Updates and their Access Control Policies[☆]

Florent Jacquemard[a], Michael Rusinowitch[b]

[a] *INRIA Saclay - IdF*
LSV, ENS Cachan 61 av. du pdt Wilson 94230 Cachan, France
[b] *INRIA Nancy - Grand Est & LORIA UMR*
615 rue du Jardin Botanique 54602 Villers-les-Nancy, France

Abstract

We propose a model for XML update primitives of the W3C XQuery Update Facility as parametrized rewriting rules of the form: "insert an unranked tree from a regular tree language L as the first child of a node labeled by a". For these rules, we give type inference algorithms, considering types defined by several classes of unranked tree automata. These type inference algorithms are directly applicable to XML static typechecking, which is the problem of verifying whether, a given document transformation always converts source documents of a given input type into documents of a given output type. We show that typechecking for arbitrary sequences of XML update primitives can be done in polynomial time when the unranked tree automaton defining the output type is deterministic and complete, and that it is EXPTIME-complete otherwise.
We then apply the results to the verification of access control policies for XML updates. We propose in particular a polynomial time algorithm for the problem of local consistency of a policy, that is, for deciding the non-existence of a sequence of authorized update operations starting from a given document that simulates a forbidden update operation.

Keywords: Program Verification, Static Typechecking, Web Security, XML Updates, XML Access Control Policies, Hedge Automata.

1. Introduction

XQuery language has been extended to XQuery Update Facility [8] in order to provide convenient means of modifying XML documents or data. The language is a candidate recommendation from W3C and adds imperative operations that permit one e.g. to update some parts of a document while leaving the rest unchanged. This includes rename, insert, replace and delete primitive operations at the node level. Compared to other transformation languages (such a XSLT), XQuery Update Facility is considered to offer concise, readable solutions.

[☆]This research was partly supported by the ARC INRIA 2010 ACCESS
Email addresses: florent.jacquemard@inria.fr (Florent Jacquemard), rusi@loria.fr (Michael Rusinowitch)

A central problem in XML document processing is *static typechecking*. This problem amounts to verifying at compile time that every output XML document which is the result of a specified query or transformation applied to an input document with a valid input type has a valid output type. However for transformation languages such as the one provided by XQuery Update Facility, the output type of (iterated) applications of update primitives are not easy to predict. Another important issue for XML data processing is the specification and enforcement of access policies. A large amount of work has been devoted to secure XML querying. But most of the work focus on read-only rights, and very few have considered update rights for a model based on XQuery Update Facility operations [e.g. 6, 17].

In the domain of formal verification of infinite state systems and programs, several approaches such as regular model checking rely on algorithms computing the rewrite closure of tree automata languages, see e.g. [5, 16]. It seems natural to consider such tree automata techniques for verification problems related to the typing of XML documents and XML transformations, in particular XML updates [8]. Indeed, XML documents are commonly represented as finite labeled unranked trees, and most of the typing formalisms currently used for XML are based on finite tree automata [30, 36].

A standard approach to XML typechecking is forward (resp. backward) *type inference*, that is, the computation of an output (resp. input) XML type (as a tree automaton) given an input (resp. output) type and a tree transformation. Then the typechecking itself can be reduced to the verification of set operations on the computed input or output type, see [28] for an example of backward type inference procedure.

In this paper, we consider the problem of typechecking arbitrary sequences of operations taken in a given set of atomic update primitives. We propose a modeling of (possibly infinite) sets of primitive update operations of the W3C XQuery Update Facility proposal [8] in terms of rewrite rules with parameters. The update operations include renaming, insertion, deletion and replacement in XML documents, and some extensions, like the deletion of one single node (preserving its descendant) instead of the deletion of a whole subtree. For several subclasses of these operations, we derive algorithms of synthesis of unranked tree automata, yielding both forward and backward type inference results. Since update operations, beside relabeling document nodes, can create and delete entire XML fragments, modifying a document's structure, it is not obvious how to infer the type of updated documents. Former tree automata completion constructions like [16] work for automata computing on ranked trees. Here, we consider unranked ordered trees, and our constructions are non trivial adaptations of former tree automata completion procedures, where, starting from an initial automaton, new transitions rules are added and existing transition rules are recursively modified. Moreover, we show that some update operations do not preserve regular tree languages (i.e. languages of hedge automata) and that for the type inference for these operations, we need to consider a larger and less mainstream class of decidable unranked tree recognizers called context-free hedge automata.

One of our motivations for this study is the static analysis of access control policies (ACP) for XML updates. We consider two approaches for this problem. The first approach addresses *rule-based* specifications of ACPs, where the

operations allowed, resp. forbidden, to a user are specified as two sets of atomic update primitives [6, 17]. We show in particular how to apply our type inference results to the verification of local consistency of ACPs, i.e. 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 which are challenging to detect according to [17]. In the second approach (*DTD-based* XML ACPs) the ACP is defined by adding security annotations to a DTD D, as in [15, 17]. In this case, it is required to check the validity of the document wrt D before applying every update operation. We show that under this restriction typechecking becomes undecidable.

Related work: Many works have employed tree automata to compute sets of descendants for standard (ranked) term rewriting (see e.g. [16]). Regular model checking [4] is extended to hedge rewriting and hedge automata in [39], which gives a procedure to compute reachability sets *approximations*. Here we compute exact reachability sets for some classes of hedge rewrite systems. For some results we need context-free hedge automata, a more general class than the regular hedge automata of [39].

When considering real programming languages like XDuce or CDuce [3] 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) [25] and macro tree transducers (MTT) [26, 34], and k-pebble tree transducers (k-PTT) [28], a powerful model defined so as to cover relevant fragments of XSLT [22] 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 [13, 27] in order to obtain PTIME typechecking procedures.

In this paper, we consider unrestricted applications of updates, unlike e.g. top-down transductions in [27]. It is shown in [28] that the set of output trees of a k-PTT for a fixed input tree is a regular tree language. In contrast, we shall see (Examples 8,9 below) that it is not the case for the iteration of some update operations, and therefore that such transformation are not expressible as k-PTT. In Theorem 2, we show that the output language of the iteration of these updates for a regular input language is recognizable by a context-free hedge automata. This can be related to the result of [14], used in [26] in the context of typechecking XML transformations, and stating that the output language of a linear stay MTT can be characterized by a context-free tree grammar (in the case of ranked trees). Theorem 2 implies that the output languages of the iteration of updates can be described by MTTs, as MTT can generate all context-free tree languages. On the other hand, each of the primitive update operations can be solely modeled by a MTT. It is however not clear whether the finite (but unbounded) iterations of updates operations can be easily expressed as a MTT relation.

In [2] the authors investigate the problem of synthesizing an output schema describing the result of an update applied to a given input schema. They show how to infer safe over-approximations for the results of both queries and updates (see the beginning of Section 3 and Section 4.4).

Recent works have also applied local Hoare reasoning to simple tree update and even to a significant subset of the XML update library in W3C Document

Object Model [18]. As far as we know this approach is not automated.

The first access control model for XML was proposed by [10] and was extended to secure updates in [7]. Static analysis has been applied to XML Access Control in [32] to determine if a query expression is guaranteed not to access to elements that are forbidden by the policy. In [17] the authors propose the XACU language. They study policy consistency and show that it is undecidable in their setting. On the positive side [6] considers policies defined in term of annotated non recursive XML DTDs and gives a polynomial algorithm for checking consistency.

Organization of the paper: we introduce the needed formal background about terms, hedge automata and rewriting systems in Section 2. Then we propose a model of XML update primitives as parametrized rewriting rules in Section 3. In Section 4 we present type synthesis algorithms for the iteration of such rules. Finally we give applications to the verification of Access Control Policies for updates in Section 5.

2. Definitions

2.1. Unranked Ordered Trees

2.1.1. 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 x, $y \ldots$ We define recursively a *hedge* over Σ and $\mathcal{X}$ as a finite (possibly empty) sequence of terms and a *term* as either a single node n labeled by a variable of $x \in \mathcal{X}$ or the application of a node n labeled by a symbol $a \in \Sigma$ to a hedge h. The term is denoted x in the first case and $a(h)$ in the second case, and n is called the *root* of the term in both cases. The empty sequence is denoted () and when h is empty, the term $a(h)$ will be simply denoted by a. The root node of $a(h)$ is called the *parent* of every root of h and every root of h is called a *child* of the root of $a(h)$. A root of a hedge $(t_1 \ldots t_n)$ is a root node of one of $t_1, ..., t_n$. A leaf of a hedge $(t_1 \ldots t_n)$ is a leaf (node without child) of one of the terms $t_1, ..., t_n$. A *path* is a sequence of nodes $n_0, \ldots, n_p$ such that for all $i < p$, n_{i+1} is a child of n_i. In this case, n_p is called a *descendant* of n_0. As usual, we can see a hedge $h \in \mathcal{H}(\Sigma, \mathcal{X})$ as a function from its set of nodes $dom(h)$ into labels in $\Sigma \cup \mathcal{X}$. The label of the node $n \in dom(h)$ is denoted by $h(n)$.

The set of hedges and terms over Σ and $\mathcal{X}$ are respectively denoted $\mathcal{H}(\Sigma, \mathcal{X})$ and $\mathcal{T}(\Sigma, \mathcal{X})$. We will sometimes consider a term as a hedge of length one, *i.e.* consider that $\mathcal{T}(\Sigma, \mathcal{X}) \subset \mathcal{H}(\Sigma, \mathcal{X})$. The sets of ground terms (terms without variables) and ground hedges are respectively denoted $\mathcal{T}(\Sigma)$ and $\mathcal{H}(\Sigma)$. 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.

2.1.2. 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 $x \in dom(\sigma)$, $y\sigma := y$ when $y \in \mathcal{X} \setminus dom(\sigma)$, $(t_1 \ldots t_n)\sigma := (t_1\sigma \ldots t_n\sigma)$ for $n \geq 0$, and $a(h)\sigma := a(h\sigma)$.

2.1.3. 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 a hedge in u in place of the node labelled by x_u. Sometimes, we write $t[s]$ in order to emphasize that s is a subterm (or subhedge) of t.

2.2. Finite Automata and Grammars

In the following proofs, we describe *finite automata* for the horizontal languages of HA transitions as tuples $B = (Q, S, i, F, \Gamma)$, where Q is a 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 $\Gamma \subseteq S \times (Q \cup \{\varepsilon\}) \times S$ is the set of transitions and ε-transitions. Every transition (s, q, s') will be denoted $s \xrightarrow{q} s'$. For $s, s' \in S$, we write $s \xrightarrow[B]{\varepsilon} s'$ to express that s' can be reached from s by a (possibly empty) sequence of ε-transitions of B, and $s \xrightarrow{q_1 \dots q_n} s'$, for $q_1, \dots, q_n \in Q$, if there exists $2(n+1)$ states $s_0, s'_0, \dots, s_n, s'_n \in S$ with $s_0 = s$, $s_n \xrightarrow[B]{\varepsilon} s'$ and $0 \leq i < n$, $s_i \xrightarrow[B]{\varepsilon} s'_i$ and $(s'_i, q_{i+1}, s_{i+1}) \in \Gamma$.

We describe below *CF grammars* as tuples $\mathcal{G} = (Q, \mathcal{N}, I, \Gamma)$, where Q is a finite input alphabet, (set of terminal symbols), $\mathcal{N}$ is a set of non terminal symbols, $I \in \mathcal{N}$ is the initial non terminal, and $\Gamma \subseteq \mathcal{N} \times (\mathcal{N} \cup Q)^*$ is a set of production rules of the form $N := w$ (with $N \in \mathcal{N}$ and $w \in (\mathcal{N} \cup Q)^*$).

2.3. Hedge Automata and Context-Free Hedge Automata

We consider two kind of types for XML documents, defined as two classes of automata for unranked trees. The first one is the class of hedge automata [30], denoted HA. It captures the expressive strength of almost all popular type formalisms for XML [31]. The second and perhaps lesser known class is the context-free hedge automata, denoted CF-HA and introduced in [33]. CF-HA are strictly more expressive than HA and we shall see that they are of interest for typing certain update operations.

A *hedge automaton* (resp. *context-free hedge automaton*) is a tuple $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ where Σ is a 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 $h, h' \in \mathcal{H}(\Sigma \cup Q)$ 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 $\xrightarrow[\mathcal{A}]{}$.

The language of a HA or CF-HA $\mathcal{A}$ in one of its states q, denoted by $L(\mathcal{A}, q)$ and also called the set of hedges of type q, is the set of ground hedges $h \in \mathcal{H}(\Sigma)$ such that $h \xrightarrow[\mathcal{A}]{*} q$. We say sometimes that a 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 hedges accepted by $\mathcal{A}$.

2.3.1. Collapsing Transitions, δ- and ε-Transitions

We consider the extension of HA and CF-HA with so called *collapsing transitions* which are special transitions of the form $L \to q$ where $L \subseteq Q^*$ is a context-free 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 there exists a collapsing transition $L \to q$ of $\mathcal{A}$ and $q_1 \dots q_n \in L$. The definition of the languages of HA and CF-HA is extended to automata with collapsing transitions accordingly.

Note that we do not exclude the case $n = 0$ in the above definition, i.e. L may contain the empty word in $L \to q$. A collapsing transition where L contains only the empty word ($L = \{()\}$) is called a *δ-transition*. A 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, and we use the same name here. We shall use below the simpler notations $() \xrightarrow{\delta} q$ and $q' \xrightarrow{\varepsilon} q$ for respectively δ- and ε-transitions.

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 do not increase the expressiveness HA or CF-HA (see [9] for HA and the proof for CF-HA is similar). One can also observe that it is also the case of δ-transitions.

Lemma 1. *Every HA (resp. CF-HA) $\mathcal{A}$ extended with ε- and δ-transitions can be transformed in polynomial time into a HA (resp. CF-HA) $\mathcal{A}'$ without collapsing transitions such that $L(\mathcal{A}') = L(\mathcal{A})$.*

Proof. Let $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$, where Δ contains both standard HA transitions of the form $a(L) \to q$ and ε- and δ-transitions. Let $Q_0 = \{q \in Q \mid () \xrightarrow{\delta} q \in \Delta\}$ and for all $q \in Q$, let $q^{-1} = \{q' \in Q \mid q' \xrightarrow[\Delta\varepsilon]{*} q \in \Delta\}$, where $\xrightarrow[\Delta\varepsilon]{*}$ denotes the reflexive and transitive closure of the relation $\{(q', q) \in Q^2 \mid q' \xrightarrow{\varepsilon} q \in \Delta\}$. Given $L \subseteq Q^*$, we construct L' by substitution of every state symbol q by the regular language $Q_0^* \, q^{-1} Q_0^*$. Note that if L is regular (resp. CF) then L' is regular (resp. CF).
Then we let $\Delta' = \{a(L') \to q \mid a(L) \to q \in \Delta\}$, and $\mathcal{A}' = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$. □

However, general collapsing transitions strictly extend HA in expressiveness, and even collapsing transitions of the form $L \to q$ where L is finite (hence regular).

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

We recall the following lemma, proved in [21], that shows that collapsing transitions can be eliminated from CF-HA, when restricting to the recognition of terms.

Lemma 2. *Every CF-HA $\mathcal{A}$ extended with collapsing transitions can be transformed in polynomial time into a CF-HA $\mathcal{A}'$ without collapsing transitions such that $L(\mathcal{A}') = L(\mathcal{A}) \cap \mathcal{T}(\Sigma)$.*

2.3.2. Properties.

It is known that for both classes of HA and CF-HA, the membership and emptiness problems are decidable in PTIME [9, 30, 33].

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 exactly one transition rule $a(L_{a,q}) \to q$ in Δ (note that the language $L_{a,q}$ might be empty).

Lemma 3. *Every HA (resp. CF-HA) $\mathcal{A}$ can be transformed in polynomial time into a normalized HA (resp. CF-HA) $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A})$.*

Proof. The transformation works by replacing iteratively every two rules $a(L_1) \to q$ and $a(L_2) \to q$ by $a(L_1 \cup L_2) \to q$. □

A CF-HA $\mathcal{A} = (Q, Q^{\mathsf{f}}, \Delta)$ is called *deterministic* iff for all two transitions rules $a(L_1) \to q_1$ and $a(L_2) \to q_2$ in Δ, either $L_1 \cap L_2 = \emptyset$ or $q_1 = q_2$. It is called *complete* if for all $a \in \Sigma$ and and $w \in Q^*$, there exists at least one rule $a(L) \to q \in \Delta$ such that $w \in L$. When $\mathcal{A}$ is deterministic (resp. complete), for all $t \in \mathcal{T}(\Sigma)$, there exists at most (resp. at least) one state $q \in Q$ such that $t \in L(\mathcal{A}, q)$. Every HA can transformed into a deterministic and complete HA recognizing the same language (see *e.g.* [9]). CF-HA can be completed but not determinized.

Finally, 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 (resp. EXPTIME) constructions of automata of polynomial (resp. exponential) sizes for the closures under union and intersection (resp. complement).

2.4. Term Rewriting Systems

We use below term rewriting rules for modeling XML update operations. For this purpose, we propose a non-standard definition of term rewriting, extending the classical one [12] 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).

2.4.1. Unranked 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 not assume the cardinality of $\mathcal{R}$ to be finite.

The rewrite relation $\xrightarrow[\mathcal{R}]{}$ of a 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 $\mathcal{R} = \{g(x) \to g(axb)\}$, $g(c) \xrightarrow[\mathcal{R}]{*} g(a^n c b^n)$ for every $n \geq 0$. ◇

2.4.2. Parameterized Term Rewriting Systems.

Let $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ be a HA. A term rewriting system over Σ parametrized by $\mathcal{A}$ (**PTRS**) 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 ($\uplus$ stands disjoint union, hence we implicitly assume that Σ and Q are disjoint sets). 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 $p \in Q$ by a ground term of $L(\mathcal{A}, p)$. Note that when there are multiple occurrences of a state p in a rule, each occurrence of p is independently replaced with a term of type p, which can generally be different from one another.

Several examples of parametrized rewrite rules can be found in Figure 1 below. We will also consider in Section 5.2 an extension of PTRS called PTRS with global constraints (PGTRS) .

2.4.3. Problems.

Given a set $L \subseteq \mathcal{H}(\Sigma, \mathcal{X})$ and a PTRS $\mathcal{R}/\mathcal{A}$, we define $post^*_{\mathcal{R}/\mathcal{A}}(L) := \{h' \in \mathcal{H}(\Sigma, \mathcal{X}) \mid \exists h \in L, h \xrightarrow[\mathcal{R}/\mathcal{A}]{}{}^* h'\}$ and $pre^*_{\mathcal{R}/\mathcal{A}}(L) := \{h \in \mathcal{H}(\Sigma, \mathcal{X}) \mid \exists h' \in L, h \xrightarrow[\mathcal{R}/\mathcal{A}]{}{}^* h'\}$.

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 term rewriting have been investigated in e.g. [19]. C. Löding [23] 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 tree language R. Then [24] 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 (see e.g. [28]) 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 $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in}) \subseteq \tau_{out}$ or equivalently $\tau_{in} \cap pre^*_{\mathcal{R}/\mathcal{A}}(\overline{\tau_{out}}) = \emptyset$ (where $\overline{\tau_{out}}$ is the complement of τ_{out}). Note that reachability is a special case of typechecking, when both τ_{in} and τ_{out} are singleton sets. Hence typechecking is undecidable whenever reachability is. One related problem, called *forward* (resp. *backward*) *type inference*, is, given a PTRS $\mathcal{R}/A$ and a HA or CF-HA language L, to construct a HA or CF-HA recognizing $post^*_{\mathcal{R}/\mathcal{A}}(L)$ (resp. $pre^*_{\mathcal{R}/\mathcal{A}}(L)$).

3. Primitive Update Facility Operations

We propose in this section a definition in term of PTRS rules of the update primitive operations of the XQuery update facility [8] and some extensions.

XQuery Update Facility [8] is a extension of XQuery with some update primitives, to be applied to a term in input, in order to rename nodes, replace a subterm by a new one, insert a new subterm at some position or delete a subterm. In the case of replacement or insertion, the new subterms in argument (called *content* nodes in [8]) are specified by positions within the term in input (using XPath expressions).

$a(x)$	$\to$	$b(x)$	REN				
$a(x)$	$\to$	$a(p\,x)$	$\mathsf{INS}_{\mathsf{first}}$	$a(x)$	$\to$	$p\,a(x)$	$\mathsf{INS}_{\mathsf{before}}$
$a(x)$	$\to$	$a(x\,p)$	$\mathsf{INS}_{\mathsf{last}}$	$a(x)$	$\to$	$a(x)\,p$	$\mathsf{INS}_{\mathsf{after}}$
$a(xy)$	$\to$	$a(x\,p\,y)$	$\mathsf{INS}_{\mathsf{into}}$				
$a(x)$	$\to$	p	RPL_1	$a(x)$	$\to$	$p_1 \ldots p_n$	RPL
$a(x)$	$\to$	$()$	DEL	$a(x)$	$\to$	x	$\mathsf{DEL}_{\mathsf{s}}$

Figure 1: Class UFO: PTRS Rules for the XQuery Update Facility Primitives and Extensions

Benedikt and Cheney have recently proposed [2] a formal model for XQuery Update Facility, with languages for update primitives and XQuery Updates and their operational semantics. In their abstract model, the subterms in arguments are approximated by states of a tree automaton (*type names* of *regular expression types* [20]).

We follow a similar approach here, representing update primitives by PTRS rules. We assume given an unranked alphabet Σ and a HA $\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$. Figure 1 displays PTRS rules, parametrized by states p, p_1,.., p_n of $\mathcal{A}$, representing infinite sets of atomic operations of the XQuery update facility [8], and some restrictions or extensions. We call UFO the class of PTRS rules in Figure 1, and $\mathsf{UFO}_{\mathsf{reg}}$ the subclass of PTRS rules of type REN, $\mathsf{INS}_{\mathsf{first}}$, $\mathsf{INS}_{\mathsf{last}}$, $\mathsf{INS}_{\mathsf{into}}$, RPL_1, or DEL.

REN renames a node: it changes its 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. $\mathsf{INS}_{\mathsf{before}}$ (resp. $\mathsf{INS}_{\mathsf{after}}$) inserts 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 a subterm by a sequence of terms of respective types $p_1, \ldots, p_n$. We call RPL_1 the particular case of RPL with $n = 1$. Note that DEL is also a special case of RPL, with $n = 0$.

The insertion rules of UFO (the rules called INS_*) do not change the label of the node at the top of the lhs of the rules. Only the rule REN permits to change the label of a node in a term, while preserving the other nodes.

Example 3. The data of patients in a hospital is stored in an XML document whose DTD can be characterized by an HA $\mathcal{A}$ with the following transition rules

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{hospital}(\{p_{\mathsf{pa}}, p_{\mathsf{epa}}\}^*) & \to & p_{\mathsf{h}}, & \mathsf{patient}(p_{\mathsf{n}}) & \to & p_{\mathsf{epa}}, \\
\mathsf{patient}(p_{\mathsf{n}}\,p_{\mathsf{t}}) & \to & p_{\mathsf{pa}}, & \mathsf{name}(p_c^*) & \to & p_{\mathsf{n}}, \\
\mathsf{treatment}(p_{\mathsf{dr}}\,p_{\mathsf{dia}}\,p_{\mathsf{da}}) & \to & p_{\mathsf{t}}, & \mathsf{drug}(p_c^*) & \to & p_{\mathsf{dr}}, \\
\mathsf{diagnosis}(p_c^*) & \to & p_{\mathsf{dia}}, & \mathsf{date}(p_c^*) & \to & p_{\mathsf{da}} \\
\multicolumn{6}{r}{\mathsf{a} \to p_c, \mathsf{b} \to p_c, \mathsf{c} \to p_c \ldots}
\end{array}$$

The state p_{h} is the *entry point* of the DTD *i.e.* it represents the type of the root element.

A DEL rule $\mathsf{patient}(x) \to ()$ will delete a patient in the base and a $\mathsf{INS}_{\mathsf{last}}$ rule $\mathsf{hospital}(x) \to \mathsf{hospital}(x\,p_{\mathsf{pa}})$ will insert a new patient, at the last position below the root node hospital. We can ensure that the patient newly added has an empty treatment list (to be completed later) using $\mathsf{hospital}(x) \to \mathsf{hospital}(x\,p_{\mathsf{epa}})$. A $\mathsf{INS}_{\mathsf{after}}$ rule $\mathsf{name}(x) \to \mathsf{name}(x)\,p_{\mathsf{t}}$ can be used to insert later a treatment next to the patient's name. ◇

input language	rules	$post^*$	pre^*
HA	$\mathsf{UFO}_{\mathsf{reg}}$	HA, PTIME Th.1	
HA	UFO	CF-HA, PTIME Th.2	HA, EXPTIME Th. 3
CF-HA	UFO	CF-HA, PTIME Th.2	

Figure 2: Summary of results

We propose also in Figure 1 another primitive not in [8]: $\mathsf{DEL}_{\mathsf{s}}$ deletes a single node n whose arguments inherit the position. In other words, it replaces a term with the hedge containing its children. This operation is employed to build user views of XML documents e.g. in [15], and can also be useful for updates as well.

Example 4. Assume that some patients of the hospital of Example 3 are grouped in one department like in $\mathsf{hospital}(\ldots \mathsf{surgery}(p^*_{\mathsf{pa}})\ldots)$, and that we want to suppress the department $\mathsf{surgery}$ while keeping its patients. This can be done with the $\mathsf{DEL}_{\mathsf{s}}$ rule $\mathsf{surgery}(x) \to x$. ◇

We will see in Section 4.2 that allowing the primitive $\mathsf{INS}_{\mathsf{before}}$, $\mathsf{DEL}_{\mathsf{s}}$ or RPL has important consequences w.r.t. type inference. Indeed, the subclass $\mathsf{UFO}_{\mathsf{reg}}$ of primitives preserves languages of HA whereas the operations in the second column of the table in Figure 1 (types RPL, $\mathsf{INS}_{\mathsf{before}}$, $\mathsf{INS}_{\mathsf{after}}$, $\mathsf{DEL}_{\mathsf{s}}$) may transform a HA language into a CF-HA language.

4. Type Inference for the Iteration of Update Primitives

In this section, we study the problem of forward and backward type inference for arbitrary finite iterations of primitive update operations of the above kind, taken in a given set. More precisely, we present constructions of HAs recognizing: the forward closure of a HA language under rules of type in $\mathsf{UFO}_{\mathsf{reg}}$ (Section 4.1) and the backward closure of a HA language under arbitrary rules of UFO (Section 4.3), and we present the construction of a CF-HA recognizing the forward closure of a CF-HA language under arbitrary rules of UFO (Section 4.2). The results are summarized in Figure 2.

4.1. Regular Forward Type Inference for the Iteration of Some Update Primitives

We want to characterize the sets of terms which can be obtained, from terms of a given type, by arbitrary application of updates operations defined 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. We start with a subset of this family of rewrite rules which preserve HA languages, under iterated application.

Theorem 1. *Let $\mathcal{A}$ be a HA on Σ, $\mathcal{R}/\mathcal{A}$ be a PTRS containing UFO rules of type $\mathsf{UFO}_{\mathsf{reg}}$ and let L be a HA language. Then $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of an HA of size polynomial and which can be constructed in PTIME in the size of $\mathcal{R}/\mathcal{A}$ and of an HA recognizing L.*

PROOF. We will construct a HA $\mathcal{A}'$ with δ- and ε-transitions recognizing $post^*_{\mathcal{R}/\mathcal{A}}(L)$. First, in order to simplify the construction, we shall assume the same HA both for defining the parameters of the rewrite system and for defining the language L. Let $\mathcal{A} = (\Sigma, P, P^{\mathsf{f}}, \Theta)$, and let $\mathcal{A}_L = (\Sigma, Q_L, Q_L^{\mathsf{f}}, \Delta_L)$ be a HA recognizing L. We assume wlog that both $\mathcal{A}$ and $\mathcal{A}_L$ are normalized and complete and that their respective state sets P and Q_L are disjoint.
Let $\mathcal{A}_0 = (\Sigma, Q_0, Q_0^{\mathsf{f}}, \Delta_0)$ be the normalized disjoint union of $\mathcal{A}$ and $\mathcal{A}_L$, defined by $Q_0 = P \uplus Q_L$, $Q_0^{\mathsf{f}} = Q_L^{\mathsf{f}}$, and Δ_0 is obtained by the disjoint union of Θ and Δ_L, and by the application of a normalization procedure, as in Lemma 3. Clearly, it holds that $post^*_{\mathcal{R}/\mathcal{A}_0}(L) = post^*_{\mathcal{R}/\mathcal{A}}(L)$. Below, the states of Q_0 will be indifferently denoted by the letters q or p.
For each $a \in \Sigma$ and $q \in Q_0$, let $L_{a,q}$ be the regular language in the transition (assumed unique) $a(L_{a,q}) \to q \in \Delta_0$, and let $\mathcal{B}_{a,q} = \big(Q_0, S_{a,q}, \mathsf{i}_{a,q}, \{\mathsf{f}_{a,q}\}, \Gamma_{a,q}\big)$, be a finite automaton recognizing $L_{a,q}$, with an input alphabet Q_0, a set of states $S_{a,q}$, an initial state $\mathsf{i}_{a,q} \in S_{a,q}$, a unique final state $\mathsf{f}_{a,q} \in S_{a,q}$ distinct from $\mathsf{i}_{a,q}$, and a set of transition rules $\Gamma_{a,q} \subseteq S_{a,q} \times Q_0 \times S_{a,q}$. The above hypotheses on $\mathcal{B}_{a,q}$ are not restrictive ($\mathcal{B}_{a,q}$ may contain ε-transitions). In particular,

if $L_{a,q} = \emptyset$, then we assume that $S_{a,q} = \{\mathsf{i}_{a,q}, \mathsf{f}_{a,q}\}$ and $\Gamma_{a,q} = \emptyset$, and

if $L_{a,q} = \{\varepsilon\}$ then we assume that $S_{a,q} = \{\mathsf{i}_{a,q}, \mathsf{f}_{a,q}\}$ and $\Gamma_{a,q} = \{\mathsf{i}_{a,q} \xrightarrow{\varepsilon} \mathsf{f}_{a,q}\}$.

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 $q \in Q_0$, and let Γ_0 be the disjoint union of all $\Gamma_{a,q}$ for $a \in \Sigma$, $q \in Q_0$ (Γ_0 is a set of transition rules over $S \times (P \cup Q_L) \times S$). We shall complete incrementally Γ_0 into Γ_1, Γ_2,... by adding some transitions rules, according to a case analysis of the update rules of $\mathcal{R}/\mathcal{A}_0$. At each step i, for each $a \in \Sigma$ and $q \in Q_0$, we let $\mathcal{B}_{i,a,q}$ be the automaton $(Q_0, S, \mathsf{i}_{a,q}, \{f_{a,q}\}, \Gamma_i)$.

Moreover, we construct incrementally, in the case analysis, a set of δ- and ε-rules. We start with $\mathcal{C}_0 = \emptyset$ and complete it into $\mathcal{C}_1$, $\mathcal{C}_2$,... by adding some δ- and ε-rules. Finally, we let, for each step $i \geq 0$

$$\Delta_i := \big\{a\big(\mathcal{B}_{i,a,q}\big) \to q \;\big|\; a \in \Sigma, q \in Q_0\big\} \cup \mathcal{C}_i \text{ and } \mathcal{A}_i = (\Sigma, Q_0, Q_0^{\mathsf{f}}, \Delta_i).$$

The constructions of the sequences (Γ_i) and $(\mathcal{C}_i)$ (and hence (Δ_i)) work in parallel, by iteration of the following case analysis: assuming that $(\Gamma_i, \mathcal{C}_i)$ is the last pair built, we define Γ_{i+1} and $\mathcal{C}_{i+1}$ by application of the first case below such that $\Gamma_{i+1} \neq \Gamma_i$ or $\mathcal{C}_{i+1} \neq \mathcal{C}_i$.

For all $i \geq 0$, $a \in \Sigma$, $q_0 \in Q_0$, we say that $\mathcal{B}_{i,a,q_0}$ is *inhabited* if there exists a word $q_1 \dots q_m \in L(\mathcal{B}_{i,a,q_0})$ with $m \geq 0$ and $L(\mathcal{A}_i, q_j) \neq \emptyset$ for all j, $1 \leq j \leq m$. This condition can be decided in polynomial time using a classical state marking algorithm (see e.g. [9]).

REN if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to b(x)$, and $q_0 \in Q_0$,
then $\Gamma_{i+1} = \Gamma_i \cup \{\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}, \mathsf{f}_{a,q_0} \xrightarrow{\varepsilon} \mathsf{f}_{b,q_0}\}$, and $\mathcal{C}_{i+1} = \mathcal{C}_i$.

$\mathsf{INS}_{\mathsf{first}}$ if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to a(p\,x)$, and $q_0 \in Q_0$,
then $\Gamma_{i+1} = \Gamma_i \cup \{\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}\}$, and $\mathcal{C}_{i+1} = \mathcal{C}_i$.

$\mathsf{INS}_{\mathsf{last}}$ if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to a(x\,p)$, and $q_0 \in Q_0$,
then $\Gamma_{i+1} = \Gamma_i \cup \{\mathsf{f}_{a,q_0} \xrightarrow{p} \mathsf{f}_{a,q_0}\}$, and $\mathcal{C}_{i+1} = \mathcal{C}_i$.

$\mathsf{INS}_{\mathsf{into}}$ if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(xy) \to a(x\,p\,y)$, $q_0 \in Q_0$, and $s \in S$ is reachable from i_{a,q_0} using the transitions of Γ_i,
then $\Gamma_{i+1} = \Gamma_i \cup \{s \xrightarrow{p} s\}$, and $\mathcal{C}_{i+1} = \mathcal{C}_i$.

RPL_1 if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to p$, $q_0 \in Q_0$, and $\mathcal{B}_{i,a,q_0}$ is inhabited,
then $\Gamma_{i+1} = \Gamma_i$, and $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{p \xrightarrow{\varepsilon} q_0\}$.

DEL if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to ()$, $q_0 \in Q_0$, and $\mathcal{B}_{i,a,q_0}$ is inhabited,
then $\Gamma_{i+1} = \Gamma_i$, and $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{() \xrightarrow{\delta} q_0\}$.

Note that the above construction, in the cases of insertion rules, add new looping transitions which summarize several insertions. Such constructions are comparable to *acceleration* techniques used in model checking [35].

Only a finite number of transitions can be added to Γ_i and $\mathcal{C}_i$, hence eventually, a fixpoint $\mathcal{A}_k$ is reached, that we will also denote $\mathcal{A}'$. The proof that the above construction is correct, *i.e.* that $L(\mathcal{A}') = post^*_{\mathcal{R}_0/\mathcal{A}_0}(L)$, is quite long but straightforward. Finally, using Lemma 1 we can remove the δ- and ε-transitions of $\mathcal{A}'$ and conclude. □

Example 5. Let $\mathcal{A}$ be the HA of Example 3, let $\mathcal{R}/\mathcal{A} = \{\mathsf{hospital}(x) \to \mathsf{hospital}(x\,p_{\mathsf{epa}})\}$, and let L be recognized by a HA $\mathcal{A}_L$ with states q_{h}, q_{pa}, q_{n}, q_{t}, q_{dr}, q_{dia}, q_{da}, q_c, a unique final state q_{h}, one transition rule $\mathsf{hospital}(q^*_{\mathsf{pa}}) \to q_{\mathsf{h}}$, and transition rules for every other symbol (patient, name,...) which mimic the corresponding transitions of $\mathcal{A}$, where every state p_a is replaced by q_a.

Let us see how the completion procedure described in the the above proof is running on this input. Following the above hypotheses, it holds that $\mathcal{B}_{\mathsf{hospital},q_{\mathsf{h}}}$ (hence Γ_0) contains 2 transitions: one loop $\mathsf{i}_{\mathsf{hospital},q_{\mathsf{h}}} \xrightarrow{q_{\mathsf{pa}}} \mathsf{i}_{\mathsf{hospital},q_{\mathsf{h}}}$ and one ε-transition $\mathsf{i}_{\mathsf{hospital},q_{\mathsf{h}}} \xrightarrow{\varepsilon} \mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}}$ ($\mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}} \xrightarrow{q_{\mathsf{pa}}} \mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}}$ and $\mathsf{i}_{\mathsf{hospital},q_{\mathsf{h}}} \xrightarrow{\varepsilon} \mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}}$ was another possibility, we choose the first option arbitrarily, wlog). The completion introduces one new looping transition $\mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}} \xrightarrow{p_{\mathsf{epa}}} \mathsf{f}_{\mathsf{hospital},q_{\mathsf{h}}}$ (above case $\mathsf{INS}_{\mathsf{last}}$), which means that in the obtained HA $\mathcal{A}'$, the transition $\mathsf{hospital}(q^*_{\mathsf{pa}}) \to q_{\mathsf{h}}$ is replaced (after normalization) by $\mathsf{hospital}(q^*_{\mathsf{pa}}\,p^*_{\mathsf{epa}}) \to q_{\mathsf{h}}$. The automaton $\mathcal{B}_{\mathsf{hospital},p_{\mathsf{h}}}$, for the rule of $\mathcal{A}$ $\mathsf{hospital}(\{p_{\mathsf{pa}}, p_{\mathsf{epa}}\}^*) \to p_{\mathsf{h}}$ is also completed similarly, and the completion stops after 2 steps. Some other transitions are added but are not important (they do not change the recognized language).

The HA $\mathcal{A}'$ obtained recognizes (in its final state q_{h}) the set of terms $\mathsf{hospital}(h_{\mathsf{pa}}\,h_{\mathsf{epa}})$, where h_{pa} and h_{epa} are finite sequences of terms respectively of the form $\mathsf{patient}(\mathsf{name}(\ldots)\,\mathsf{treatment}(\ldots))$ and $\mathsf{patient}(\mathsf{name}(\ldots))$. ◇

Example 6. Let $\mathcal{A}$ be again the HA of Example 3, let $\mathcal{R}/\mathcal{A} = \{\mathsf{hospital}(x) \to \mathsf{hospital}(p_{\mathsf{h}}\,x), \mathsf{patient}(x) \to ()\}$, and let $L = \{\mathsf{hospital}(\mathsf{patient})\}$. This language is recognized by the following HA

$$\mathcal{A}_L = \big(\{q_0, q_1\}, \{q_1\}, \{\mathsf{patient} \to q_0, \mathsf{hospital}(q_0) \to q_1\}\big).$$

The completion adds some looping transitions $\mathsf{i}_{\mathsf{hospital},q_1} \xrightarrow{p_{\mathsf{h}}} \mathsf{i}_{\mathsf{hospital},q_1}$ and $\mathsf{i}_{\mathsf{hospital},p_{\mathsf{h}}} \xrightarrow{p_{\mathsf{h}}} \mathsf{i}_{\mathsf{hospital},p_{\mathsf{h}}}$ (case $\mathsf{INS}_{\mathsf{first}}$), and two δ-transitions $() \xrightarrow{\delta} q_0$ and $() \xrightarrow{\delta} p_{\mathsf{pa}}$ (case DEL). Note that the later δ-transitions are added because $L(\mathcal{B}_{0,\mathsf{patient},q_0}) \neq \emptyset$ (it contains the empty word) and $L(\mathcal{B}_{0,\mathsf{patient},p_{\mathsf{pa}}}) \neq \emptyset$. Some other transitions are added but are not important. Hence, in addition to the transitions of $\mathcal{A}$

and $\mathcal{A}_L$, the completed HA $\mathcal{A}'$ contains the transitions $\mathsf{hospital}(p_{\mathsf{h}}^* \, q_0) \to q_1$, $\mathsf{hospital}(\{p_{\mathsf{h}}^* \, p_{\mathsf{pa}} \, p_{\mathsf{epa}}\}^*) \to p_{\mathsf{h}}$, and the 2 above δ-transitions. $\diamond$

Corollary 1. *Typechecking is EXPTIME-complete for the iteration of rules of type in* $\mathsf{UFO}_{\mathsf{reg}}$ *and PTIME-complete when the output type is given by a deterministic and complete HA.*

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 2, $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in})$ is a HA language. Hence $post^*_{\mathcal{R}/\mathcal{A}}(\tau_{in}) \cap \overline{\tau_{out}}$ is a HA language. The size of the HA for the complement $\overline{\tau_{out}}$ can be exponential in the size of the HA for τ_{out} if this latter HA is non-deterministic, and it is polynomial otherwise. Testing the emptiness of the above intersection language solves the problem.

Regarding the lower bounds, the EXPTIME-hardness follows the fact that the inclusion problem is already EXPTIME-complete for ranked tree automata [37], and the PTIME-hardness from the fact that the inclusion problem is PTIME-hard for deterministic HA. $\square$

Regarding the problem of type synthesis, if we are given $\mathcal{R}/\mathcal{A}$ and an input type τ_{in} as a HA, Theorem 2 provides in PTIME an output type presented as a HA of polynomial size.

4.2. CF Forward Type Inference for the Iteration of All Update Primitives

Theorem 1 is not true for all the rules of UFO: some rules of UFO do not preserve HA languages in general. It is evident for RPL.

Example 7. Let $\Sigma = \{a, b, c, d\}$, let $L = \{d(a)\}$, let $\mathcal{A}$ be the HA

$$\mathcal{A} = \big(\Sigma, \{p_a, p_b, p_c\}, \{\}, \{a \to p_a, b \to p_b, c \to p_c\}\big),$$

and let $\mathcal{R}/\mathcal{A}$ contains the single RPL rule $a(x) \to p_b p_a p_c$. It holds that $post^*_{\mathcal{R}/\mathcal{A}}(L) = \{d(b^n \, a \, c^n) \mid n \geq 0\}$, which is a CF-HA language but not a HA language. $\diamond$

The examples below show that there is also no preservation of HA languages for rules of type $\mathsf{DEL_s}$ and combinations of $\mathsf{INS_{before}}$, $\mathsf{INS_{after}}$ and REN.

Example 8. 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. $\diamond$

The above example involves the primitive $\mathsf{DEL_s}$ which does not correspond to the deletion primitive of [8]. However, a similar result can be achieved by simpler insertion rules.

Example 9. Let $\Sigma = \{a, b, a', b', c\}$ and $\mathcal{A} = \big(\Sigma, \{p_0, p_1, p_2, p_3\}, \emptyset, \{0 \to p_0, 1 \to p_1, 2 \to p_2, 3 \to p_3\}\big)$, and let us consider a $\mathcal{R}/\mathcal{A}$ with the following rules

$$\begin{array}{lcllcll}
a(x) & \to & p_0 \, a(x) & b(x) & \to & p_1 \, b(x) & \\
a(x) & \to & a'(x) & b(x) & \to & b'(x) & \\
a'(x) & \to & a'(x) \, p_2 & b'(x) & \to & b'(x) \, p_3 & \\
a'(x) & \to & b(x) & b'(x) & \to & a(x) & b'(x) \to ()
\end{array}$$

The intersection of the closure $post^*_{\mathcal{R}/\mathcal{A}}\big(\{c(ab)\}\big)$ with $\big\{c\big((0^*1^*)^n(3^*2^*)^m\big) \mid n, m \geq 0\big\}$ (the latter is a HA language) is the set $\big\{c\big((0^*1^*)^n(3^*2^*)^n\big) \mid n \geq 0\big\}$ which is not a HA language. $\diamond$

However, we prove in Theorem 2 that the image of a HA language under arbitrary iterations of rewrite rules in the class UFO belongs to CF-HA. In the construction of Theorem 2, we use, as a technical convenience, a new kind of transitions called *collapsing transitions with lookahead* (*cl*-transitions).
Assume given a set of states Q and a finite set P of expressions of the form $a(L)$ where $a \in \Sigma$ and $L \subseteq Q^*$ is a context-free language (presented e.g. by a CF grammar over Q). A *cl*-transition is a rule of the form $L' \to q$ where L' is a CF language over $Q \cup P$.

The extension of the move relation for these transitions is defined by $u[u_1 \dots u_n] \xrightarrow[\mathcal{A}]{} u[q]$ if every u_i is either a state of Q or a term $a(q_1 \dots q_m)$ with $a \in \Sigma$ and $q_1, \dots, q_m \in Q$, there exists a *cl*-transition $L' \to q$ of $\mathcal{A}$ and $u_1 \dots u_n \in L'$. The languages of HA and CF-HA extended with *cl*-transitions are defined accordingly.

Lemma 4. *Every CF-HA $\mathcal{A}$ extended with a finite number of cl-transitions can be transformed in polynomial time into a CF-HA $\mathcal{A}'$ (without collapsing or cl-transitions) such that $L(\mathcal{A}') = L(\mathcal{A}) \cap \mathcal{T}(\Sigma)$.*

Proof. For every *cl*-transitions $L' \to q$, we introduce: one new state $q_{a(L)}$ for every expression $a(L)$ occurring in L', one new CF-HA transition $a(L) \to q_{a(L)}$ and one new collapsing transition $L'' \to q$ where L'' is obtained from L' by replacing every expression $a(L)$ by $q_{a(L)}$. The *cl*-transition $L' \to q$ is then deleted. After these operations, we obtain a new CF-HA $\mathcal{A}''$ extended with collapsing transitions. It is easy to see that $L(\mathcal{A}'') = L(\mathcal{A})$. We can conclude by applying Lemma 2. $\square$

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

Proof. The construction is essentially the same as for Theorem 1, with the incremental definition of a finite sequence $\mathcal{A}_0, \mathcal{A}_1, \dots$ by addition of transitions. The only difference is that the automata $\mathcal{A}_0, \dots$ are HA with collapsing and *cl*-transitions, and not simply δ- and ε-transition as in the case of Theorem 1. More precisely, let $\mathcal{A} = (\Sigma, P, P^{\mathsf{f}}, \Theta)$ and $\mathcal{A}_L = (\Sigma, Q_L, Q^{\mathsf{f}}_L, \Delta_L)$ be normalized and complete disjoint HAs, with $\mathcal{A}_L$ recognizing L, and let $\mathcal{A}_0 = (\Sigma, Q_0, Q^{\mathsf{f}}_0, \Delta_0)$ be their normalized disjoint union. For each $a \in \Sigma$ and $q \in Q_0$, the automaton $\mathcal{B}_{a,q} = \big(Q_0, S_{a,q}, \mathsf{i}_{a,q}, \{\mathsf{f}_{a,q}\}, \Gamma_{a,q}\big)$, is defined like in the proof of Theorem 1, and Γ_0 is the disjoint union of all transition sets $\Gamma_{a,q}$. We define an increasing (wrt inclusion) sequence $\Gamma_0, \Gamma_1, ...$ by addition of finite automata transitions. For all $i \geq 0$, and all $a \in \Sigma$ and $q \in Q_0$, we let $\mathcal{B}_{i,a,q}$ be the automaton $(Q_0, S, \mathsf{i}_{a,q}, \{\mathsf{f}_{a,q}\}, \Gamma_i)$.

Moreover, we also consider $\mathcal{C}_0 = \emptyset$ and define an increasing sequence $\mathcal{C}_0, \mathcal{C}_1, \dots$ by addition of collapsing transitions and *cl*-transitions. The construction of both sequences work in parallel, by the iteration of a case analysis, for each type of rewrite rule, until a fixpoint is reached. At each step $i \geq 0$, we let $\Delta_i := \big\{a\big(\mathcal{B}_{i,a,q}\big) \to q \;\big|\; a \in \Sigma, q \in Q_0\big\} \cup \mathcal{C}_i$ and $\mathcal{A}_i = (\Sigma, Q_0, Q^{\mathsf{f}}_0, \Delta_i)$.

The constructions for the cases REN, $\mathsf{INS}_{\mathsf{first}}$, $\mathsf{INS}_{\mathsf{last}}$, $\mathsf{INS}_{\mathsf{into}}$, are exactly the same as in the proof of Theorem 1. The cases DEL and RPL_1 are particular cases of RPL. The other cases, $\mathsf{INS}_{\mathsf{before}}$, $\mathsf{INS}_{\mathsf{after}}$, RPL, DEL_s are treated altogether below, by the addition of *cl*-transitions. We use in this construction a binary relation $\trianglerighteq$ over Σ defined as the reflexive and transitive closure of $\{(a,b) \mid a(x) \to b(x) \in \mathcal{R}/\mathcal{A}\}$.

> let $a \in \Sigma$, $q_0 \in Q_0$, and let $P_1 = \{p \mid a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}_0\}$, $P_2 = \{p \mid a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}_0\}$, $P_3 = \{p_1 \dots p_n \mid a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}_0\}$, and $P_4 = L(\mathcal{B}_{i,a,q_0})$ if $\mathcal{R}/\mathcal{A}_0$ contains a rule $a(x) \to x$, or $P_4 = \emptyset$ otherwise.
> If $P_1 \cup P_2 \cup P_3 \cup P_4 \neq \emptyset$ and $\mathcal{B}_{i,a,q_0}$ is inhabited, then $\Gamma_{i+1} = \Gamma_i$, and $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{P_1^* \left(b(L(\mathcal{B}_{i,b,q_0})) \mid P_3 \mid P_4\right) P_2^* \to q_0 \mid a \trianglerighteq b\}$.

Note that the above sets P_1 to P_4 only depend on $\mathcal{R}/\mathcal{A}$, and not on the current state of the construction. In this case, like for the other insertion primitives, we use an acceleration. This trick was necessary to the correctness of the construction.

We assume that the automata $\mathcal{B}_{i,a,q_0}$ in the above *cl*-transitions are dynamically updated at each construction step, i.e. at step $i+1$, $P_1^* \left(b(L(\mathcal{B}_{i,b,q_0})) \mid P_3 \mid L(\mathcal{B}_{i,a,q_0})\right) P_2^* \to q_0$ (when $P_4 \neq \emptyset$) is replaced by $P_1^* \left(b(L(\mathcal{B}_{i+1,b,q_0})) \mid P_3 \mid L(\mathcal{B}_{i+1,a,q_0})\right) P_2^* \to q_0$. The idea is that the language below a is defined by pointers to the states i_{a,q_0} and f_{a,q_0} and the current transition set Γ_i.

With the two above tricks (acceleration and pointers), the incremental construction terminates with a fixpoint after a polynomial number k of iterations. Let $\mathcal{A}' = \mathcal{A}_k$. It is a HA extended with collapsing and *cl*-transitions. The proof that $L(\mathcal{A}') = post^*_{\mathcal{R}_0/\mathcal{A}_0}(L)$ follows the same principle as for Theorem 1. It can be found in Appendix B. Finally, using Lemma 2 and Lemma 4 we can conclude that $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is a CF-HA language. □

The above result still holds when L is a CF-HA language. In this case, we consider CF grammars instead of finite automata in the completion process. The case of rewrite rules handled in Theorem 1 are treated by the addition of production rules of regular grammars. The other cases are treated exactly as above.

Example 10. Let us come back again to Example 3, with a slight variation $\mathcal{A}'$ of $\mathcal{A}$, obtained by the replacement of the rule $\mathsf{patient}(p_{\mathsf{n}}) \to p_{\mathsf{epa}}$ by $\mathsf{patient}'(p_{\mathsf{n}}) \to p_{\mathsf{epa}}$, where $\mathsf{patient}'$ is a new symbol (for patients without treatment). We consider the following PTRS

$$\mathcal{R}/\mathcal{A}' = \{\mathsf{patient}'(x) \to \mathsf{patient}(x), \mathsf{patient}(x) \to p_t\,\mathsf{patient}(x),\}$$

and the language $L = \{\mathsf{hospital}(\mathsf{patient}')\}$ recognized by the following HA

$$\mathcal{A}_L = \big(\{q_0, q_1\}, \{q_1\}, \{\mathsf{patient}' \to q_0, \mathsf{hospital}(q_0) \to q_1\}\big).$$

The above completion procedure introduce the following automata ε-transitions (case REN): $\mathsf{i}_{\mathsf{patient},q} \xrightarrow{\varepsilon} \mathsf{i}_{\mathsf{patient}',q}$ $\mathsf{f}_{\mathsf{patient}',q} \xrightarrow{\varepsilon} \mathsf{f}_{\mathsf{patient},q}$ for all states q (actually only p_{pa}, p_{epa} and q_0 are relevant in this case). Moreover, the following *cl*-transitions are added (case INS_{before}): $p_t^*\,\mathsf{patient}(L(\mathcal{B}_{k,\mathsf{patient},q})) \to q$ where q is p_{pa} or q_0 and k is the last completion step.

The automaton obtained after the completion recognizes $\mathsf{hospital}(\mathsf{patient}')$ and all the terms of the form $\mathsf{hospital}(t_1 \dots t_n\ \mathsf{patient})$ where $n \geq 0$ and $t_1, \dots, t_n \in L(\mathcal{A}', p_\mathsf{t})$. ◇

Corollary 2. *Typechecking is EXPTIME-complete for* UFO *and PTIME-complete when the output type is given by a deterministic and complete HA.*

Proof. The proof for the upper bound works as in Corollary 1, because the intersection of a CF-HA and a HA language is a CF-HA language (there is an effective PTIME construction of an CF-HA of polynomial size), and emptiness of CF-HA is decidable in PTIME. The arguments of Corollary 1 for lower bounds are still valid here because HA are special cases of CF-HA. □

Regarding the problem of type synthesis for a $\mathcal{R}/\mathcal{A}$ in UFO, if an input type τ_{in} is given as a HA or CF-HA, then Theorem 2 provides in PTIME an output type, presented as a CF-HA of polynomial size. Unlike HA, CF-HA are not popular type schemes, but HA solely do not permit to extend the results of Theorem 2 to the whole class UFO, in particular for the operations $\mathsf{INS}_{\mathsf{before}}$, $\mathsf{INS}_{\mathsf{after}}$ and RPL of [8], as we have seen above.

One may wonder to what extent the CF-HA produced by Theorem 2, given a HA for L and a $\mathcal{R}/\mathcal{A}$, is actually an HA. This problem is actually undecidable, since the problem of knowing whether a given CF language is regular is undecidable and every CF language can be described by the closure $post^*_{\mathcal{R}/\mathcal{A}}(L)$ for some $\mathcal{A}$, $\mathcal{R}$ and L (see Example 7).

4.3. Regular Backward Type Inference for the Iteration of All Update Primitives

Since UFO rules do not preserve HA languages (by iteration), we may attempt to perform typechecking using pre^* computations (backward type inference), like e.g. for k-pebble tree transducer [28]. The next theorem shows that this is indeed possible, though EXPTIME, since the class of HA languages is preserved by pre^* when using UFO rules.

Theorem 3. *Given a HA $\mathcal{A}$ on Σ and a PTRS $\mathcal{R}/\mathcal{A} \in$ UFO, for all HA language L, $pre^*_{\mathcal{R}/\mathcal{A}}(L)$ is the language of a HA of size exponential and which can be constructed in EXPTIME in the size of $\mathcal{R}/\mathcal{A}$ and of an HA recognizing L.*

Proof. Let $\mathcal{A} = (\Sigma, P, P^\mathsf{f}, \Theta)$, and let $\mathcal{A}_L = (\Sigma, Q_L, Q^\mathsf{f}_L, \Delta_L)$ be a HA recognizing L. We assume *wlog* that both $\mathcal{A}$ and $\mathcal{A}_L$ are normalized and complete. Let $P_0 = P \cup Q_L$, $Q^\mathsf{f}_0 = Q^\mathsf{f}_L$ and $\Delta_0 = \Theta \cup \Delta_L$. We assume given, for each $a \in \Sigma$, $q \in Q_0$, a finite automaton $\mathcal{B}_{a,q} = (P_0, S_{a,q}, \mathsf{i}_{a,q}, \{\mathsf{f}_{a,q}\}, \Gamma_{a,q})$ recognizing the regular language $L_{a,q}$ in the transition $a(L_{a,q}) \to q \in \Delta_0$, and following the assumptions described in the proof of Theorem 1. Moreover, for all $a, b \in \Sigma$, and all states s, s' of $\mathcal{B}_{a,q}$, we add a new state $q_{b,s,s'}$. In other words, we let

$$Q_0 = P_0 \cup \{q_{b,s,s'} \mid a, b \in \Sigma, q \in P_0, s, s' \in S_{a,q}\}.$$

Let $\mathcal{C}$ be the smallest set of of finite automata over Q_0 such that:

- $\mathcal{C}$ contains every $\mathcal{B}_{a,q}$ for $a \in \Sigma$, $q \in P_0$,
- for all $\mathcal{B} \in \mathcal{C}$, $\mathcal{B} = (Q_0, S, \mathsf{i}, F, \Gamma)$ and all states $s, s' \in S$, the automaton $\mathcal{B}_{s,s'} := (Q_0, S, s, \{s'\}, \Gamma)$ is in $\mathcal{C}$,

- for all $\mathcal{B} \in \mathcal{C}$, $\mathcal{B} = (Q_0, S, \mathsf{i}, F, \Gamma) \in \mathcal{C}$, $q \in Q_0$ and all states $s, s' \in S$, the automata $(Q_0, S, \mathsf{i}, F, \Gamma \cup \{s \xrightarrow{q} s'\})$ and $(Q_0, S, \mathsf{i}, F, \Gamma \cup \{s \xrightarrow{\varepsilon} s'\})$, respectively denoted by $\mathcal{B} + s \xrightarrow{q} s'$ and $\mathcal{B} + s \xrightarrow{\varepsilon} s'$ also belong to $\mathcal{C}$.

Note that $\mathcal{C}$ is finite with this definition, though exponential. Moreover, every $\mathcal{B} \in \mathcal{C}$ has a unique final state which we will denote $\mathsf{f}_\mathcal{B}$ and its initial state is denoted $\mathsf{i}_\mathcal{B}$. For the sake of notations, we make no distinction below between $\mathcal{B} \in \mathcal{C}$ and its language $L(\mathcal{B})$.

Let $\mathcal{A}_0 = (\Sigma, Q_0, Q_0^\mathsf{f}, \Delta_0)$. We will incrementally add transitions to $\mathcal{A}_0$, according to the rules of $\mathcal{R}/\mathcal{A}$, until a fixpoint automaton is reached which recognizes $pre^*_{\mathcal{R}/\mathcal{A}}(L)$. More precisely, we construct a finite sequence of HA $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$, where for all $i \leq k$, $\mathcal{A}_i = (\Sigma, Q_0, Q_0^\mathsf{f}, \Delta_i)$, and such that the language of the final element $\mathcal{A}_k$ is $pre^*_{\mathcal{R}/\mathcal{A}}(L)$. The transition sets Δ_i are constructed recursively by iteration of the following case analysis until a fixpoint is reached (only a finite number of transitions can be added in the construction).

We use below an extension of the move relation of HA, from states to set of states (single states are considered as singleton sets), defined by $a(M_1 \ldots M_p) \hookrightarrow_{\Delta_i} q_0$ (where $M_1, \ldots, M_p \subseteq Q_0$ and $q_0 \in Q_0$) iff there exists a transition $a(N) \to q \in \Delta_i$ such that $M_1 \ldots M_p \subseteq N$.

REN if $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$, $\mathcal{B} \in \mathcal{C}$ and $q \in Q_0$, such that $b(\mathcal{B}) \hookrightarrow_{\Delta_i} q$, then let $\Delta_{i+1} := \Delta_i \cup \{a(\mathcal{B}) \to q\}$.

$\mathsf{INS}_\mathsf{first}$ if $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$, $\mathcal{B} \in \mathcal{C}$ and $q, q_p \in Q_0$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$ and $a(q_p\,\mathcal{B}) \hookrightarrow_{\Delta_i} q$, then $\Delta_{i+1} := \Delta_i \cup \{a(\mathcal{B}) \to q\}$.

$\mathsf{INS}_\mathsf{last}$ if $a(x) \to b(x\,p) \in \mathcal{R}/\mathcal{A}$, $\mathcal{B} \in \mathcal{C}$ and $q, q_p \in Q_0$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$ and $a(\mathcal{B}\,q_p) \hookrightarrow_{\Delta_i} q$, then $\Delta_{i+1} := \Delta_i \cup \{a(\mathcal{B}) \to q\}$.

$\mathsf{INS}_\mathsf{into}$ if $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, $\mathcal{B} \in \mathcal{C}$, s, s' are states of $\mathcal{B}$, and $q, q_p \in Q_0$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[\mathcal{B}]{q_p} s'$, and $a(\mathcal{B}) \hookrightarrow_{\Delta_i} q$ then $\Delta_{i+1} := \Delta_i \cup \big\{a(\mathcal{B} + s \xrightarrow{\varepsilon} s') \to q\big\}$.

$\mathsf{INS}_\mathsf{before}$ if $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $\mathcal{B}, \mathcal{B}' \in \mathcal{C}$, s, s' are states of $\mathcal{B}$, and $q, q_p, q' \in Q_0$ such that $b(\mathcal{B}) \to q \in \Delta_i$, $a(\mathcal{B}') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[\mathcal{B}]{q_p q'} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(\mathcal{B} + s \xrightarrow{q'} s') \to q\big\}$.

$\mathsf{INS}_\mathsf{after}$ if $a(x) \to a(x)\,p \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $\mathcal{B}, \mathcal{B}' \in \mathcal{C}$, s, s' are states of $\mathcal{B}$, and $q, q_p, q' \in Q_0$ such that $b(\mathcal{B}) \to q \in \Delta_i$, $a(\mathcal{B}') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[\mathcal{B}]{q' q_p} s'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(\mathcal{B} + s \xrightarrow{q'} s') \to q\big\}$.

RPL if $a(x) \to p_1 \ldots p_n \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $\mathcal{B}, \mathcal{B}' \in \mathcal{C}$, s, s' are states of $\mathcal{B}$, and $q, q', q_1, \ldots, q_n \in Q_0$ such that $b(\mathcal{B}) \to q \in \Delta_i$, $a(\mathcal{B}') \hookrightarrow_{\Delta_i} q'$, $L(\mathcal{A}_i, q_j) \cap L(\mathcal{A}, p_j) \neq \emptyset$ for all $1 \leq j \leq n$, $s \xrightarrow[\mathcal{B}]{q_1 \ldots q_n} s'$ then $\Delta_{i+1} := \Delta_i \cup \big\{b(\mathcal{B} + s \xrightarrow{q'} s')) \to q\big\}$.

DEL if $a(x) \to () \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $\mathcal{B}, \mathcal{B}' \in \mathcal{C}$, s is a state of $\mathcal{B}$, $q, q' \in Q_0$ such that $b(\mathcal{B}) \to q \in \Delta_i$, $a(\mathcal{B}') \hookrightarrow_{\Delta_i} q'$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(\mathcal{B} + s \xrightarrow{q'} s) \to q\big\}$.

DEL_s if $a(x) \to x \in \mathcal{R}/\mathcal{A}$, $b \in \Sigma$, $\mathcal{B} \in \mathcal{C}$, s, s' are states of $\mathcal{B}$, $q \in Q_0$ such that $b(\mathcal{B}) \to q \in \Delta_i$, then $\Delta_{i+1} := \Delta_i \cup \big\{b(\mathcal{B} + s \xrightarrow{q_{a,s,s'}} s') \to q, a(\mathcal{B}_{s,s'}) \to q_{a,s,s'}\} \cup \{a(q^\mathsf{f}) \to q^\mathsf{f} \mid q^\mathsf{f} \in Q_0^\mathsf{f}\}\big\}$.

Note that RPL_1 is a special case of RPL. No state is added to the original automaton $\mathcal{A}_0$ and all the transitions added involve horizontal languages of the set $\mathcal{C}$, which is finite (every transition of $\mathcal{A}'$ has the form $a(\mathcal{B}) \to q_0$ for some $\mathcal{B} \in \mathcal{C}$), hence the iteration of the above operations terminates with an automaton $\mathcal{A}'$. We show in Appendix C that $L(\mathcal{A}') = pre^*_{\mathcal{R}/\mathcal{A}}(L)$. □

Example 11. We modify slightly the DTD Example 3 in order to represent departments in an hospital, with two additional transition rules for $\mathcal{A}$: (for the sake of readability, we write below h for $\mathsf{hospital}$, p for $\mathsf{patient}$ and s for $\mathsf{surgery}$):

$$\mathsf{h}(\{p_{\mathsf{dpt}}, p_{\mathsf{pa}}, p_{\mathsf{epa}}\}^*) \to p_{\mathsf{h}},\ \mathsf{s}(\{p_{\mathsf{pa}}, p_{\mathsf{epa}}\}^*) \to p_{\mathsf{dpt}},$$

and let $\mathcal{R}/\mathcal{A}$ contains the $\mathsf{DEL}_{\mathsf{s}}$ rule of Example 4: $\mathsf{s}(x) \to x$, which suppress the department s while keeping its patients.

Let $L = \mathsf{h}(\mathsf{p})$ be recognized by the HA $\mathcal{A}_L = \big(\{q_0, q_1\}, \{q_1\}, \{\mathsf{p} \to q_0, \mathsf{h}(q_0) \to q_1\}\big)$. Following our assumptions on the finite automata (see proof of Theorem 1), $\mathcal{B}_{\mathsf{p},q_0}$ and $\mathcal{B}_{\mathsf{h},q_1}$ contain each one transition, respectively $\mathsf{i}_{\mathsf{p},q_0} \xrightarrow{\varepsilon} \mathsf{f}_{\mathsf{p},q_0}$ and $\mathsf{i}_{\mathsf{h},q_1} \xrightarrow{q_0} \mathsf{f}_{\mathsf{h},q_1}$.

The rule $\mathsf{s}(x) \to x$ causes the addition of the transitions $\mathsf{s}(q_1) \to q_1$, $\mathsf{p}\big(\mathcal{B}_{\mathsf{p},q_0} + s \xrightarrow{q_{\mathsf{s},s,s'}} s'\big) \to q_0$, $\mathsf{s}(\mathcal{B}_{\mathsf{p},q_0,s,s'}) \to q_{\mathsf{s},s,s'}$ for all $s, s' \in \{\mathsf{i}_{\mathsf{p},q_0}, \mathsf{f}_{\mathsf{p},q_0}\}$, and $\mathsf{h}\big(\mathcal{B}_{\mathsf{h},q_1} + s \xrightarrow{q_{\mathsf{s},s,s'}} s'\big) \to q_0$, $\mathsf{s}(\mathcal{B}_{\mathsf{h},q_1,s,s'}) \to q_{\mathsf{s},s,s'}$ for all $s, s' \in \{\mathsf{i}_{\mathsf{h},q_1}, \mathsf{f}_{\mathsf{h},q_1}\}$.

In particular, the terms $\mathsf{h}(\mathsf{s}(\mathsf{p}))$ and $\mathsf{h}(\mathsf{s}(\mathsf{s}(\mathsf{p})))$ are accepted with the respective runs $q_1(q_{\mathsf{s},\mathsf{i}_{\mathsf{h},q_1},\mathsf{f}_{\mathsf{h},q_1}}(q_0))$ and $q_1(q_{\mathsf{s},\mathsf{i}_{\mathsf{h},q_1},\mathsf{f}_{\mathsf{h},q_1}}(q_{\mathsf{s},\mathsf{i}_{\mathsf{h},q_1},\mathsf{f}_{\mathsf{h},q_1}}(q_0)))$. ◇

Regarding the problem of type synthesis for a $\mathcal{R}/\mathcal{A} \in \mathsf{UFO}$, if an output type τ_{out} is given, then Theorem 3 provides in EXPTIME an input type, presented as a HA of exponential size.

4.4. Toward typechecking XQuery Updates

We have considered above the application to documents of a given input type of arbitrary sequences of update primitives (amongst a given finite set of primitives). Our results can be applied to the verification of set of allowed updates primitives, see in particular Section 5 and application to the verification of access control policies for updates.

In the XQuery Update Facility [8] however, an update is not just an update primitive. It is an XQuery expression u containing update primitives which is interpreted in several phases on a given document t. First, u is converted into a sequence w of primitive updates to be applied to t (the *pending list* generated by u over t). The sequence w is then analysed (sanity check) and possibly modified into w' (according to a priority order for the update primitives) and afterward w' is applied to t, giving the updated document t'.

[2] propose an abstract representation of the set of the pending lists which can be generated by an update expression u over the trees in an input type L (e.g. an HA language), by a regular expression Ω over the alphabet of all possible update primitives. Ω is called *effect expression* in [2]. The alphabet of update primitives corresponds roughly to the set of rewrite rules presented in Figure 1. Note that for a given Σ, a fixed HA $\mathcal{A}$, and a fixed number n for RPL, this set is finite. Let us call it $\mathcal{R}/\mathcal{A}$. Hence it seems that the effect of the application of u to the trees in L can be represented in our settings, by the set

$$post^{\Omega}_{\mathcal{R}/\mathcal{A}}(L) = \{h' \in \mathcal{H}(\Sigma) \mid \exists h \in L, \exists r_1 \dots r_n \in \Omega, h \xrightarrow[r_1]{} \dots \xrightarrow[r_n]{} h'\}.$$

The computation of $post^{\Omega}_{\mathcal{R}/\mathcal{A}}(L)$ (given a finite PTRS $\mathcal{R}/\mathcal{A}$, Ω a regular expression over $\mathcal{R}/\mathcal{A}$ and an HA recognizing L) is an interesting problem, apparently related to the study of languages of context-free grammars with regulated rewriting [11], and is left to further work.

5. ACP for XML Updates

In this last section we study some models of Access Control Policies (ACP) for the update operations defined in Section 3, and the verification problems for these ACP. We consider two kind of formalisms from the literature for the specification of XML ACPs. The first formalism is the most widespread. It consists in defining an ACP as a set of updates rules, partitioned into authorized and forbidden operations. The second one is a more recent proposal of [17] based on [15], where the ACP is defined by adding security annotations to a DTD.

5.1. Local Consistency of Rule-based ACPs

An ACP for XML updates can be defined by a pair $(\mathcal{R}_{\mathsf{a}}/\mathcal{A}, \mathcal{R}_{\mathsf{f}}/\mathcal{A})$ of PTRS, where $\mathcal{R}_{\mathsf{a}}$ contains allowed operations and $\mathcal{R}_{\mathsf{f}}$ contains forbidden operations (see e.g. [6]). Such an ACP is called *inconsistent* [6, 17] if some forbidden operation can be simulated through a sequence of allowed operations, *i.e.* if there exists $t, u \in \mathcal{T}(\Sigma)$ such that $t \xrightarrow[\mathcal{R}_{\mathsf{f}}/\mathcal{A}]{} u$ and $t \xrightarrow[\mathcal{R}_{\mathsf{a}}/\mathcal{A}]{}^{*} u$.

Example 12. 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_1 is forbidden: $\mathsf{name}(x) \to p_{\mathsf{n}}$. 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 new $\mathsf{patient}$, then we have an inconsistency in the sense of [6] 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 problem individually for terms of $\mathcal{T}(\Sigma)$. More precisely, we solve the following problem called *local inconsistency*: given a HA $\mathcal{A}$ over Σ and a term $t \in \mathcal{T}(\Sigma)$, an ACP $(\mathcal{R}_{\mathsf{a}}/\mathcal{A}, \mathcal{R}_{\mathsf{f}}/\mathcal{A})$ is locally inconsistent if there exists $u \in \mathcal{T}(\Sigma)$ such that $t \xrightarrow[\mathcal{R}_{\mathsf{f}}/\mathcal{A}]{} u$ and $t \xrightarrow[\mathcal{R}_{\mathsf{a}}/\mathcal{A}]{}^{*} u$?

Theorem 4. *Local inconsistency is decidable in PTIME for* UFO *ACPs.* □

Proof. It can be easily shown that the set $\{u \in \mathcal{T}(\Sigma) \mid t \xrightarrow[\mathcal{R}_{\mathsf{f}}/\mathcal{A}]{} u\}$ is the language of a HA of size polynomial and constructed in PTIME on the sizes of $\mathcal{A}$, $\mathcal{R}_{\mathsf{f}}$ and t. By Theorem 2, $post^{*}_{\mathcal{R}_{\mathsf{a}}/\mathcal{A}}(\{t\})$ is the language of a CF-HA of polynomial size and constructed in polynomial time on the sizes of $\mathcal{A}$, $\mathcal{R}_{\mathsf{a}}$ and t. The ACP is locally inconsistent w.r.t. t iff the intersection of the two above languages is not empty, and this property can be tested in PTIME. □

It is shown in [17] that inconsistency is undecidable for an ACP defined by a pair of rewrite system $(\mathcal{R}_{\mathsf{a}}, \mathcal{R}_{\mathsf{f}})$ of a kind strictly more general than the above PTRS (roughly, they extend the PTRS with the possibility to select the rewrite positions by XPath expressions). Moreover, for such rewrite systems, the problem of reachability (whether a given term t can be obtained from a given term s using instances of rules of $\mathcal{R}_{\mathsf{a}}$ which are not in $\mathcal{R}_{\mathsf{f}}$) is also undecidable [29], therefore local consistency is undecidable as well in this case. A decidable fragment is also presented in [29]. It is an open question whether inconsistency is decidable or not for PTRS of type $\mathsf{UFO}_{\mathsf{reg}}$ or UFO.

5.2. Local Consistency of DTD-based ACPs

We recall that a DTD over Σ is function D that maps Σ to regular expressions over Σ. The dependency graph of a DTD D is a directed graph on the set of vertices Σ such that the set of edges contains all (a, b) such that b occurs in the regular expression $D(a)$. A DTD is non recursive if this graph is acyclic.

Following the principle of DTD-based ACPs [15], [17] have proposed the language $\mathsf{XACU}_{\mathsf{annot}}$ for the definition of ACP for XML updates in presence of a DTD D. The idea is to add to D some security annotations specifying the authorizations for the update operations for XML documents valid for D. This formalism of [15, 17] imposes the condition that every document t to which we want to apply an update operation (under the given ACP) must be valid for the DTD D.

In our rewrite-based formalism, the latter condition may be expressed by adding global constraints to the parametrized rewrite rules of Section 2.4. These global constraints restrict the rewrite relation to terms in a given HA language. Theorem 5 below shows that, unfortunately, adding such constraints to parametrized rewrite rules of type REN or RPL 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 PTRS, denoted $\mathcal{R}/\mathcal{A}$, (see Section 2.4.2) and $L \subseteq \mathcal{T}(\Sigma)$ an HA language. We say that L is the constraint of R. The rewrite relation generated by the PGTRS is defined as the restriction of the relation defined in Section 2.4.2 to ground terms such that for the application of a rule $\ell \to r \in \mathcal{R}/\mathcal{A}$ to a term t, we require that $t \in L$.

Theorem 5. *Reachability is undecidable for PGTRS's with rules in* $\mathsf{UFO}_{\mathsf{reg}}$ *and constraint given by a non recursive DTD.* □

Proof. The proof is a variant of the one given by A. Spelten [38] for subterm and flat prefix rewriting. We reduce the halting problem of a Deterministic Turing Machine (TM) $\mathcal{M}$ that work on half a tape (unbounded on the right).

TM configurations are encoded as flat terms. We consider the same tape alphabet $\Gamma = \{0, 1, \flat\}$, ($\flat$ is the blank symbol) of $\mathcal{M}$, let $S = \{s_1, s_2, \ldots, s_n\}$ be the state set of $\mathcal{M}$ and Θ be the set of instructions of $\mathcal{M}$. We consider the following alphabet Σ for the representation of the configurations of $\mathcal{M}$.

$$\Sigma := \{g\} \cup \{0, 1, \flat\} \cup (S \times \Gamma) \cup (\Theta \times \Gamma)\}.$$

For instance, the TM configuration with tape $abcde\flat\flat\ldots$, symbol d under head, state s, will be represented by the following flat term of $\mathcal{T}(\Sigma)$: $g(abc\langle s, d\rangle e\flat\flat)$.

We shall also use a trivial HA automata $\mathcal{A} = (\Sigma, Q, Q, \Delta)$ which recognizes only constant symbols by taking $Q = \{p_\sigma \mid \sigma \in \Sigma\}$ and $\Delta = \{\sigma \to p_\sigma \mid \sigma \in \Sigma\}$.

We define a PGTRS $\mathcal{R}/\mathcal{A}$ such that every transition of $\mathcal{M}$ can be simulated by a sequence of (at most three) rewrite steps with $\mathcal{R}/\mathcal{A}$. Let us first introduce some standard auxiliary PTRS rules and some word regular languages for controlling rule applications.

For each instruction θ of $\mathcal{M}$ of type: *"In state s reading a go to state r and write b"*, we define the following TRS rule: $\langle s, a\rangle(x) \to \langle r, b\rangle(x)$.
We also define the regular word language $L_{\langle s,a\rangle} = \Gamma^*\langle s, a\rangle\,\Gamma^*$.

For each instruction θ of $\mathcal{M}$ of type: *"In state s reading a go to state r and move right"*, we define the following PTRS rules of types REN and $\mathsf{INS}_{\mathsf{after}}$ (we recall that $p_\flat$ is a state of $\mathcal{A}$):

$$\begin{array}{rcll} b(x) & \to & \langle\theta,b\rangle(x) & \text{for all } b\in\{0,1,\flat\} \\ \langle\theta,b\rangle(x) & \to & \langle r,b\rangle(x) & \text{for all } \ b\in\Gamma. \end{array} \qquad \begin{array}{rcl} \flat(x) & \to & \flat(x)\ p_\flat \\ \langle s,a\rangle(x) & \to & a(x) \end{array}$$

We also define the regular word languages:

$$\begin{array}{rcl} L_{\langle s,a\rangle} & = & \Gamma^*\langle s,a\rangle\,\Gamma^* \\ L_{\langle\theta,a\rangle} & = & \Gamma^*\langle\theta,a\rangle\,\Gamma^* \end{array} \qquad L_{\langle s,a\rangle\langle\theta,b\rangle} = \Gamma^*\langle s,a\rangle\langle\theta,b\rangle\,\Gamma^* \quad \text{for all } \ b\in\Gamma.$$

For each instruction θ of $\mathcal{M}$ of type: *"In state s reading a go to state r and move left"*, we define the following TRS rules:

$$\begin{array}{rcll} b(x) & \to & \langle\theta,b\rangle(x) & \text{for all } b\in\{0,1\} \\ \langle\theta,b\rangle(x) & \to & \langle r,b\rangle(x) & \text{for all } b\in\{0,1\} \end{array} \qquad \langle s,a\rangle(x) \to a(x)$$

We also define the regular word languages:

$$\begin{array}{rcl} L_{\langle s,a\rangle} & = & \Gamma^*\langle s,a\rangle\,\Gamma^* \\ L_{\langle\theta,a\rangle} & = & \Gamma^*\langle\theta,a\rangle\,\Gamma^* \end{array} \qquad L_{\langle\theta,b\rangle\langle s,a\rangle} = \Gamma^*\langle\theta,b\rangle\langle s,a\rangle\,\Gamma^* \ \text{ for all } \ b\in\{0,1\}.$$

The constraint of the PGTRS will be defined by the non recursive DTD $D : g \to L$ where L is the finite union of the regular languages associated to the instructions of $\mathcal{M}$ as above. Since the machine to be simulated is deterministic, the union is disjoint.

Our final PGTRS is given by $\mathcal{R}/\mathcal{A}$ and L so that the rewrite rules in $\mathcal{R}/\mathcal{A}$ can only be applied to terms satisfying the DTD D. With the above constraint, the PGTRS rules of $\mathcal{R}/\mathcal{A}$ can only be applied to terms valid for the DTD D, ensuring a correct chaining for the application of these rules.

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 $t_1 \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t_2$. The theorem follows. □

Note that the above result also holds for PGTRS's whose rules are ground (without variables nor parameters): in the above rewrite rules, every variable x could be replaced by the empty hedge (), and every parameter such as $p_\flat$ could be replaced by the corresponding ground term $\flat$. Hence the above result can be contrasted with the decidability of reachability for ground term rewriting [19].

In [1] the authors study the more general problem of *satisfiability* for active XML documents in the context and unranked unordered terms. This property is shown decidable for insertions constrained by an unordered DTD, but undecidable when they are constrained by an unordered HA.

Corollary 3. *Local inconsistency is undecidable for PGTRS with rules in* UFO *and with constraint given by a non recursive DTD.*

6. Conclusion

We have proposed a model for the primitive XML updates operations of [8] based on term rewriting systems parametrized by hedge automata (PTRS), and

studied the problems of type inference and typechecking for arbitrary iteration of such operations. We have also studied some extensions of the model with restriction of the application of update operations to documents conforming to a fixed non recursive DTD (PGTRS). Finally, we have shown how to apply our results to show the decidability of the property of local inconsistency of access control policies for XML updates.

One of our main results of forward type inference (Theorem 2) requires to use CF-HA (a strict extension of hedge automata) for output types. One may wonder whether this result could be adapted to compute regular over-approximations of output types, leading to an approximating forward type inference algorithm, in an approach similar to e.g. [39]. It could also be interesting to apply a similar approach for studying updates of unranked unordered trees.

Acknowledgments. The authors wish to thank Serge Abiteboul, Pierre Bourhis, Sebastian Maneth and Luc Segoufin for discussions about XML updates and access control.

References

[1] S. Abiteboul, P. Bourhis, B. Marinoiu, Satisfiability and relevance for queries over active documents, in: J. Paredaens, J. Su (Eds.), Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, ACM, 2009, pp. 87–96.

[2] M. Benedikt, J. Cheney, Semantics, types and effects for xml updates, in: DBPL, Database Programming Languages - DBPL 2009, 12th International Symposium, Lyon, France, August 24, 2009. Proceedings, volume 5708 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 1–17.

[3] V. Benzaken, G. Castagna, A. Frisch, Cduce: An xml-centric general-purpose language, in: Proceedings of the ACM International Conference on Functional Programming.

[4] A. Bouajjani, B. Jonsson, M. Nilsson, T. Touili, Regular model checking, in: Proc. of the 12th Int. Conf. on Computer Aided Verification, volume 1855 of *LNCS*, pp. 403–418.

[5] A. Bouajjani, T. Touili, On computing reachability sets of process rewrite systems, in: J. Giesl (Ed.), Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings, volume 3467 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 484–499.

[6] L. Bravo, J. Cheney, I. Fundulaki, Accon: checking consistency of xml write-access control policies, in: EDBT 2008, 11th International Conference on Extending Database Technology, Nantes, France, March 25-29, 2008, Proceedings, volume 261 of *ACM International Conference Proceeding Series*, ACM, 2008, pp. 715–719.

[7] S. C. Lim, S.H. Son, Access control of xml documents considering update operations, in: Proc. of ACM Workshop on XML Security.

[8] D. Chamberlin, J. Robie, Xquery update facility 1.0, W3C Candidate Recommendation. http://www.w3.org/TR/xquery-update-10/, 2009.

[9] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, Tree automata techniques and applications, Available on: http://tata.gforge.inria.fr/, 2007. Release October, 12th 2007.

[10] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, P. Samarati, Securing xml documents, in: Proceedings of the 7th International Conference on Extending Database Technology (EDBT 2000), volume 1777 of *Lecture Notes in Computer Science*, Springer, 2000, pp. 121–135.

[11] J. Dassow, G. Paun, A. Salomaa, Handbook of formal languages, Handbook of Formal Languages, volume 2, Springer, 1997, pp. 101–154.

[12] N. Dershowitz, J.P. Jouannaud, Rewrite systems, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science (Vol. B: Formal Models and Semantics), North-Holland, Amsterdam, 1990, pp. 243–320.

[13] J. Engelfriet, S. Maneth, H. Seidl, Deciding equivalence of top-down xml transformations in polynomial time, J. Comput. Syst. Sci. 75 (2009) 271–286.

[14] J. Engelfriet, H. Vogler, Macro tree transducers, J. Comp. Syst. Sci. 31 (1985) 71–146.

[15] W. Fan, C.Y. Chan, M. Garofalakis, Secure xml querying with security views, in: Proceedings of the 2004 ACM SIGMOD international conference on Management of data (SIGMOD'04), ACM, New York, NY, USA, 2004, pp. 587–598.

[16] G. Feuillade, T. Genet, V. Viet Triem Tong, Reachability Analysis over Term Rewriting Systems, Journal of Automated Reasoning 33 (3-4) (2004) 341–383.

[17] I. Fundulaki, S. Maneth, Formalizing xml access control for update operations, in: SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies, ACM, New York, NY, USA, 2007, pp. 169–174.

[18] P.A. Gardner, G.D. Smith, M.J. Wheelhouse, U.D. Zarfaty, Local hoare reasoning about dom, in: PODS '08: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, New York, NY, USA, 2008, pp. 261–270.

[19] R. Gilleron, Decision problems for term rewrite systems and recognizable tree languages, in: 8^th Annual Symposium on Theoretical Aspects of Computer Science, volume 480 of *Lecture Notes in Computer Science*, pp. 148–159.

[20] H. Hosoya, J. Vouillon, B.C. Pierce, Regular expression types for xml, ACM Trans. Program. Lang. Syst. 27 (2005) 46–90.

[21] F. Jacquemard, M. Rusinowitch, Closure of Hedge-automata languages by Hedge rewriting, in: A. Voronkov (Ed.), Proceedings of the 19th International Conference on Rewriting Techniques and Applications (RTA'08), volume 5117 of *Lecture Notes in Computer Science*, Springer, Hagenberg, Austria, 2008, pp. 157–171.

[22] M. Kay, XSL Transformations (XSLT) 2.0, W3C Working Draft, World Wide Web Consortium, 2003. Available at `http://www.w3.org/TR/xslt20`.

[23] C. Löding, Ground tree rewriting graphs of bounded tree width, in: Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS'02), volume 2285 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 559–570.

[24] C. Löding, A. Spelten, Transition graphs of rewriting systems over unranked trees, in: L. Kucera, A. Kucera (Eds.), Mathematical Foundations of Computer Science 2007, 32nd International Symposium, MFCS 2007, Ceský Krumlov, Czech Republic, August 26-31, 2007, Proceedings, volume 4708 of *Lecture Notes in Computer Science*, pp. 67–77.

[25] S. Maneth, A. Berlea, T. Perst, H. Seidl, Xml type checking with macro tree transducers, in: 24th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS), pp. 283–294.

[26] S. Maneth, T. Perst, H. Seidl, Exact xml type checking in polynomial time, in: T. Schwentick, D. Suciu (Eds.), Proceedings of the 11th International Conference on Database Theory (ICDT 2007), volume 4353 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 254–268.

[27] W. Martens, 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), ACM, 2004, pp. 23–34.

[28] T. Milo, D. Suciu, V. Vianu, Typechecking for xml transformers, J. of Comp. Syst. Sci. 66 (2003) 66–97.

[29] N. Moore, Computational complexity of the problem of tree generation under fine-grained access control policies, Inf. Comput. 209 (2011) 548–567.

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

[31] M. Murata, D. Lee, M. Mani, Taxonomy of xml schema languages using formal language theory, in: In Extreme Markup Languages.

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

[33] H. Ohsaki, H. Seki, T. Takai, Recognizing boolean closed a-tree languages with membership conditional rewriting mechanism, in: Proc. of the 14th Int. Conference on Rewriting Techniques and Applications (RTS 2003),

volume 2706 of *Lecture Notes in Computer Science*, Springer Verlag, 2003, pp. 483–498.

[34] T. Perst, H. Seidl, Macro forest transducers, Information Processing Letters 89 (2004) 141–149.

[35] A. Pnueli, E. Shahar, Liveness and acceleration in parameterized verification, in: In CAV˘201900, Springer Verlag, 2000, pp. 328–343.

[36] T. Schwentick, Automata for xml - a survey, J. Comput. Syst. Sci. 73 (2007) 289–315.

[37] H. Seidl, Deciding equivalence of finite tree automata, SIAM Journal of Computing 19 (1990) 424–437.

[38] A. Spelten, Rewriting Systems over Unranked Trees, Master's thesis, Diplomarbeit, RWTH Aachen, 2006.

[39] T. Touili, Computing transitive closures of hedge transformations, in: In Proc. 1st International Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS'07), eWIC Series, British Computer Society, 2007.

Appendix

A. Proof of Theorem 1

We prove in this section the correctness of the automata construction presented in Section 4.1. More precisely, we show that the fixpoint HA with δ- and ε-transitions $\mathcal{A}'$, constructed in the proof of Theorem 1, given a PTRS $\mathcal{R}/\mathcal{A}$ and a HA $\mathcal{A}_L$ recognizing the language L, is such that $L(\mathcal{A}') = post^*_{\mathcal{R}/\mathcal{A}}(L)$.

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

Proof. We show more generally that for all $q \in Q_0$ and all $t \in L(\mathcal{A}', q)$, there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

The above computation of $\mathcal{A}'$ on t involves some transition rules of $\mathcal{A}'$ which may be in $\mathcal{A}_0$ or have been added at some step i of the completion procedure (remember that the rules constructed in the proof of Theorem 1 can be standard HA rules of the form $a(L) \to q$, or δ- of ε-rules). We call *index* of a transition rule r, either 0 if $r \in \Delta_0$ or the smallest $i > 0$ such that $r \in \Delta_i$ and $r \notin \Delta_{i-1}$. The proof of the Lemma works by induction on the multiset $\mathcal{M}$ of the indexes of the transitions rules of $\mathcal{A}'$ involved in the reduction $t \xrightarrow[\mathcal{A}']{*} q$.

Base case. If all the indexes in $\mathcal{M}$ are 0, then $t \in L(\mathcal{A}_0, q)$ and we let $u = t$.

Induction step. Assume that the reduction $t \xrightarrow[\mathcal{A}']{*} q$ has a measure $\mathcal{M}$, and that it involves a transition rule $r \in \Delta_{i+1} \setminus \Delta_i$, r of index $i + 1 > 0$. We analyze the case that permitted the construction of this rule r.

REN. Assume that the application of r in the reduction $t \xrightarrow[\mathcal{A}']{*} q$ involves an ε-transition $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$ of $\mathcal{B}_{i+1,b,q_0}$, and that this ε-transition was added to Γ_{i+1} because $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$. Let

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

be the above reduction of t by $\mathcal{A}'$, such that the above ε-transition is involved in the step $t[b(q_1 \dots q_n)] \xrightarrow[r]{} t[q_0]$, where the transition $r = b\big(L(\mathcal{B}_{i+1,b,q_0})\big) \to q_0$ is applied. Hence $q_1 \dots q_n \in L(\mathcal{B}_{i+1,b,q_0})$, with $\mathsf{i}_{b,q} \xrightarrow[\mathcal{B}_{i+1,b,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$, and the first step in this computation is $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$. The last step must be $\mathsf{f}_{a,q_0} \xrightarrow{\varepsilon} \mathsf{f}_{b,q_0}$, using an ε-transition added to Γ_{i+1} in the same step as $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$. By removing these first and last steps, we get $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$, hence $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0})$. Therefore, we have a reduction

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

Hence $t' \in L(\mathcal{A}', q)$, and the measure $\mathcal{M}'$ of the above reduction $t' \xrightarrow[\mathcal{A}']{*} q$ is strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ 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 $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

INS$_{\mathsf{first}}$. Assume that the application of r in the reduction $t \xrightarrow[\mathcal{A}']{*} q$ involves a transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$ which has been added to Γ_{i+1} because $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$. Let

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

be the above reduction of t by $\mathcal{A}'$, with $t_p \in L(\mathcal{A}', p)$, and such that the transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$ is involved in the step $t[a(p\,q_1 \dots q_n)] \xrightarrow[r]{} t[q_0]$, where the the transition $r = a\big(L(\mathcal{B}_{i+1,a,q_0})\big) \to q_0$ is applied. Hence $p\,q_1 \dots q_n \in L(\mathcal{B}_{i+1,a,q_0})$, with $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i+1,a,q_0}]{p\,q_1 \dots q_n} \mathsf{f}_{a,q_0}$, and the first step in this computation is $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$. By deleting this first step, we obtain the derivation $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$, hence $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0})$. Therefore, we have a reduction

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

(meaning that $t' \in L(\mathcal{A}', q)$) with a measure $\mathcal{M}'$ strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}]{*} t'$.

Moreover, the measure of the sub-reduction $t_p \xrightarrow[\mathcal{A}']{*} p$ is strictly smaller than $\mathcal{M}$. Hence by induction hypothesis, there exists $u_p \in L(\mathcal{A}, p)$ such that $u_p \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t_p$, and we have

$$t' = t[a(h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(u_p\,h)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(t_p\,h)] = t$$

(the first step uses $a(x) \to a(p\,x)$). We conclude that $u \xrightarrow[\mathcal{R}]{*} t$.

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

$\mathsf{INS}_{\mathsf{into}}$**.** Assume that the application of r in the reduction $t \xrightarrow[A']{*} q$ involves an transition $s \xrightarrow{p} s$ which was added to Γ_{i+1} because $a(xy) \to a(xpy) \in \mathcal{R}/\mathcal{A}$ and $s \in S$ is reachable from i_{a,q_0} for some $q_0 \in Q_0$. Let

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

be the above reduction of $\mathcal{A}'$, with $t_p \in L(\mathcal{A}', p)$, and such that the transition $s \xrightarrow{p} s$ is involved in the step $t[a(q_1 \dots q_n\,p\,q'_1 \dots q'_m)] \xrightarrow[\mathcal{A}']{} t[q_0]$, where the transition $a(\mathcal{B}_{i+1,a,q_0}) \to q_0$ is applied. More precisely, we assume that $q_1 \dots q_n\,p\,q'_1 \dots q'_m \in L(\mathcal{B}_{i+1,a,q_0})$, because

$$\mathsf{i}_{a,q} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n} s \xrightarrow[\mathcal{B}_{i+1,a,q_0}]{p} s \xrightarrow[\mathcal{B}_{i,a,q_0}]{q'_1 \dots q'_m} \mathsf{f}_{a,q}.$$

By deleting the middle step $s \xrightarrow{p} s$, we obtain $\mathsf{i}_{a,q} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n\,q'_1 \dots q'_m} \mathsf{f}_{a,q}$, hence $q_1 \dots q_n\,q'_1 \dots q'_m \in L(\mathcal{B}_{i,a,q_0})$. Therefore, we have a reduction

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

(hence $t' \in L(\mathcal{A}', q)$) with a measure strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Moreover, the measure of the sub-reduction $t_p \xrightarrow[\mathcal{A}']{*} p$ is strictly smaller than $\mathcal{M}$, hence by induction hypothesis, there exists $u_p \in L(\mathcal{A}, p)$ such that $u_p \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t_p$. Hence, we have a reduction

$$t' = t[a(h\,\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[a(h\,u_p\,\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[a(h\,t_p\,\ell)] = t$$

whose first step involves $a(xy) \to b(xpy)$, and we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

RPL_1. Assume that $r = p \xrightarrow{\varepsilon} q_0$, and that this ε-transition was added to $\mathcal{C}_{i+1}$ because $a(x) \to p \in \mathcal{R}/\mathcal{A}$ and $\mathcal{B}_{i,a,q_0}$ is inhabited. Let

$$t = t[t_p] \xrightarrow[\mathcal{A}']{*} t[p] \xrightarrow[r]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

be the above reduction of t by $\mathcal{A}'$, with $t_p \in L(\mathcal{A}', p)$. The measure of the sub-reduction $t_p \xrightarrow[\mathcal{A}']{*} p$ is strictly smaller than $\mathcal{M}$, hence by induction hypothesis, there exists $u_p \in L(\mathcal{A}, p)$ such that $u_p \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t_p$.

Moreover, by hypothesis, there exists $q_1 \dots q_m \in L(\mathcal{B}_{i,a,q_0})$ $(m \geq 0)$ and some terms $s_1 \in L(\mathcal{A}_i, q_1), \dots, s_m \in L(\mathcal{A}_i, q_m)$ and hence we have a reduction

$$t' = t[a(s_1 \dots s_m)] \xrightarrow[\mathcal{A}_i]{*} t[a(q_1 \dots q_m)] \xrightarrow[\mathcal{A}_i]{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. It holds that

$$t' = t[a(s_1 \dots s_m)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[u_p] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t[t_p] = t$$

(the first rewrite step applies $a(x) \to p$), and hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

DEL. Assume that $r = () \xrightarrow{\delta} q_0$, and that this δ-transition was added to $\mathcal{C}_{i+1}$ because $a(x) \to () \in \mathcal{R}/\mathcal{A}$ and $\mathcal{B}_{i,a,q_0}$ is inhabited. Let

$$t = t[\,] \xrightarrow[r]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

be the above reduction of t by $\mathcal{A}'$. By hypothesis, there exists $q_1 \dots q_m \in L(\mathcal{B}_{i,a,q_0})$ and there exists some terms $s_1 \in L(\mathcal{A}_i, q_1), \dots, s_m \in L(\mathcal{A}_i, q_m)$. Hence we have a reduction

$$t' = t[a(s_1 \dots s_m)] \xrightarrow[\mathcal{A}_i]{*} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

of measure strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t'$. Moreover, $t' = t[a(s_1 \dots s_m)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[\,] = t$, and hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. □

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

Proof. We show that for all $u \in L$, if $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$, then $u \in L(\mathcal{A}')$, 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 involves a rewrite rule $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$:

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

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 $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$, i.e. $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$ (remember that k is the last construction step - fixpoint of the construction). By construction, the ε-transitions $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$ and $\mathsf{f}_{a,q_0} \xrightarrow{\varepsilon} \mathsf{f}_{b,q_0}$ have been added to the automaton $\mathcal{B}_{k,a,q_0}$, at some construction step $i \leq k$. Hence $\mathsf{i}_{b,q_0} \xrightarrow[\mathcal{B}_{k,b,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$ and $q_1 \dots q_n \in L(\mathcal{B}_{k,b,q_0})$. Therefore there exists a reduction sequence:

$$t = t[b(h)] \xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q_L^\mathsf{f}$$

and $t \in L(\mathcal{A}')$.

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

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q_L^\mathsf{f}$$

with $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$, i.e. $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$. By construction, the transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$ has been added to $\mathcal{B}_{k,a,q_0}$. Hence $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{p} \mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$, i.e. $p\,q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$ and there exists a reduction sequence

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

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

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

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

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

with $q_1 \dots q_n\,q'_1 \dots q'_m \in L(\mathcal{B}_{k,a,q_0})$, i.e. $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} s \xrightarrow[\mathcal{B}_{k,a,q_0}]{q'_1 \dots q'_m} \mathsf{f}_{a,q_0}$ for some state $s \in S$. By construction, the looping transition $s \xrightarrow{p} s$ has been added to Γ_{i+1} at some step $i \leq k$. Hence $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} s \xrightarrow[\mathcal{B}_{k,a,q_0}]{p} s \xrightarrow[\mathcal{B}_{k,a,q_0}]{q'_1 \dots q'_m} \mathsf{f}_{a,q_0}$, i.e. $q_1 \dots q_n\,p\,q'_1 \dots q'_m \in L(\mathcal{B}_{k,a,q_0})$ and we have

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

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

RPL_1. 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:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}.$$

Hence $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$ and $L(\mathcal{A}_i, q_j) \neq \emptyset$ for $1 \leq j \leq n$. It follows that an ε-transition $p \xrightarrow{\varepsilon} q_0$ has been added to $\mathcal{A}'$, and there exists a reduction sequence

$$t = t[t_p] \xrightarrow[\mathcal{A}']{*} t[p] \xrightarrow[\mathcal{A}']{\varepsilon} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}.$$

Hence $t \in L(\mathcal{A}')$.

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

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

By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}.$$

It follows that $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$ and $L(\mathcal{A}_i, q_j) \neq \emptyset$ for $1 \leq j \leq n$. Hence, a δ-transition $() \xrightarrow{\delta} q_0$ has been added to $\mathcal{A}'$, and there exists a reduction sequence

$$t = t[()] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}},$$

hence $t \in L(\mathcal{A}')$. □

B. Proof of Theorem 2

We prove here the correctness of the CF-HA construction presented in Section 4.2, i.e. we show that $L(\mathcal{A}') = post^*_{\mathcal{R}/\mathcal{A}}(L)$ holds, when $\mathcal{A}'$ is the HA with collapsing and cl-transitions constructed in the proof of Theorem 2, given a PTRS $\mathcal{R}/\mathcal{A} \in \mathsf{UFO}$ and a HA $\mathcal{A}_L$ recognizing the language L. It follows that $post^*_{\mathcal{R}/\mathcal{A}}(L)$ is a CF-HA language by Lemmas 2 and 4.

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

Proof. We show more generally that for all $q \in Q_0$ and all hedge h such that $h \xrightarrow[\mathcal{A}']{*} q$, there exists a term $u \in L(\mathcal{A}_0, q) = L(\mathcal{A}_L, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h$. As in the case of Lemma 5 (Theorem 1), the proof works by induction on the multiset $\mathcal{M}$ of the indexes of the transitions rules rules of $\mathcal{A}'$ involved in the reduction $h \xrightarrow[A']{*} q$.

The base case is identical as for Lemma 5. For the induction step, we assume that the reduction $h \xrightarrow[A']{*} q$ has a measure $\mathcal{M}$, and that it applies a transition rule $r \in \Delta_{i+1} \setminus \Delta_i$, of index $i + 1 > 0$. We analyze the case that permitted the construction of this rule r.

The cases of rewrite rules of type REN, $\mathsf{INS}_{\mathsf{first}}$, $\mathsf{INS}_{\mathsf{last}}$, $\mathsf{INS}_{\mathsf{into}}$, and DEL are treated similarly as in the proof of Lemma 5. The only little difference is that, because of the collapsing (and cl) transitions, an hedge of length larger than 1 can be recognized by $\mathcal{A}'$ in a single state. Hence every hypothesis $t \xrightarrow[\mathcal{A}']{*} q$, for some term t, in the proof of Lemma 5 must be generalized to $h \xrightarrow[\mathcal{A}']{*} q$, for some

hedge h. Let us detail the changes for the cases REN and $\mathsf{INS}_{\mathsf{first}}$ below. The cases $\mathsf{INS}_{\mathsf{last}}$, $\mathsf{INS}_{\mathsf{into}}$ are treated similarly.

REN. Assume that $r = b\big(L(\mathcal{B}_{i+1,b,q_0})\big) \to q_0$ and that the application of this rule in the reduction $h \xrightarrow[\mathcal{A}']{*} q$ involves an ε-transition $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$ of $\mathcal{B}_{i+1,b,q_0}$, which has been added to Γ_{i+1} because $a(x) \to b(x) \in \mathcal{R}/\mathcal{A}$. The above reduction $h \xrightarrow[\mathcal{A}']{*} q$ has the form

$$h = h[b(h_0)] \xrightarrow[\mathcal{A}']{*} h[b(q_1 \dots q_n)] \xrightarrow[r]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q$$

such that the above ε-transition is involved in the step $h[b(q_1 \dots q_n)] \xrightarrow[r]{} h[q_0]$, where the the transition $r = b\big(L(\mathcal{B}_{i+1,b,q_0})\big) \to q_0$ is applied. Hence $q_1 \dots q_n \in L(\mathcal{B}_{i+1,b,q_0})$, with $\mathsf{i}_{b,q} \xrightarrow[\mathcal{B}_{i+1,b,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$, and the first step in this computation is $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$. The last step must be $\mathsf{f}_{a,q_0} \xrightarrow{\varepsilon} \mathsf{f}_{b,q_0}$, using an ε-transition added to Γ_{i+1} in the same step as $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$. By removing these first and last steps, we obtain $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$, hence $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0})$. Therefore, we have a reduction

$$h' = h[a(h_0)] \xrightarrow[\mathcal{A}']{*} h[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}_i]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q.$$

with a measure $\mathcal{M}'$ strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h'$. Since $h' = t[a(h_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} h[b(h_0)] = h$, with $a(x) \to b(x)$, we conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$.

$\mathsf{INS}_{\mathsf{first}}$. Assume that $r = a\big(L(\mathcal{B}_{i+1,a,q_0})\big) \to q_0$ and that the application of this rule in the reduction $h \xrightarrow[\mathcal{A}']{*} q$ involves a transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$ which has been added to Γ_{i+1} because $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$. The above reduction of $h \xrightarrow[\mathcal{A}']{*} q$ has the form

$$h = h[a(h_p\, h_0)] \xrightarrow[\mathcal{A}']{*} h[a(p\, q_1 \dots q_n)] \xrightarrow[r]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q$$

where the hedges h_p and h_0 are such that $h_p \xrightarrow[\mathcal{A}']{*} p$ and $h_0 \xrightarrow[\mathcal{A}']{*} q_1 \dots q_n$, and the transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0} \in \Gamma_{i+1} \backslash \Gamma_i$ is involved in the step $t[a(p\, q_1 \dots q_n)] \xrightarrow[r]{} t[q_0]$, where the the transition r is applied. It means that $p\, q_1 \dots q_n \in L(\mathcal{B}_{i+1,a,q_0})$, with $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i+1,a,q_0}]{p\, q_1 \dots q_n} \mathsf{f}_{a,q_0}$, and the first step in this computation is $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$. By deleting this first step, we obtain the derivation $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{i,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$, and hence $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0})$. Therefore, we have a reduction

$$h' = h[a(h_0)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure $\mathcal{M}'$ strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}]{*} h'$.

Moreover, the measure of the sub-reduction $h_p \xrightarrow[\mathcal{A}']{*} p$ is strictly smaller than $\mathcal{M}$. Hence by induction hypothesis, there exists $u_p \in L(\mathcal{A}_0, p) = L(\mathcal{A}, p)$ such that $u_p \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h_p$, and we have

$$h' = h[a(h_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} h[a(u_p\, h_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[a(h_p\, h_0)] = h$$

(the first step uses $a(x) \to a(p\,x)$). We conclude that $u \xrightarrow[\mathcal{R}]{*} h$.

The cases RPL_1 and DEL are subcases of RPL. Let us now detail the four remaining cases, $\mathsf{INS}_{\mathsf{before}}$, $\mathsf{INS}_{\mathsf{after}}$, RPL, $\mathsf{DEL}_{\mathsf{s}}$, altogether.

Assume that $r = P_1^* \big(a(L(\mathcal{B}_{i,a,q_0})) \mid P_3 \mid P_4\big) P_2^* \to q_0$ and that this cl-transition has been added to $\mathcal{C}_{i+1}$ because one of the sets P_1, P_2, P_3, P_4 is not empty and $\mathcal{B}_{i,a,q_0}$ is inhabited. In this case, the reduction $h \xrightarrow[\mathcal{A}']{*} q$ can have one of the following 3 forms

$$h = h[h_1\, a(h_0)\, h_2] \xrightarrow[\mathcal{A}']{*} h[w_1\, a(q_1 \dots q_n)\, w_2] \xrightarrow[r]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q \qquad (\mathsf{INS}_{\mathsf{before/after}})$$

when $P_1 \cup P_2 \neq \emptyset$, $w_1 \in P_1^*$ and $w_2 \in P_2^*$ are sequences of states of $\mathcal{A}$, $h_1 \xrightarrow[\mathcal{A}']{*} w_1$, $h_2 \xrightarrow[\mathcal{A}']{*} w_2$, $h_0 \xrightarrow[\mathcal{A}']{*} q_1 \dots q_n$, and $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0})$.

$$h = h[h_1 h_0 h_2] \xrightarrow[\mathcal{A}']{*} h[w_1\, p_1 \dots p_n w_2] \xrightarrow[r]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q \qquad (\mathsf{RPL})$$

where w_1, w_2, h_1, h_2 are as above, $p_1 \dots p_n \in P_3$ because $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$ and the hedge h_0 contains some subhedges $h_{0,j} \in L(\mathcal{A}', p_j)$ for $1 \leq j \leq n$.

$$h = h[h_1 h_0 h_2] \xrightarrow[\mathcal{A}']{*} h[w_1\, q_1 \dots q_n w_2] \xrightarrow[r]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q \qquad (\mathsf{DEL}_{\mathsf{s}})$$

where w_1, w_2, h_1, h_2 are as above, $a(x) \to x \in \mathcal{R}/\mathcal{A}$, $q_1 \dots q_n \in L(\mathcal{B}_{i,a,q_0}) = P_4$, and $h_0 \xrightarrow[\mathcal{A}']{*} q_1 \dots q_n$.

Now, we analyze the three above cases.

($\mathsf{INS}_{\mathsf{before/after}}$). By construction, $\mathcal{A}'$ contains a transition $a\big(L(\mathcal{B}_{i,a,q_0})\big) \to q_0$, and hence we have

$$h' = h[a(h_0)] \xrightarrow[\mathcal{A}']{*} h[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}_i]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with a measure strictly smaller than $\mathcal{M}$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}]{*} h'$.

Moreover, for all state p occurring in w_1 there exists a subhedge h_p of h_1 such that $h_p \xrightarrow[\mathcal{A}']{*} p$, and this reduction is a sub-reduction of $h_1 \xrightarrow[\mathcal{A}']{*} w_1$ with a measure strictly smaller than $\mathcal{M}$. Hence by induction hypothesis, there exists a term $u_p \in L(\mathcal{A}_0, p)$ such that $u_p \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h_p$. The situation is similar for h_2 and w_2, and altogether, we have two hedges h_1', h_2' such that $h_1' \xrightarrow[\mathcal{A}_0]{*} w_1$, $h_2' \xrightarrow[\mathcal{A}_0]{*} w_2$, and $h_1' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h_1$, $h_2' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h_2$. Therefore, we have the following rewrite sequence

$$h' = h[a(h_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} h[h_1'\, a(h_0)\, h_2'] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[h_1\, a(h_0)\, h_2] = h$$

(the first step involves the rewrite rules of type $\mathsf{INS}_{\mathsf{before}}$ and $\mathsf{INS}_{\mathsf{after}}$ used in the definition of P_1 and P_2), and we can conclude that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h$.

(RPL). The measure of each sub-reduction $h_{0,j} \xrightarrow[\mathcal{A}']{*} p_j$ is strictly smaller than $\mathcal{M}$ for all $1 \leq j \leq n$, hence by induction hypothesis, there exists $u_j \in L(\mathcal{A}_0, p_j) = L(\mathcal{A}, p_j)$ such that $u_j \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h_{0,j}$.

Moreover, by hypothesis, $\mathcal{B}_{i,a,q_0}$ is inhabited, hence there exists $q_1 \dots q_m \in L(\mathcal{B}_{i,a,q_0})$ ($m \geq 0$), and some hedge h_0' such that $h_0' \xrightarrow[\mathcal{A}_i]{*} q_1 \dots q_m$. Hence there exists a reduction

$$h' = h[a(h_0')] \xrightarrow[\mathcal{A}_i]{*} h[a(q_1 \dots q_m)] \xrightarrow[\mathcal{A}_i]{*} h[q_0] \xrightarrow[\mathcal{A}']{*} q$$

and its measure is strictly smaller than $\mathcal{M}$, because the index of every transition rule of $\mathcal{A}_i$ is strictly smaller that the index of r, which is $i+1$. By induction hypothesis, it follows that there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h'$. Finally, it holds that

$$h' = h[a(h'_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[h'_1 a(h'_0) h'_2] \xrightarrow[\mathcal{R}/\mathcal{A}]{} h[h'_1 u_1 \dots u_n h'_2] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[h_1 h_0 h_2] = h$$

where h'_1 and h'_2 are as above (the second rewrite step applies the rule $a(x) \to p_1 \dots p_n$), and hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h$.

($\mathsf{DEL_s}$). There exists a reduction

$$h' = h[a(h_0)] \xrightarrow[\mathcal{A}']{*} h[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}_i]{} h[q_0] \xrightarrow[\mathcal{A}']{*} q$$

and its measure is strictly smaller than $\mathcal{M}$, because the index of the transition rule $a\big(L(\mathcal{B}_{i,a,q_0})\big) \to q_0$ is i and the index of r is $i+1$. Hence, by induction hypothesis, there exists $u \in L(\mathcal{A}_0, q)$ such that $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h'$. Moreover, we have the following rewrite sequence

$$h' = h[a(h_0)] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[h'_1\, a(h_0) h'_2] \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h[h_1\, a(h_0) h_2] \xrightarrow[\mathcal{R}/\mathcal{A}]{} h[h_1 h_0 h_2] = h$$

where h'_1 and h'_2 are as above and the rewrite rule $a(x) \to x$ is used, and hence $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} h$. □

Lemma 8. *$L(\mathcal{A}') \supseteq post^*_{\mathcal{R}/\mathcal{A}}(L)$.*

Proof. We show that for all term $u \in L$, 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, $u = t \in L$, and we are done since $L = L(\mathcal{A}_L) \subseteq L(\mathcal{A}')$.

Induction step ($k+1$ *rewrite steps*). We analyse the type of rewrite rule used in the last rewrite step of $u \xrightarrow[\mathcal{R}/\mathcal{A}]{*} t$. The cases of rules REN, $\mathsf{INS_{first}}$, $\mathsf{INS_{last}}$, $\mathsf{INS_{into}}$ are nearly identical as in the proof of Lemma 6 (Theorem 1). We just detail the two first cases, REN, $\mathsf{INS_{first}}$, and continue with the other rules.

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

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

By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t'[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} t''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q^{\mathsf{f}}_L$$

Note that t' may be different from t'', depending on the transition rule applied in the middle reduction step, marked with (1), which made a disappear. There are two cases:

(i) (1) applies a *cl*-transition $r = P_1^* \big(a(L(\mathcal{B}_{k,a,q_0})) \mid P_3 \mid P_4\big) P_2^* \to q_0$, with $p \in P_1$. In this case, we might have $t' \neq t''$, because the context t' may contain siblings of $a(q_1 \dots q_n)$ which are states of P_1 and P_2 removed by the application of r in (1) (hence they are not in t'').

(*ii*) (1) applies the transition $a(L(\mathcal{B}_{k,a,q_0})) \to q_0$. In this case, $t' = t''$.

In both cases, it holds that $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$, i.e. $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$ (remember that k is the last construction step - fixpoint of the construction).

We show that there exists a reduction sequence:

$$t = t[b(h)] \xrightarrow[\mathcal{A}']{*} t'[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(2)} t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$$

If we were in case (*i*), a *cl*-transition $r = P_1^* \left(b(L(\mathcal{B}_{k,b,q_0})) \mid P_3 \mid P_4\right) P_2^* \to q_0$ can be applied in (2) and $t''' = t''$. This *cl*-transition exists because $a \trianglerighteq b$.
If we were in case (*ii*), we can apply in (2) the transition $b(L(\mathcal{B}_{k,b,q_0})) \to q_0$ (and $t''' = t'$). Indeed, by construction, the ε-transitions $\mathsf{i}_{b,q_0} \xrightarrow{\varepsilon} \mathsf{i}_{a,q_0}$ and $\mathsf{f}_{a,q_0} \xrightarrow{\varepsilon} \mathsf{f}_{b,q_0}$ have been added to the automaton $\mathcal{B}_{k,a,q_0}$, hence $\mathsf{i}_{b,q_0} \xrightarrow[\mathcal{B}_{k,b,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$ and $q_1 \dots q_n \in L(\mathcal{B}_{k,b,q_0})$. Altogether, $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{first}}$. The last rewrite step of the sequence involves a rewrite rule $a(x) \to a(p\,x) \in \mathcal{R}/\mathcal{A}$, where p is a state of $\mathcal{A}$.

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

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$. Hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t'[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} t''[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$$

with the same possibilities (*i*) and (*ii*) as above for (1) and with $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$, i.e. $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{a,q_0}$. We show that there exists a reduction sequence

$$t = t[a(t_p\,h)] \xrightarrow[\mathcal{A}']{*} t'[a(p\,q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(2)} t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L.$$

with a similar case analysis as above for (2). Indeed, by construction, the transition $\mathsf{i}_{a,q_0} \xrightarrow{p} \mathsf{i}_{a,q_0}$ has been added to $\mathcal{B}_{k,a,q_0}$. Hence $\mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{p} \mathsf{i}_{a,q_0} \xrightarrow[\mathcal{B}_{k,a,q_0}]{q_1 \dots q_n} \mathsf{f}_{b,q_0}$, i.e. $p\,q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$. In conclusion $t \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{before}}$. The last rewrite step of the sequence involves a rewrite rule $a(x) \to p\,a(x) \in \mathcal{R}/\mathcal{A}$, where p is a state of $\mathcal{A}$.

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

with $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$, hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t'[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} t''[q_0] \xrightarrow[\mathcal{A}']{*} q_\mathsf{f} \in Q^\mathsf{f}_L$$

The reduction in the middle, marked with (1), can either involve

(*i*) a *cl*-transition $r = P_1^* \left(a(L(\mathcal{B}_{i,a,q_0})) \mid P_3 \mid P_4\right) P_2^* \to q_0$, with $p \in P_1$. In this case, $t' \neq t''$ because a left sibling p of $a(q_1 \dots q_n)$ is part of t' and removed from t'' by the application of r, or

(*ii*) the transition $a(L(\mathcal{B}_{k,a,q_0})) \to q_0$ (in this case, $t' = t''$).

In both cases, $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$ and

$$t = t[t_p\, a(h)] \xrightarrow[\mathcal{A}']{*} t'[p\, a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(2)} t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$$

In the above case (i), the middle rule (2) also involves the cl-transition r, because $p \in P_1$. In the case (ii), we can apply r instead of $a(L(\mathcal{B}_{k,a,q_0})) \to q_0$, and $t''' = t'$. In both cases, we can apply the rest of the reduction $t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$ and $t \in L(\mathcal{A}')$.

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

RPL. The last rewrite step of the sequence involves a rewrite rule $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$, where $p_1, \dots, p_n$ are states of $\mathcal{A}$.

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

with $t_i \in L(\mathcal{A}, p_i)$ for all $i \leq n$. By induction hypothesis, $t[a(h)] \in L(\mathcal{A}')$, hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t'[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} t''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$$

The reduction (1) as above can involves either (i) a cl-transition $r = P_1^* \left(a(L(\mathcal{B}_{i,a,q_0})) \mid P_3 \mid P_4\right) P_2^* \to q_0$ - it exists because $p_1 \dots p_n \in P_3$ (in this case $t' \neq t''$), or (ii) the transition $a(L(\mathcal{B}_{k,a,q_0})) \to q_0$ (and $t' = t''$). In both cases $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$, and

$$t = t[t_1 \dots t_n] \xrightarrow[\mathcal{A}']{*} t'[p_1 \dots p_n] \xrightarrow[\mathcal{A}']{(2)} t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$$

The middle step (2), in both cases (i) and (ii), applies the cl-transition r.
We conclude that $t \in L(\mathcal{A}')$.

DEL. It is a particular case of RPL, with $n = 0$.

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

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

By induction hypothesis, $u[a(h)] \in L(\mathcal{A}')$, hence there exists a reduction sequence:

$$t[a(h)] \xrightarrow[\mathcal{A}']{*} t'[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} t''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$$

with the same two possibilities for the step (1) as in the above cases, with $q_1 \dots q_n \in L(\mathcal{B}_{k,a,q_0})$. As $\mathcal{A}'$ contains, by construction, a cl-transition $r = P_1^* \left(a(L(\mathcal{B}_{i,a,q_0})) \mid P_3 \mid P_4\right) P_2^* \to q_0$, with $P_4 = L(\mathcal{B}_{k,a,q_0})$, we can show, by the similar case analysis as above, that

$$t = t[h] \xrightarrow[\mathcal{A}']{*} t'[q_1 \dots q_n] \xrightarrow[\mathcal{A}']{} t'''[q_0] \xrightarrow[\mathcal{A}']{*} q_{\mathsf{f}} \in Q_L^{\mathsf{f}}$$

and hence $t \in L(\mathcal{A}')$. □

C. Proof of Theorem 3

We show in this section the correctness of the automata construction presented in Section 4.3: for the HA $\mathcal{A}'$ constructed in the proof of Theorem 3, given a PTRS $\mathcal{R}/\mathcal{A} \in \mathsf{UFO}$ and a HA $\mathcal{A}_L$ recognizing a language L, it holds that $L(\mathcal{A}') = pre^*_{\mathcal{R}/\mathcal{A}}(L)$.

Lemma 9. $L(\mathcal{A}') \subseteq pre^*_{\mathcal{R}/\mathcal{A}}(L)$.

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 of indexes of transition rules of $\mathcal{A}'$ involved in the reduction, where the index of the transition rule r of $\mathcal{A}'$ is 0 if $r \in \Delta_0$ or the smallest $i > 0$ such that $r \in \Delta_i$ and $r \notin \Delta_{i-1}$.

Base case. If all the indexes in $\mathcal{M}$ are 0, then all the transition involved are in Δ_0. It means that $t \in L(\mathcal{A}_L, q)$ and we let $u = t$.

Induction step. Assume that the reduction $t \xrightarrow[\mathcal{A}']{*} q$ has a measure $\mathcal{M}$ and has the following form

$$t = t[a(h)] \xrightarrow[\mathcal{A}']{*} t[a(q_1 \dots q_n)] \xrightarrow[r]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q \tag{C.1}$$

and that the middle step $t[a(q_1 \dots q_n)] \xrightarrow[r]{} t[q_0]$ applies a transition rule $r = a(\mathcal{B}) \to q_0$ ($q_1 \dots q_n \in L(\mathcal{B})$) added to Δ_{i+1} for some $i \geq 0$. We analyse the cases which permitted the addition of this transition rule r to Δ_{i+1}.

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

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

with a measure strictly smaller than $\mathcal{M}$. 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 $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

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

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

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

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

INS$_{\mathsf{into}}$: Assume that the transition $r = a(\mathcal{B}) \to q_0$ has been added to Δ_{i+1} because $a(xy) \to a(x\,p\,y) \in \mathcal{R}/\mathcal{A}$, there is $\mathcal{B}' \in \mathcal{C}$, s, s' states of $\mathcal{B}'$, $q_0, q_p \in Q_0$, such that $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[\mathcal{B}']{q_p} s'$, $a(\mathcal{B}') \hookrightarrow_{\Delta_i} q_0$ and $\mathcal{B} = \mathcal{B}' + s \xrightarrow{\varepsilon} s'$. In this case, let $t = a(h\,\ell)$, and let reduction (C.1) have the form

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

with $q_1 \dots q_n\, q'_1 \dots q'_m \in L(\mathcal{B})$ by $\mathsf{i}_\mathcal{B} \xrightarrow[\mathcal{B}']{q_1 \dots q_n} s \xrightarrow[\mathcal{B}]{\varepsilon} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_m} \mathsf{f}_\mathcal{B}$ (remember that $\mathsf{i}_\mathcal{B}$ and $\mathsf{f}_\mathcal{B}$ are resp. the initial and final states of $\mathcal{B}$). Hence, by construction, we have $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}]{q_1 \dots q_n} s \xrightarrow[\mathcal{B}]{q_p} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_m} \mathsf{f}_{\mathcal{B}'}$ ($\mathsf{i}_{\mathcal{B}'} = \mathsf{i}_\mathcal{B}$ and $\mathsf{f}_{\mathcal{B}'} = \mathsf{f}_\mathcal{B}$) and there exists a reduction

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

for some $t_p \in L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p)$, with a measure strictly smaller than $\mathcal{M}$. 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 $t \xrightarrow[\mathcal{A}']{*} q$ has the form

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

with $r = b(\mathcal{B}) \to q_0$ ($q_1 \dots q_n \in L(\mathcal{B})$), which was added to Δ_{i+1} for some $i \geq 0$. in one of the five following cases.

$\mathsf{INS}_{\mathsf{before}}$: Assume that the transition $r = b(\mathcal{B}) \to q_0$ has been added to Δ_{i+1} because $a(x) \to p\, a(x) \in \mathcal{R}/\mathcal{A}$, there is $\mathcal{B}', \mathcal{B}'' \in \mathcal{C}$, s, s' are states of $\mathcal{B}'$, $q_0, q_p, q'_0 \in Q_0$, such that $b(\mathcal{B}') \to q_0 \in \Delta_i$, $a(\mathcal{B}'') \hookrightarrow_{\Delta_i} q'_0$, $L(\mathcal{A}_i, q_p) \cap L(\mathcal{A}, p) \neq \emptyset$, $s \xrightarrow[\mathcal{B}']{q_p\, q'_0} s'$, and $\mathcal{B} = \mathcal{B}' + s \xrightarrow{q'_0} s'$.
In this case, let $t = b(h\, a(v)\ell)$, and let the above reduction (C.2) have the form

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

with $q_1 \dots q_n\, q'_0\, q'_1 \dots q'_m \in L(\mathcal{B})$ by $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q_1 \dots q_n} s \xrightarrow[\mathcal{B}]{q'_0} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_m} \mathsf{f}_{\mathcal{B}'}$. Hence, by construction, we have $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q_1 \dots q_n} s \xrightarrow[\mathcal{B}]{q_p\, q'_0} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_m} \mathsf{f}_{\mathcal{B}'}$ ($\mathsf{i}_\mathcal{B} = \mathsf{i}_{\mathcal{B}'}$ and $\mathsf{f}_\mathcal{B} = \mathsf{f}_{\mathcal{B}'}$) and there exists a reduction

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

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

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

RPL: Assume that the transition $r = b(\mathcal{B}) \to q_0$ has been added to Δ_{i+1} because $a(x) \to p_1 \dots p_n \in \mathcal{R}/\mathcal{A}$, there is $\mathcal{B}', \mathcal{B}'' \in \mathcal{C}$, s, s' are states of $\mathcal{B}'$, $q_0, q'_0, q_1, \dots, q_n \in Q_0$, such that $b(\mathcal{B}') \to q_0 \in \Delta_i$, $a(\mathcal{B}'') \hookrightarrow_{\Delta_i} q'_0$, $L(\mathcal{A}_i, q_j) \cap L(\mathcal{A}, p_j) \neq \emptyset$ for all $j \leq n$, $s \xrightarrow[\mathcal{B}']{q_1 \dots q_n} s'$, and $\mathcal{B} = \mathcal{B}' + s \xrightarrow{q'_0} s'$.
In this case, let $t = b(ha(v)\, \ell)$, and let the above reduction (C.2) have the form

$$t = t[b(h\, a(v)\, \ell)] \xrightarrow[\mathcal{A}']{*} t[b(q'_1 \dots q'_k\, q'_0\, q'_{k+1} \dots q'_m)] \xrightarrow[r]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

with $q'_1 \dots q'_k\, q'_0\, q'_{k+1} \dots q'_m \in L(\mathcal{B})$ by $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q'_1 \dots q'_k} s \xrightarrow[\mathcal{B}]{q'_0} s' \xrightarrow[\mathcal{B}']{q'_{k+1} \dots q'_m} \mathsf{f}_{\mathcal{B}'}$. Hence, by construction, we have $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q'_1 \dots q'_k} s \xrightarrow[\mathcal{B}']{q_1 \dots q_n} s' \xrightarrow[\mathcal{B}']{q'_{k+1} \dots q'_m} \mathsf{f}_{\mathcal{B}'}$ ($\mathsf{i}_\mathcal{B} = \mathsf{i}_{\mathcal{B}'}$ and $\mathsf{f}_\mathcal{B} = \mathsf{f}_{\mathcal{B}'}$) and there exists a reduction

$$t' = t[b(h\, t_1 \dots t_n\, \ell)] \xrightarrow[\mathcal{A}']{*} t[b(q'_1 \dots q'_k\, q_1 \dots q_n\, q'_{k+1} \dots q'_m)] \xrightarrow[\mathcal{A}_i]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q$$

where for all $j \leq n$, $t_j \in L(\mathcal{A}_i, q_j) \cap L(\mathcal{A}, p_j)$, with a measure strictly smaller than $\mathcal{M}$. Therefore, by induction hypothesis, there exists $u \in L(\mathcal{A}_L, q)$ such that $t' \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. Since $t = t[a(h\, a(v)\ell)] \xrightarrow[\mathcal{R}/\mathcal{A}]{} t[b(h\, t_1 \dots t_n\, \ell)] = t'$, using the rule $a(x) \to p_1 \dots p_n$, we can conclude that $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$.

DEL: this case is a particular case of RPL with $n = 0$.

DEL$_\mathsf{s}$: Assume that the transition $r = b(\mathcal{B}) \to q_0$ has been added to Δ_{i+1} because $a(x) \to x \in \mathcal{R}/\mathcal{A}$, there is $\mathcal{B}' \in \mathcal{C}$, s, s' are states of $\mathcal{B}'$, such that $b(\mathcal{B}') \to q_0 \in \Delta_i$, and $\mathcal{B} = \mathcal{B}' + s \xrightarrow{q_{a,s,s'}} s'$.
In this case, let $t = b(ha(v)\ell)$, and let the above reduction (C.2) have the form

$$\begin{aligned} t = t[b(h\, a(v)\ell)] &\xrightarrow[\mathcal{A}']{*} t[b(q_1 \dots q_m\, a(v_1 \dots v_k)\, q'_1 \dots q'_{m'})] \\ &\xrightarrow[\mathcal{A}_{i+1}]{*} t[b(q_1 \dots q_m\, q_{a,s,s'}\, q'_1 \dots q'_{m'})] \xrightarrow[r]{} t[q_0] \xrightarrow[\mathcal{A}']{*} q \end{aligned}$$

with $v_1 \dots v_k \in L(\mathcal{B}'_{s,s'})$ by $s \xrightarrow[\mathcal{B}']{v_1 \dots v_k} s'$, and $q_1 \dots q_m\, q_{a,s,s'}\, q'_1 \dots q'_{m'} \in L(\mathcal{B})$ by $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q_1 \dots q_m} s \xrightarrow[\mathcal{B}]{q_{a,s,s'}} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_{m'}} \mathsf{f}_{\mathcal{B}'}$.

Hence, we have $\mathsf{i}_{\mathcal{B}'} \xrightarrow[\mathcal{B}']{q_1 \dots q_m} s \xrightarrow[\mathcal{B}']{v_1 \dots v_k} s' \xrightarrow[\mathcal{B}']{q'_1 \dots q'_{m'}} \mathsf{f}_{\mathcal{B}'}$ ($\mathsf{i}_{\mathcal{B}} = \mathsf{i}_{\mathcal{B}'}$ and $\mathsf{f}_{\mathcal{B}} = \mathsf{f}_{\mathcal{B}'}$) and there exists a reduction

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

with a measure strictly smaller than $\mathcal{M}$. 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 $t \xrightarrow[\mathcal{R}/\mathcal{A}]{*} u$. □

Lemma 10. *$L(\mathcal{A}') \supseteq pre^*_{\mathcal{R}/\mathcal{A}}(L)$.*

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 the rewrite rule used in the first rewrite step.

REN. Assume that the above rewrite sequence is

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

By induction hypothesis, $u[b(h)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f}$$

where $q, q_1, \dots, q_n \in Q_0$, $q^\mathsf{f} \in Q_0^\mathsf{f}$. Let $b(\mathcal{B}) \to q$ be the transition involved at the above step (1), because $q_1 \dots q_n \in \mathcal{B}$. By construction, a transition $a(\mathcal{B}) \to q$ has been added to $\mathcal{A}'$. It follows that

$$u = u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f},$$

hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{first}}$. Assume that the above rewrite sequence is

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

for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[a(t_p\, h)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[a(t_p\, h)] \xrightarrow[\mathcal{A}']{*} u[a(q_p\, q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{(1)} u[q_0] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q_0, q_p, q_1, \dots, q_n \in Q_0$ and $q^{\mathsf{f}} \in Q_0^{\mathsf{f}}$. Hence $L(\mathcal{A}', q_p) \cap L(\mathcal{A}, p)$ is not empty because it contains t_p. Let $a(\mathcal{B}) \to q$ be the transition applied at step (1). We have $q_p\, q_1 \dots q_n \in L(\mathcal{B})$, hence $\mathcal{B}$ contains a transition $\mathsf{i}_{\mathcal{B}} \xrightarrow{q_p} s$ for some state s of $\mathcal{B}$. It holds that $q_1, \dots, q_n \in L(\mathcal{B}_{s,\mathsf{f}_{\mathcal{B}}})$ and $q_p\, L(\mathcal{B}_{s,\mathsf{f}_{\mathcal{B}}}) \subseteq L(\mathcal{B})$, hence $a(q_p\, \mathcal{B}_{s,\mathsf{f}_{\mathcal{B}}}) \hookrightarrow_{\Delta_k} q_0$. It follows that the transition $a(\mathcal{B}_{s,\mathsf{f}_{\mathcal{B}}}) \to q_0$ has been added to $\mathcal{A}'$, and it permits the reduction

$$u = u[a(h)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_n)] \xrightarrow[\mathcal{A}']{} u[q_0] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}},$$

hence $u \in L(\mathcal{A}')$.

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

$\mathsf{INS}_{\mathsf{into}}$. Assume that the above rewrite sequence is

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

for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[a(h\, t_p\, \ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[a(h\, t_p\, \ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_m\, q_p\, q_1' \dots q_n')] \xrightarrow[\mathcal{A}']{(1)} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q_p, q_1, \dots, q_m, q_1', \dots, q_n' \in Q_0$ and $q^{\mathsf{f}} \in Q_0^{\mathsf{f}}$. Hence $L(\mathcal{A}', q_p) \cap L(\mathcal{A}, p)$ is not empty because it contains t_p, and the transition rule applied at the above step (1) has the form $a(\mathcal{B}) \to q$, where $q_1 \dots q_m q_p q_1' \dots q_n' \in L(\mathcal{B})$, by $\mathsf{i}_{\mathcal{B}} \xrightarrow[\mathcal{B}]{q_1 \dots q_m} s \xrightarrow[\mathcal{B}]{q_p} s' \xrightarrow[\mathcal{B}]{q_1' \dots q_n'} \mathsf{f}_{\mathcal{B}}$ for some states s, s' of $\mathcal{B}$. Therefore, a transition $a(\mathcal{B} + s \xrightarrow{\varepsilon} s') \to q$ has been added to $\mathcal{A}'$, and $q_1 \dots q_m q_1' \dots q_n' \in L(\mathcal{B} + s \xrightarrow{\varepsilon} s')$. It follows that

$$u = u[a(h\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_m q_1' \dots q_n')] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}},$$

hence that $u \in L(\mathcal{A}')$.

$\mathsf{INS}_{\mathsf{before}}$. Assume that the above rewrite sequence is

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

for some $t_p \in L(\mathcal{A}, p)$. By induction hypothesis, $u[b(h\, t_p\, a(v)\ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h\, t_p\, a(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q_1 \dots q_m\, q_p\, q' q_1' \dots q_n')] \xrightarrow[\mathcal{A}']{(1)} u[q] \xrightarrow[\mathcal{A}']{*} q^{\mathsf{f}}$$

where $q, q', q_p, q_1, \dots, q_m, q_1', \dots, q_n' \in Q_0$, $q^{\mathsf{f}} \in Q_0^{\mathsf{f}}$, 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 applied at step (1) has the form $b(\mathcal{B}) \to q$ with $\mathcal{B} \in \mathcal{C}$ and $q_1 \dots q_m q_p\, q'\, q_1' \dots q_n' \in$

$L(\mathcal{B})$, with $\mathsf{i}_\mathcal{B} \xrightarrow[\mathcal{B}]{q_1 \dots q_m} s \xrightarrow[\mathcal{B}]{q_p q'} s' \xrightarrow[\mathcal{B}]{q'_1 \dots q'_n} \mathsf{f}_\mathcal{B}$, for some of states s and s' of $\mathcal{B}$. Hence, a transition $b(\mathcal{B} + s \xrightarrow{q'} s') \to q$ has been added to $\mathcal{A}'$, and $q_1 \dots q_m q' q'_1 \dots q'_n \in L(\mathcal{B} + s \xrightarrow{q'} s')$. It follows that

$$u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q_1 \dots q_m q' q'_1 \dots q'_n)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f},$$

hence that $u \in L(\mathcal{A}')$.

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

RPL. Assume that the above rewrite sequence is

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

for some $t_1, \dots, t_n$ respectively in $L(\mathcal{A}, p_1), \dots, L(\mathcal{A}, p_n)$. By induction hypothesis, $u[b(ht_1 \dots t_n \ell)] \in L(\mathcal{A}')$, i.e. there exists a reduction sequence

$$u[b(h\, t_1 \dots t_n\, \ell)] \xrightarrow[\mathcal{A}']{*} u[b(q'_1 \dots q'_k\, q_1 \dots q_n\, q'_{k+1} \dots q'_m)] \xrightarrow[\mathcal{A}']{\rho} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f}$$

where $q, q_1, \dots, q_n, q'_1, \dots, q'_m \in Q_0$, $q^\mathsf{f} \in Q_0^\mathsf{f}$, and for all $j \leq n$, $L(\mathcal{A}', q_{p_j}) \cap L(\mathcal{A}, p_j)$ contains t_j, and the transition rule applied at step (1) has the form $b(\mathcal{B}) \to q$ with $q'_1 \dots q'_k\, q_1 \dots q_n\, q'_{k+1} \dots q'_m \in L(\mathcal{B})$, by $\mathsf{i}_\mathcal{B} \xrightarrow[\mathcal{B}]{q'_1 \dots q'_k} s \xrightarrow[\mathcal{B}]{q_1 \dots q_n} s' \xrightarrow[\mathcal{B}]{q'_{k+1} \dots q'_m} \mathsf{f}_\mathcal{B}$, for some states s and s' of $\mathcal{B}$.

Let $q' \in Q_0$ be such that $a(v) \xrightarrow[\mathcal{A}']{*} q'$. By construction, a transition $b(\mathcal{B} + s \xrightarrow{q'} s') \to q$ has been added to $\mathcal{A}'$, and $q'_1 \dots q'_k\, q'\, q'_{k+1} \dots q'_m \in L(\mathcal{B} + s \xrightarrow{q'} s')$. It follows that

$$u = u[b(ha(v)\ell)] \xrightarrow[\mathcal{A}']{*} u[a(q'_1 \dots q'_k\, q'\, q'_{k+1} \dots q'_m)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f},$$

hence that $u \in L(\mathcal{A}')$.

DEL. it is a particular case of RPL with $n = 0$.

DEL$_\mathsf{s}$. Assume that the above rewrite sequence is

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

By induction hypothesis, $u[b(hv\ell)] \in L(\mathcal{A}')$, i.e. there is a reduction sequence

$$u[b(hv\ell)] \xrightarrow[\mathcal{A}']{*} u[b(q'_1 \dots q'_m\, q_1 \dots q_n\, q'_{k+1} \dots q'_m)] \xrightarrow[\mathcal{A}']{(1)} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f}$$

where $q, q'_1, \dots, q'_m, q_1, \dots, q_n \in Q_0$ and $q^\mathsf{f} \in Q_0^\mathsf{f}$.
The transition rule applied at step (1) has the form $b(\mathcal{B}) \to q$ with $\mathcal{B} \in \mathcal{C}$ and $q'_1 \dots q'_m\, q_1 \dots q_n\, q'_{k+1} \dots q'_m \in L(\mathcal{B})$, with a sequence $\mathsf{i}_\mathcal{B} \xrightarrow[\mathcal{B}]{q'_1 \dots q'_m} s \xrightarrow[\mathcal{B}]{q_1 \dots q_n} s' \xrightarrow[\mathcal{B}]{q'_{k+1} \dots q'_m} \mathsf{f}_\mathcal{B}$, where s, s' are two states of $\mathcal{B}$.

By construction, a transition $a(\mathcal{B}_{s,s'}) \to q_{a,s,s'}$ has been added to $\mathcal{A}'$, and we have $a(v) \xrightarrow[\mathcal{A}']{*} a(q_1 \dots q_n) \xrightarrow[\mathcal{A}']{*} q_{a,s,s'}$ because $q_1 \dots q_n \in L(\mathcal{B}_{s,s'})$.

Moreover, a transition $b(\mathcal{B} + s \xrightarrow{q_{a,s,s'}} s) \to q$ has been added to $\mathcal{A}'$, and hence

$$u = u[b(h\, a(v)\, \ell)] \xrightarrow[\mathcal{A}']{*} u[a(q'_1 \dots q'_k\, q_{a,s,s'}\, q'_{k+1} \dots q'_m)] \xrightarrow[\mathcal{A}']{} u[q] \xrightarrow[\mathcal{A}']{*} q^\mathsf{f},$$

therefore $u \in L(\mathcal{A}')$. □