Learning residual finite-state tree automata from membership queries and finite positive data

Anna Kasprzik

Technical report 11-1, University of Trier, Germany
kasprzik@informatik.uni-trier.de

Abstract. In [1], Denis et al. introduce a special case of NFA, so-called *residual finite-state automata* (RFSA), where each state represents a residual language of the language recognized. RFSA have the advantageous property that there is a unique state-minimal RFSA for every regular language which makes them an attractive concept in the design of learning algorithms due to their succinctness.
Denis et al. [2] present an algorithm learning regular string languages from given data that returns an RFSA, and Bollig et al. [3] describe an algorithm that learns regular string languages from membership and equivalence queries and returns the state-minimal saturated RFSA.
The notion of RFSA can be extended to trees: *Residual finite-state tree automata* (RFTA) have been defined in [4]. For every regular tree language there is a unique state-minimal RFTA, which can be exponentially more succinct than the corresponding deterministic tree automaton.
In this paper we adapt concepts established in [3] for the inference of RFSA to trees, and we present the algorithm RESI that learns regular tree languages from membership queries and a positive sample and returns a state-minimal saturated RFTA. RESI is of polynomial complexity due to a technique which is an adaptation of the one from [5] to RFTA.

1 Introduction

The area of Grammatical Inference centers on *learning algorithms*: Algorithms that infer a description (e.g., a grammar or an automaton) for an unknown formal language from given information in finitely many steps. Various conceivable learning settings have been outlined, and based on those a range of algorithms have been developed. One of the language classes studied most extensively with respect to its algorithmical learnability is the class of regular string languages.

Possible sources of information include *membership queries* (MQs) where a learner may query an oracle if a certain element is in the target language L, and *equivalence queries* (EQs) where a learner may ask if the current hypothesis is correct and is given a counterexample if this is not the case. Moreover, a learner can for example be presented with a *positive sample*, i.e., a finite subset of L.

A significant part of the existing algorithms, of which Angluin's well-known algorithm LSTAR [6] for the inference of regular string languages from MQs and EQs was one of the first, use the device of an *observation table* (first suggested in

[7]). If such a table fulfils certain conditions then we can directly derive a deterministic finite-state automaton (DFA) from it, and if the information entered into the table is sufficient then this DFA is isomorphic to the minimal DFA for L. Consequently, the states of that DFA correspond exactly to the equivalence classes of L under the Myhill-Nerode relation (see for example [8]).

There is a price to pay for the uniqueness of the minimal DFA: In a worst case it can have exponentially many more states than a minimal non-deterministic finite-state automaton (NFA) for the same language, and as for many applications a small number of states is a desirable feature it seems worth considering if there is a way to obtain an NFA instead. In [1], Denis et al. introduce a special case of NFA, so-called *residual finite-state automata* (RFSA), where each state represents a residual language of the language recognized. There is a natural correspondence between the residual languages and the equivalence classes of a language (for the exact definition of a residual language see Section 2). Thus, contrary to NFA in general, RFSA also have the advantageous property that for every regular language the state-minimal transition-maximal RFSA is unambiguous which makes them an attractive choice for descriptions in the design of learning algorithms and their applications due to their succinctness since that RFSA can still be exponentially smaller than the corresponding minimal DFA.

There are algorithms learning regular string languages from MQs and EQs (e.g., LSTAR [6]), and from MQs and positive samples (e.g., [9]). Moreover, the scope of interest has been extended from strings to trees – [10] and [11] present algorithms learning regular tree languages from MQs and EQs, and an algorithm learning regular tree languages from MQs and positive samples is given in [5]. The outputs of all algorithms mentioned so far are deterministic finite-state automata. However, in [2] Denis et al. also present an algorithm learning regular string languages from given data that returns an unambiguous RFSA, and eventually Bollig et al. [3] describe an LSTAR-like algorithm learning from MQs and EQs that in case of success returns the state-minimal RFSA mentioned above.

The notion of RFSA can be equally extended to trees: *Residual finite-state tree automata* (RFTA) have been defined and studied in [4]. As in the string case, for every regular tree language L the state-minimal transition-maximal RFTA $\mathcal{R}_L$ is unique and can be exponentially more succinct than the corresponding deterministic tree automaton (DFTA). Hence, this makes them particularly attractive for areas that use trees as their main structures such as information retrieval or computational linguistics. To complete the picture, we present a learning algorithm that infers a regular tree language from MQs and a finite positive sample, and returns $\mathcal{R}_L$ if the data is sufficient. The algorithm RESI is of a polynomial complexity due to a technique based on the one used in [5] for DFTA which we have adapted to and verified for the more intricate residual case.

Section 2 contains definitions and notions related to regular tree languages and residual languages, and some useful tools and properties for algorithmical learning already well-established in the literature. In Section 3 we first adapt some further concepts that were established in [3] for the specific inference of

RFSA to trees, then we describe our algorithm, prove a range of lemmata to show its correctness, and discuss its complexity. We conclude in Section 4.

2 Preliminaries

2.1 Regular tree languages and residual languages

We presume a basic knowledge of trees and the associated terminology. For a more extensive introduction into term-based trees also see [12].

A *ranked alphabet* Σ is a finite set of symbols in which each symbol is associated with a rank $n \in \mathbb{N}$. We write Σ_n to denote the set of all symbols associated with a specific rank $n \geq 0$. Let us fix Σ for the rest of this paper.

Definition 1. *The* set $\mathbb{T}_\Sigma$ of all trees over Σ *is inductively defined as the smallest set of expressions such that* $t = f(t_1, \ldots, t_n) \in \mathbb{T}_\Sigma$ *for all* $n \geq 0$, $f \in \Sigma_n$, *and* $t_1, \ldots, t_n \in \mathbb{T}_\Sigma$. *For* $n = 0$ *we simply write* t *as* f.
We call $t_1, \ldots, t_n$ *the* direct subtrees *of* t, *and we define the* set of all subtrees of t *as* $Subt(t) := \{t\}$ *for* $n = 0$ *and as* $Subt(t) := \{t\} \cup Subt(t_1) \cup \ldots \cup Subt(t_n)$ *otherwise, and we define* $Subt(L) := \{t \mid \exists t' \in L : t \in Subt(t')\}$ *for a set* $L \subseteq \mathbb{T}_\Sigma$. *A node labeled by a symbol from* Σ_0 *is called a* leaf. *The* size $|t|$ *of* t *is the number of nodes in* t. *Any set* $L \subseteq \mathbb{T}_\Sigma$ *represents a* tree language.

A tree can be decomposed into a subtree and the remaining part, a *context*:

Definition 2. *Let* $\Box \notin \Sigma$ *be a special symbol of rank 0 (a leaf label). A tree* $e \in \mathbb{T}_{\Sigma\cup\{\Box\}}$ *in which* $\Box$ *occurs exactly once is called a* context, *and the set of all contexts in* $\mathbb{T}_{\Sigma\cup\{\Box\}}$ *is denoted by* $\mathbb{C}_\Sigma$. *For* $e \in \mathbb{C}_\Sigma$ *and* $t \in \mathbb{T}_\Sigma \cup \mathbb{C}_\Sigma$, $e[\![t]\!]$ *denotes the tree obtained by substituting* t *for the leaf labeled by* $\Box$ *in* e. *The depth* $cdp(e)$ *of a context* e *is the length of the path from the leaf labeled by* $\Box$ *to the root. For* $L \subseteq \mathbb{T}_\Sigma$, *we define* $Cont(L) := \{e \in \mathbb{C}_\Sigma \mid \exists t \in Subt(L) : e[\![t]\!] \in L\}$.

Definition 3. *A* finite-state tree automaton (FTA) *is a tuple* $\mathcal{A} = \langle \Sigma, Q, F, \delta \rangle$ *where* Q *is the finite set of states,* $F \subseteq Q$ *is the set of accepting states, and* δ *is the transition relation defined by a set of mappings of the form* $\langle f, q_1 \cdots q_n \rangle \mapsto q$ *for* $n \geq 0$, $f \in \Sigma_n$, *and* $q_1, \ldots, q_n, q \in Q$ *where* $q_1 \cdots q_n$ *stands short for the sequence* $\langle q_1, \ldots, q_n \rangle$ *which for* $n = 0$ *is written as* $\langle\rangle$. *We also use* δ *to denote a function such that* $\delta(\langle f, q_1 \cdots q_n \rangle) = \{q \in Q \mid \exists \langle f, q_1 \cdots q_n \rangle \mapsto q \in \delta\}$.
From δ *we can derive a function* $\delta^* : \mathbb{T}_\Sigma \longrightarrow 2^Q$ *such that* $\delta^*(f(t_1, \ldots, t_n)) = \{q \in Q \mid \exists \langle f, q_1 \cdots q_n \rangle \mapsto q \in \delta : \forall i \in \{1, \ldots, n\} : q_i \in \delta^*(t_i)\}$,[1]
and a function $\delta^+ : Q \times \mathbb{C}_\Sigma \longrightarrow 2^Q$ *such that* $\delta^+(q, \Box) = \{q\}$ *and for* $e \neq \Box$,

$$\begin{aligned}\delta^+(q,e) = \{q' \in Q \mid \exists e', e'' \in \mathbb{C}_\Sigma : e = e'[\![e'']\!] \wedge \exists q'' \in Q : q' \in \delta^+(q'', e') \wedge \\ \exists n \geq 1 : \exists t_1, \ldots, t_n \in \mathbb{T}_\Sigma \cup \{\Box\} : \exists f \in \Sigma_n : \\ e'' = f(t_1, \ldots, t_n) \wedge \exists i \in \{1, \ldots, n\} : t_i = \Box \wedge \\ \exists \langle f, q_1 \cdots q_n \rangle \mapsto q'' \in \delta : q_i = q \wedge \forall j \in \{1, \ldots, n\} \setminus \{i\} : q_j \in \delta^*(t_j)\}.\end{aligned}$$

Intuitively, $\delta^*(t)$ *is the set of all states that the tree* t *ends up in and* $\delta^*(q, e)$ *is*

[1] We assume $\{1, \ldots, n\} = \emptyset$ for $n = 0$.

the set of all states that can be reached from state q by the context e.
For $q \in Q$, let $\mathcal{L}_q := \{t \in \mathbb{T}_\Sigma \mid q \in \delta^*(t)\}$ and $\mathcal{C}_q := \{e \in \mathbb{C}_\Sigma \mid \delta^+(q,e) \cap F \neq \emptyset\}$. Intuitively, $\mathcal{L}_q$ is the set of all trees that can end up in q and $\mathcal{C}_q$ is the set of all contexts that can lead from q into an accepting state.
We write $\mathcal{A}(t) = 1$ for $t \in \mathbb{T}_\Sigma$ if $\delta^*(t) \cap F \neq \emptyset$, $\mathcal{A}(t) = 0$ if $\delta^*(t) \cap Q \neq \emptyset$ but $\delta^*(t) \cap F = \emptyset$, and $\mathcal{A}(t) = *$ if $\delta^*(t) = \emptyset$.
The language accepted by $\mathcal{A}$ is defined as $\mathcal{L}(\mathcal{A}) := \{t \in \mathbb{T}_\Sigma \mid \delta^*(t) \cap F \neq \emptyset\}$.
Any tree language accepted by an FTA is called recognizable *or* regular*.*

If for no $\langle f, q_1 \cdots q_n\rangle \mapsto q \in \delta$ there is $\langle f, q_1 \cdots q_n\rangle \mapsto q' \in \delta$ with $q' \neq q$ then $\mathcal{A}$ is deterministic (a DFTA)*. We may write $\delta^*(t) = q$ for $\delta^*(t) = \{q\}$.*
If $\mathcal{A}$ is a DFTA and for all $n \geq 0$ with $\Sigma_n \neq \emptyset$ and all $f \in \Sigma_n$, $q_1, \ldots, q_n \in Q$ there is $\langle f, q_1 \cdots q_n\rangle \mapsto q \in \delta$ then $\mathcal{A}$ is total*.*

The equivalence relation $\equiv_L$ for $L \subseteq \mathbb{T}_\Sigma$ is defined such that $s \equiv_L t$ for $s, t \in \mathbb{T}_\Sigma$ iff $e[\![s]\!] \in L \Leftrightarrow e[\![t]\!] \in L$ for all $e \in \mathbb{C}_\Sigma$. The *index* I_L of L is the cardinality of the set $\{[t]_L \mid t \in \mathbb{T}_\Sigma\}$ where $[t]_L$ denotes the equivalence class containing t. The Myhill-Nerode theorem (see [8] for strings, [12] for trees) states that I_L is finite iff L is recognizable by a finite-state automaton. As a consequence, for every regular tree language L there is a total DFTA $\mathcal{A}_L$ with I_L states and each state recognizes a different equivalence class under $\equiv_L$, and $\mathcal{A}_L$ is the unique state-minimal total DFTA recognizing L up to isomorphism. If $\mathbb{T}_\Sigma \setminus Subt(L) \neq \emptyset$ then there is a non-total DFTA for L with one less state (i.e., it lacks the failure state for non-subtrees of L). In cases where it matters, we denote $\mathcal{A}_L$ by $\mathcal{A}_L^\bullet$ and the not necessarily total minimal DFTA without a failure state for L by $\mathcal{A}_L^\circ$.

The following three definitions are based on [4].

Definition 4. *The* (bottom-up) residual language $t^{-1}L$ *of a tree $t \in \mathbb{T}_\Sigma$ with respect to $L \subseteq \mathbb{T}_\Sigma$ is defined as the set $\{e \in \mathbb{C}_\Sigma \mid e[\![t]\!] \in L\}$.*

There is a natural correspondence between the equivalence classes and the residual languages of L expressed by $s^{-1}L = t^{-1}L$ iff $s \equiv_L t$ for any $s, t \in \mathbb{T}_\Sigma$. As a consequence, every tree language L has I_L residual languages, and any pair of equivalence classes of L can be distinguished by their differing – but not necessarily disjoint – sets of contexts.

Definition 5. *A* (bottom-up) residual finite-state tree automaton (RFTA) *is an FTA $\mathcal{R} = \langle \Sigma, Q, F, \delta\rangle$ satisfying $\forall q \in Q : \exists t \in \mathbb{T}_\Sigma : \mathcal{C}_q = t^{-1}\mathcal{L}(\mathcal{R})$.*

The class of languages recognized by bottom-up RFTA corresponds to the class of regular tree languages as a whole (see [4]).

Definition 6. *Let $L \subseteq \mathbb{T}_\Sigma$. A residual language $t^{-1}L$ of L is* composite *iff*

$$t^{-1}L = \bigcup_{s^{-1}L \subsetneq t^{-1}L} s^{-1}L.$$

Otherwise we say that it is prime*. The set of prime residual languages of L is denoted by $\mathcal{P}(L)$. We also define $\mathcal{P}_t(L) := \{x \in \mathcal{P}(L) \mid x \subseteq t^{-1}L\}$ for $t \in \mathbb{T}_\Sigma$.*

We remark that if the empty union is defined as the empty set then by the definition above the empty set must be classified as composed which also implies that prime residual languages are intrinsically non-empty.
For a regular $L \subseteq \mathbb{T}_\Sigma$, one can define an RFTA $\mathcal{R}_L = \langle \Sigma, Q_L, F_L, \delta_L \rangle$ by

- $Q_L := \mathcal{P}(L)$,
- $F_L := \{q \in Q_L \mid \Box \in q\}$, and
- $\delta_L := \{\langle f, q_1 \cdots q_n \rangle \mapsto q \mid f \in \Sigma_n \wedge \exists t_1, \ldots, t_n \in \mathbb{T}_\Sigma : \forall i \in \{1, \ldots, n\} :$ $q_i = {t_i}^{-1}L \wedge q \subseteq f(t_1, \ldots, t_n)^{-1}L\}$.

For each $q \in Q_L$ we have $\mathcal{C}_q = q$. Moreover, $\mathcal{R}_L$ is transition-maximal, or *saturated*, i.e., no transition can be added to δ_L without increasing the language that is recognized. Up to isomorphism $\mathcal{R}_L$ is the unique state-minimal saturated RFTA recognizing L (see [4]).[2]

2.2 Learning tools

The type of learner we consider in this paper tries to infer an FTA for an unknown regular tree language $L \subseteq \mathbb{T}_\Sigma$ from given information. It solves this task by means of an *observation table* in which it keeps track of the information it has obtained and processed so far (also see [7, 6]). The rows of the table are labeled by trees (set S), the columns by contexts (set E). Let us fix $L \subseteq \mathbb{T}_\Sigma$.

Definition 7. *A triple $T = \langle S, E, obs \rangle$ with two finite non-empty sets $S \subseteq \mathbb{T}_\Sigma$, $E \subseteq \mathbb{C}_\Sigma$, and $\Box \in E$ is called an* observation table *if*

- *S is* subtree-closed*, i.e.,* *$f(s_1, \ldots, s_n) \in S$ implies $s_1, \ldots, s_n \in S$ for all $f(s_1, \ldots, s_n) \in \mathbb{T}_\Sigma$, and*
- *$obs : \mathbb{T}_\Sigma \times \mathbb{C}_\Sigma \longrightarrow \{0, 1\}$ is a function such that* *$obs(s, e) = 1$ if $e[\![s]\!] \in L$ and $obs(s, e) = 0$ if $e[\![s]\!] \notin L$.*

The row *of $s \in \mathbb{T}_\Sigma$ is $row(s) := \{\langle e, obs(s, e) \rangle \mid e \in E\}$, and $row(X) := \{row(s) \mid s \in X\}$ for any set $X \subseteq \mathbb{T}_\Sigma$, and $row(s)(e) := obs(s, e)$ for $s \in \mathbb{T}_\Sigma$ and $e \in E$.*[3] *We say that two trees $s_1, s_2 \in \mathbb{T}_\Sigma$ are* obviously different *and write $s_1 <> s_2$ if $row(s_1) \neq row(s_2)$. For $row(s_1) = row(s_2)$ we write $s_1 \approx s_2$.*

Definition 8. *An observation table $T = \langle S, E, obs \rangle$ is said to be* consistent *iff, for all $f(s_1, \ldots, s_n), f(s'_1, \ldots, s'_n) \in S$,*
$$s_i \approx s'_i \text{ for all } i \text{ with } 1 \leq i \leq n \text{ implies } f(s_1, \ldots, s_n) \approx f(s'_1, \ldots, s'_n).$$

[2] We remark that the definition of prime residual languages given by Bollig et al. [3] implies that $\emptyset$ must be classified as prime. As a consequence, their canonical RFSA differs from the one defined by the authors they cite, Denis et al. [1], in that it can have one more state q with $\mathcal{C}_q = \emptyset$, which will be assigned to all input strings.

[3] A note in comparison to the definition usually used in the literature: We define *obs* for the entire domain of $\mathbb{T}_\Sigma \times \mathbb{C}_\Sigma$, and *row* for $\mathbb{T}_\Sigma$ instead of restricting them to $S \times E$ and S, respectively, in order to express that while any concrete table built by the learner is finite theoretically a learner can always establish the row for any given tree with the help of the oracle. However, the sets in the range of *row* must be of finite and equal cardinality (namely, $|E|$) to make rows comparable in linear time.

From a table $T = \langle S, E, obs \rangle$ we can derive an FTA $\mathcal{A}_T = \langle \Sigma, Q_T, F_T, \delta_T \rangle$ with:

- $Q_T := row(S)$,
- $F_T := \{q \in row(S) \mid q(\Box) = 1\}$, and
- $\delta_T := \{\langle f, \langle q_1, \ldots, q_n \rangle\rangle \mapsto q \mid f \in \Sigma_n \wedge \exists s_1, \ldots, s_n, s \in S :$ $s = f(s_1, \ldots, s_n) \wedge \forall i \in \{1, \ldots, n\} : q_i = row(s_i) \wedge q = row(s)\}$.

Note that as S is subtree-closed $\neg \exists q' \in Q_T : \mathcal{L}_{q'} = \emptyset$, i.e., all states can be reached. If T is consistent then $\mathcal{A}_T$ is deterministic and isomorphic to $\mathcal{A}^{\bullet}_{\mathcal{L}(\mathcal{A}_T)}$ (see [11]) or to $\mathcal{A}^{\circ}_{\mathcal{L}(\mathcal{A}_T)}$ (see [5]), depending on the properties of the set S.[4]

In particular, a sample of L from which a learner can derive its set S of candidates to represent different states can have the following important property:

Definition 9 ([5]). *A finite set $X_+ \subseteq L$ is* representative *for a tree language $L \subseteq \mathbb{T}_\Sigma$ with $\mathcal{A}^{\circ}_L = \langle \Sigma, Q, F, \delta \rangle$ iff for every transition $\langle f, q_1 \cdots q_n \rangle \mapsto q \in \delta$ there is $f(s_1, \ldots, s_n) \in Subt(X_+)$ with $q_i = \delta^*(s_i)$ for all i with $1 \leq i \leq n$.*

Intuitively, the set X_+ is representative for L if in order to parse the entirety of the elements in X_+ every transition of $\mathcal{A}_L$ has to be used at least once.

Definition 10. *A finite set $X_+ \subseteq L$ is* R-representative *for a regular tree language $L \subseteq \mathbb{T}_\Sigma$ if it is representative for L and for each $x \in \mathcal{P}(L)$ there is $e \in Cont(X_+)$ such that $e \in x$ but $e \notin \bigcup_{y \in \mathcal{P}(L) \wedge y \subsetneq x} y$.*

Intuitively, X_+ is R-representative for L if for every prime residual language x of L there is a context $e \in Cont(X_+)$ proving that x is indeed prime.

In the next section we will establish a more concrete notational base for the specific inference of RFTA, and then present and analyze our learning algorithm.

3 Learning residual finite-state tree automata

3.1 Observation tables and RFTA

We need some more tools and notions to relate observation tables and RFTA. For the following definitions (based on [3]), let us fix a table $T = \langle S, E, obs \rangle$.

Definition 11. *For $r_1, r_2 \in row(\mathbb{T}_\Sigma)$ and $R \subseteq row(\mathbb{T}_\Sigma)$, we define*
$r_1 \sqcup r_2 := \{\langle e, x \rangle \mid e \in E \wedge x \in \{0,1\} \wedge (r_1(e) = 1 \vee r_2(e) = 1 \Leftrightarrow x = 1)\}$, *and*
$\bigsqcup R := \{\langle e, x \rangle \mid e \in E \wedge x \in \{0,1\} \wedge (\exists r \in R : r(e) = 1 \Leftrightarrow x = 1)\}$.
For two rows $r, r' \in row(\mathbb{T}_\Sigma)$ we say that r is covered *by r', and denote it by $r \sqsubseteq r'$, if $r(e) = 1$ implies $r'(e) = 1$ for all $e \in E$.*
A row $r \in row(S)$ is said to be composed *if there are $r_1, \ldots, r_n \in row(S) \setminus \{r\}$ such that $r = r_1 \sqcup \ldots \sqcup r_n$. If r is not composed and there is at least one $e \in E$*

[4] For readers of [6, 11, 3]: In this setting we do not need the notion of closedness since the set S does not grow over time. Our definition of consistency is based on [5].

with $r(e) = 1$ *then* r *is said to be* prime.[5] *We define* $\mathcal{P}_S := \{row(s) \mid s \in S \wedge row(s) \text{ is prime}\}$ *and* $\mathcal{P}_s := row(\{s' \in S \mid s' \sqsubseteq s\}) \cap \mathcal{P}_S$ *for* $s \in \mathbb{T}_\Sigma$.

Definition 12. *The table* T *is* R-consistent *iff*
for all $f(s_1, \ldots, s_n), f(s'_1, \ldots, s'_n) \in S$,

$$s_i \sqsubseteq s'_i \text{ for all } i \text{ with } 1 \leq i \leq n \text{ implies } f(s_1, \ldots, s_n) \sqsubseteq f(s'_1, \ldots, s'_n).$$ [6]

Let us consider a small example to clarify the two definitions given above.

Example 1. In the small observation table given in Figure 1,

- the row of $f(a, b)$ is the only one that is not prime because it can be composed from the rows of a and of $f(a, a)$ or $f(b, b)$,
- the row of b covers all others, the row of $f(a, b)$ covers those of a, $f(a, a)$ and $f(b, b)$, and the rows of $f(a, a)$ and $f(b, b)$ cover each other.

Moreover, this table is not R-consistent because the row of b covers the row of a but the row of $f(b, b)$ does not cover the row of $f(a, b)$.

	$\Box$	$f(\Box, b)$	$f(a, \Box)$
a	0	1	0
b	1	1	1
$f(a, a)$	1	0	0
$f(a, b)$	1	1	0
$f(b, b)$	1	0	0

Fig. 1. An example for an observation table and relations in it

If T is R-consistent then
we can derive an FTA $\mathcal{R}_T = \langle \Sigma, Q_T, F_T, \delta_T \rangle$ from T defined by:

- $Q_T := \mathcal{P}_S$,
- $F_T := \{r \in \mathcal{P}_S \mid r(\Box) = 1\}$, and
- $\delta_T := \{\langle f, q_1 \cdots q_n \rangle \mapsto q \mid q_1, \ldots, q_n, q \in \mathcal{P}_S \wedge \exists s_1, \ldots, s_n, f(s_1, \ldots, s_n) \in S : \forall i \in \{1, \ldots, n\} : q_i = row(s_i) \wedge q \sqsubseteq row(f(s_1, \ldots, s_n))\}$.

As S is subtree-closed all states are reachable, i.e., for all $q \in Q_T$ we have $\mathcal{L}_q \neq \emptyset$. Note that R-consistency is needed to ensure that δ_T is well-defined (compare the explanation for the string case in [3]): Consider $s_1, \ldots, s_n, s'_1, \ldots, s'_n \in \mathcal{P}_S$ with $s_i \approx s'_i$ for $1 \leq i \leq n$. Obviously, $s_i \sqsubseteq s'_i$ and $s'_i \sqsubseteq s_i$. R-consistency of T implies that if there are $s, s' \in S$ with $s = f(s_1, \ldots, s_n)$ and $s' = f(s'_1, \ldots, s'_n)$ then we have $s \sqsubseteq s'$ and $s' \sqsubseteq s$ and hence $s \approx s'$ as well such that $row(s)$ and $row(s')$ cover the same (prime) rows, i.e., states in Q_T.

[5] We require the existence of e in addition to the original definition by Bollig et al. [3] because in accordance with the assumption that prime residual languages are intrinsically non-empty we do not want to classify a row containing only 0s as prime.

[6] Footnote 4 applies for the notions of R-closedness and R-consistency analogously.

3.2 The algorithm RESI

We fix a regular tree language $L \subseteq \mathbb{T}_\Sigma$ as the target to be inferred and an R-representative sample X_+ of L for the rest of this subsection. Our learner is given X_+ as an input. In addition, it has access to a *membership oracle* which can be consulted with queries of the form '$t \in L$?' for some $t \in \mathbb{T}_\Sigma$ and which we will represent by a blackbox automaton $\mathcal{O}$ that correctly recognizes L. The pseudo-code of the algorithm RESI is given in Figure 2. For the proper definition of T inside RESI we let the function *obs* be restricted to the domain $S \times E$.

Input: A positive sample X_+ of L, a membership oracle $\mathcal{O}$.
Output: An FTA.

$S := \mathit{Subt}(X_+)$; $E := \mathit{Cont}(X_+)$;
$T := \langle S, E, \mathit{obs} \rangle$;
while there are $f(s_1, \ldots, s_n), f(s'_1, \ldots, s'_n) \in S$ with
$s_i \sqsubseteq s'_i$ for $1 \leq i \leq n$ but $\neg(f(s_1, \ldots, s_n) \sqsubseteq f(s'_1, \ldots, s'_n))$ **do**
  find $e \in E$ with $\mathcal{O}(e[\![f(s_1, \ldots, s_n)]\!]) = 1$ and $\mathcal{O}(e[\![f(s'_1, \ldots, s'_n)]\!]) = 0$;
  for $i = 1, \ldots, n$ **do**
    if $\mathcal{O}(e[\![f(t'_1, \ldots, t'_n)]\!]) = 0$ for $t'_i = s'_i$ and
$\forall j \in \{1, \ldots, n\} \setminus \{i\} : t'_j = s_j$ **then**
      $E := E \cup \{f(t_1, \ldots, t_n) \mid t_i = \square \ \wedge \ \forall j \in \{1, \ldots, n\} \setminus \{i\} : t_j = s_j\}$;
    end if;
  end for;
  $T := \langle S, E, \mathit{obs} \rangle$;
end while;
return $\mathcal{R}_T$.

Fig. 2. Algorithm RESI

The following lemma is essential for the learner's progress because it states that as long as the learning process is not concluded it is possible to derive a separating context for two individual candidates directly from the table itself. It is based on a corresponding one from [5] for the inference of a DFTA in the same setting but adapted to and proven for the more intricate residual case.

Lemma 1. *As long as T is not R-consistent, there are $f(s_1, \ldots, s_n), f(s'_1, \ldots, s'_n) \in S$ with $s_i \sqsubseteq s'_i$ for all i with $1 \leq i \leq n$ and $\neg(f(s_1, \ldots, s_n) \sqsubseteq f(s'_1, \ldots, s'_n))$ such that there is an index $j \in \{1, \ldots, n\}$ with $s_j^{-1}L \not\subseteq s_j'^{-1}L$ but $s_k^{-1}L \subseteq s_k'^{-1}L$ for all other $k \in \{1, \ldots, n\} \setminus \{j\}$.*

Proof. We prove this by a contradiction. Assume $e \in \mathbb{C}_\Sigma$ to be a context such that $e[\![s]\!] \in L$ and $e[\![s']\!] \notin L$ for some $s, s' \in S$ with $s \sqsubseteq s'$ but $s^{-1}L \not\subseteq s'^{-1}L$ (which must exist due to the R-inconsistency of T) and the depth $\mathit{cdp}(e)$ to be minimal. The fact that e cannot be in E but $\square$ is implies $e \neq \square$ and thus there are $e_1, e_2 \in \mathbb{C}_\Sigma$ such that $e_1[\![e_2]\!] = e$ and $e_2 = f(s_1, \ldots, s_n)$ with $s_j = \square$ for some

$j \in \{1, \ldots, n\}$ and $n \geq 1$. Clearly, $cdp(e_1) < cdp(e)$. Since we have required the depth of e to be minimal e_1 cannot fulfil the role of a separating context for any pair of trees $t, t' \in S$ with $t \sqsubseteq t'$ but $t^{-1}L \not\subseteq t'^{-1}L$.

As $e[\![s]\!] = e_1[\![e_2[\![s]\!]]\!] \in L$ we know that $e_2[\![s]\!] \in Subt(L)$, and as X_+ is representative for L there is $f(t_1, \ldots, t_n) \in S$ with $t_i \equiv_L s_i$ for all $i \in \{1, \ldots, n\} \setminus \{j\}$ and $t_j \equiv_L s$ and $f(t_1, \ldots, t_n) \equiv_L e_2[\![s]\!]$. Since $f(t_1, \ldots, t_n) \in Subt(X_+)$ there is a context $e_3 \in Cont(X_+) \subseteq E$ such that $e_3[\![f(t_1, \ldots, t_n)]\!] \in X_+ \subseteq L$. Moreover, the context $e_3[\![e_4]\!]$ where e_4 is obtained by replacing the tree t_j in $f(t_1, \ldots, t_n)$ by $\Box$ is also in $Cont(X_+)$ and in E. Obviously, we have $e_3[\![e_4[\![s]\!]]\!] \in L$ and $obs(s, e_3[\![e_4]\!]) = 1$. The precondition $s \sqsubseteq s'$ implies that $obs(s', e_3[\![e_4]\!]) = 1$ as well and $e_3[\![e_4[\![s']\!]]\!] \in L$. Note that $e_4[\![s']\!] \equiv_L e_2[\![s']\!]$ and hence $e_2[\![s']\!] \in Subt(L)$.

The fact that e is separating for s and s' entails $e_2[\![s]\!]^{-1}L \not\subseteq e_2[\![s']\!]^{-1}L$. Again, as X_+ is representative for L there is $f(t'_1, \ldots, t'_n) \in S$ with $t'_i \equiv_L t_i$ for all $i \in \{1, \ldots, n\} \setminus \{j\}$ and $t'_j \equiv_L s'$ and $f(t'_1, \ldots, t'_n) \equiv_L e_2[\![s']\!]$. In contradiction to the claim, let us suppose $f(t_1, \ldots, t_n) \sqsubseteq f(t'_1, \ldots, t'_n)$. In that case e_1 would fulfil the role of a separating context as well since we have $e_1[\![f(t_1, \ldots, t_n)]\!] \in L$ but $e_1[\![f(t'_1, \ldots, t'_n)]\!] \notin L$. However, this would violate the assumption that the depth of e be minimal and thus the claim is proven. ■

RESI exploits Lemma 1 in lines 3–9. RESI searches for an R-inconsistency, a separating context $e \in E$, and then for all pairs $s, s' \in S$ as specified in Lemma 1 by successively substituting s'_i for s_i in $e[\![f(s_1, \ldots, s_n)]\!]$. If the R-inconsistency persists then RESI adds the corresponding separating context for s_i and s'_i to E. As a consequence, as soon as the while-loop is exited the final table T is R-consistent and it is possible to derive $\mathcal{R}_T$ from T as specified above.

3.3 Proofs for RESI

It is clear that RESI terminates since in each execution of the while-loop at least one row inclusion is eliminated (also consider the proof of Lemma 1), and no new inclusion between rows can be introduced by adding elements to E.
Let $T = \langle S, E, obs \rangle$ be the final table. We abbreviate $\mathcal{L}(\mathcal{R}_T)$ as L_R.

Theorem 1. *$\mathcal{R}_T$ is a state-minimal saturated RFTA, and $L_R = L$.*

We show this by establishing and proving a range of lemmata that explore the various interrelations between the final table T and the residual languages of L. Obviously, T is R-consistent. Moreover, consider the following property:

Definition 13. *We say that T is* exclusive *for L if for any $s_1, s_2 \in S$ with ${s_1}^{-1}L \not\subseteq {s_2}^{-1}L$ we have $\neg(s_1 \sqsubseteq s_2)$.*

When RESI terminates T is exclusive for L. This follows directly from the fact that since X_+ is representative for L all possible transitions between the residual languages of L are represented in S and from the way RESI resolves R-inconsistencies which entails that for any $s_1, s_2 \in S$ with ${s_1}^{-1}L \not\subseteq {s_2}^{-1}L$ there is a context $e \in E$ with $obs(s_1, e) = 1$ but $obs(s_2, e) = 0$. For ${s_1}^{-1}L \neq \{\Box\}$ and ${s_2}^{-1}L \neq \{\Box\}$ this can be deduced from the proof of Lemma 1. In the special case of $s^{-1}L = \{\Box\}$ for some $s \in S$, it suffices to observe that $obs(s, e') = 0$ for all contexts $e' \in E \setminus \{\Box\}$ and for all $s' \in S$ with $s'^{-1}L \not\subseteq s^{-1}L$ there is a context $e'' \in E \setminus \{\Box\}$ with $obs(s', e'') = 1$ due to $Cont(X_+) \subseteq E$, and that on the other hand for each $s'' \in S$ with $s^{-1}L \not\subseteq s''^{-1}L$ we must have $row(s'', \Box) = 0$.

The fact that T is exclusive for L entails that if $s_1 \sqsubseteq s_2$ for $s_1, s_2 \in S$ then we can conclude ${s_1}^{-1}L \subseteq {s_2}^{-1}L$. Hence, the inclusion relations among rows in T exactly reflect those among corresponding residual languages of L. Furthermore:

Lemma 2. *For all $s \in S$ we have $s^{-1}L \in \mathcal{P}(L) \Leftrightarrow row(s) \in \mathcal{P}_S$.*

Proof. "$\Rightarrow$": Let $s^{-1}L \in \mathcal{P}(L)$. As X_+ is representative for L, for all $t^{-1}L \in \mathcal{P}(L)$ there is $t' \in S$ with $t' \equiv_L t$ and $t' \approx t$. Obviously, $t^{-1}L \subseteq s^{-1}L$ implies $t' \sqsubseteq s$ and $t^{-1}L \subsetneq s^{-1}L$ implies $s^{-1}L \not\subseteq t^{-1}L$. In that case, as X_+ is also R-representative for L we have $\neg(s \sqsubseteq t')$ as well, i.e., there is $e \in E$ with $row(s)(e) = 1$ but $row(t')(e) = 0$. As a consequence, $row(s)$ must be prime in T.
"$\Leftarrow$": Let $row(s) \in \mathcal{P}_S$. As T is exclusive for L, for all $s' \in S$, if $s' \sqsubseteq s$ then this implies $s'^{-1}L \subseteq s^{-1}L$. If $s' \sqsubseteq s$ but $s' <> s$ then this means that there is a context $e \in E$ with $obs(s, e) = 1$ but $obs(s', e) = 0$ and hence $s'^{-1}L \subsetneq s^{-1}L$. As X_+ is representative for L there is no $s''^{-1}L \in \mathcal{P}(L)$ with $s''^{-1}L \subseteq s^{-1}L$ but $row(s'') \notin row(S)$ and hence the residual language $s^{-1}L$ must be prime. ∎

We will say that T is *faithful* for L to express the fact that S is R-representative and T exclusive for L, along with the consequences stated above, i.e., that all prime residual languages of L and all transitions and inclusion relations between them are mirrored by components of T. Recall that L has a unique state-minimal saturated RFTA denoted by $\mathcal{R}_L = \langle \Sigma, Q_L, F_L, \delta_L \rangle$, and note the exact parallelism between the definitions of δ_L and δ_T. In order to show the relation between $\mathcal{R}_T$ and $\mathcal{R}_L$ we shall also prove some lemmata about $\mathcal{R}_L$.

Lemma 3. *We have $x \subseteq t^{-1}L$ for all $t \in \mathbb{T}_\Sigma$ and all $x \in \delta_L^*(t)$.*

Proof. By induction over the depth of t. If $t = a$ for some $a \in \Sigma_0$ then the claim follows directly from the definition of δ_L. Let $t = f(t_1, \ldots, t_n)$. Then $\delta_L^*(t) = \{q \in Q_L \mid \exists \langle f, q_1 \cdots q_n \rangle \mapsto q \in \delta_L : \forall i \in \{1, \ldots, n\} : q_i \in \delta_L^*(t_i)\}$. Let $q \in \delta_L^*(t)$ and assume the claim to hold for $t_1, \ldots, t_n$. By the definition of δ_L there are $t'_1, \ldots, t'_n \in \mathbb{T}_\Sigma$ and $q_i \in \delta_L^*(t_i)$ such that $q_i = t_i'^{-1}L$ for $1 \leq i \leq n$ and $q \subseteq f(t'_1, \ldots, t'_n)^{-1}L$. By the induction assumption we have $q_i \subseteq {t_i}^{-1}L$. This yields $t_i'^{-1}L \subseteq {t_i}^{-1}L$ and $f(t'_1, \ldots, t'_n)^{-1}L \subseteq f(t_1, \ldots, t_n)^{-1}L$ and hence $q \subseteq t^{-1}L$. ∎

Corollary 1. *We have $q \sqsubseteq row(s)$ for all $s \in Subt(L)$ and all $q \in \delta_T^*(s)$.*[7]

Proof. By Lemma 3 and the fact that T is R-consistent and faithful for L. ∎

Lemma 4. *For all $q \in Q_L$ and all $t \in \mathbb{T}_\Sigma$ with $t^{-1}L = q$ we have $q \in \delta_L^*(t)$.*

Proof. As q is prime there are $e \in q$ such that $e \notin q'$ for all q' with $q' \subsetneq q$. Let $t \in \mathbb{T}_\Sigma$ with $t^{-1}L = q$ and assume $q \notin \delta_L^*(t)$. By Lemma 3 we have $q'' \subseteq q$ for all $q'' \in \delta_L^*(t)$. By the argument above no $q'' \in \delta_L^*(t)$ can contain e. However, since $q'' = \mathcal{C}_{q''}$ this would imply that $\mathcal{R}_L$ does not accept $e[\![t]\!]$, a contradiction. ∎

[7] Corollary 1 and the proof of Lemma 3 are inspired by a related one in [3] for strings.

Corollary 2. *For all $s \in Subt(L)$ with $row(s) \in Q_T$ we have $row(s) \in \delta_T^*(s)$.*

Proof. By Lemma 4 and the fact that T is faithful for L. ∎

Lemma 5. *For all $q \in Q_T$ with $q = row(s)$ for some $s \in S$, for all $e \in \mathbb{C}_\Sigma$, $e \in s^{-1}L \Leftrightarrow e \in \mathcal{C}_q$.*

Proof. This follows directly from the faithfulness of T as well. ∎

Lemma 6. *For all $s \in Subt(L)$ and all $e \in s^{-1}L$ there is $q \in \delta_T^*(s)$ with $e \in \mathcal{C}_q$.*

Proof. If $s = a$ for some $a \in \Sigma_0$ then $s \in S$ because S is representative for L, and $\delta_T^*(s) = \mathcal{P}_s$ by the definition of δ_T. By Lemma 2 we have $row(s') \in \mathcal{P}_s \Leftrightarrow s'^{-1}L \in \mathcal{P}_s(L)$ for all $s' \in S$, and the claim follows from Lemma 5.
Let $s = f(s_1, \ldots, s_n)$ and assume the claim to hold for $s_1, \ldots, s_n$. For $1 \leq i \leq n$, we define the context $e_i \in \mathbb{C}_\Sigma$ such that e_i is obtained by replacing s_i by $\square$ in $e[\![s]\!]$. By the induction assumption there is $q_i \in \delta_T^*(s_i)$ with $e_i \in \mathcal{C}_{q_i}$ for all i with $1 \leq i \leq n$. Since T is faithful for L and by Lemma 5 there is $q \in Q_T$ with $e \in \mathcal{C}_q$ and a transition $\langle f, q_1 \cdots q_n \rangle \mapsto q \in \delta_T$, and thus the claim is shown. ∎

Lemma 7. *For all $s \in Subt(L)$ we have $s \in L \Leftrightarrow \mathcal{R}_T(s) = 1$.*

Proof. "$\Rightarrow$": Let $s \in L$. Then the claim follows immediately from Lemma 6 since there must be $q \in \delta_T^*(s)$ with $\square \in \mathcal{C}_q$ and thus $q \in F_T$.
"$\Leftarrow$": If $\mathcal{R}_T(s) = 1$ then there is $q \in \delta_T^*(s)$ with $q(\square) = 1$ by the definition of F_T, and since $q \sqsubseteq row(s)$ by Corollary 1 this yields $obs(s, \square) = 1$ and $s \in L$. ∎

Lemma 8 establishes the connection between the inclusion relation of prime rows in the table and the sets of contexts defined by the corresponding states:

Lemma 8. *For all $q_1, q_2 \in Q_T$ we have $q_1 \sqsubseteq q_2 \Leftrightarrow \mathcal{C}_{q_1} \subseteq \mathcal{C}_{q_2}$.*

Proof: "$\Rightarrow$": Let $q_1 \sqsubseteq q_2$ and $e \in \mathcal{C}_{q_1}$. If $e = \square$ then $q_1(\square) = 1$ and, as $q_1 \sqsubseteq q_2$, $q_2(\square) = 1$ as well and thus $q_2 \in F_T$ and $\square \in \mathcal{C}_{q_2}$. Otherwise, since T is exclusive for L, $q_1 \sqsubseteq q_2$ implies ${s_1}^{-1}L \subseteq {s_2}^{-1}L$ for all $s_1, s_2 \in S$ with $row(s_1) = q_1$ and $row(s_2) = q_2$. This implies $e \in {s_2}^{-1}L$ and we obtain $e \in \mathcal{C}_{q_2}$ by Lemma 5.
"$\Leftarrow$": Assume $\neg(q_1 \sqsubseteq q_2)$. By the definition of $\sqsubseteq$ there exists $e \in E$ with $q_1(e) = 1$ but $q_2(e) = 0$. We have $e \in \mathcal{C}_{q_1}$ by Lemma 5. For all $s \in Subt(L)$ with $row(s) = q_2$ we have $q_2 \in \delta_T^*(s)$ by Corollary 2. Since $\mathcal{R}_T$ cannot accept $e[\![s]\!]$ due to Lemma 7 we can conclude that $e \notin \mathcal{C}_{q_2}$, and hence we obtain a proof for $\mathcal{C}_{q_1} \not\subseteq \mathcal{C}_{q_2}$. ∎

Unlike in [3] where Bollig et al. can rely on the last equivalence query which must have been answered in the positive by the oracle to establish the correctness of their learner's solution, in our setting $L_R = L$ has to be proven explicitly. The inclusion $L \subseteq L_R$ is a direct consequence of Lemma 7. To show that RESI does not overgeneralize, i.e., that the converse $L_R \subseteq L$ holds as well it remains to show that $\mathcal{R}_T$ does not accept any tree that is not a subtree of L.

Lemma 9. *For all $s \notin Subt(L)$ we have $\delta_T^*(s) = \emptyset$.*

Proof. Obviously, we have $row(s)(e) = 0$ for all $e \in E$.
If $s = a$ for some $a \in \Sigma_0$ then the claim follows directly from the definition of δ_T since there cannot be a prime row in $row(S)$ that is covered by $row(a)$.
Let $s = f(s_1, \ldots, s_n)$ and assume the claim to hold for $s_1, \ldots, s_n$. If there is s_j with $s_j \notin Subt(L)$ for some $j \in \{1, \ldots, n\}$ then we have $\delta_T^*(s_j) = \emptyset$ by the induction assumption, and we obtain $\delta_T^*(s) = \emptyset$ directly by the definition of δ_T^*.
Assume $s_1, \ldots, s_n \in Subt(L)$. Clearly for any $f(s'_1, \ldots, s'_n) \in Subt(L)$ there is at least one index $k \in \{1, \ldots, n\}$ such that $s_k'^{-1}L \not\subseteq s_k{}^{-1}L$ because otherwise we would have $f(s_1, \ldots, s_n) \in Subt(L)$. Moreover, since T is exclusive for L, if the tree $f(s'_1, \ldots, s'_n)$ is in S then there is an excluding context $e \in E$ with $obs(s'_k, e) = 1$ and $obs(s_k, e) = 0$, i.e., we have $\neg(s'_k \sqsubseteq s_k)$ in T.
Let $q_1, \ldots, q_n \in Q_T$ with $q_i \in \delta_T^*(s_i)$ for $1 \leq i \leq n$. Then there are elements $t_1, \ldots, t_n \in S$ with $q_i = row(t_i)$ for $1 \leq i \leq n$. However, by the argument above $t = f(t_1, \ldots, t_n)$ cannot be a subtree of L, which implies $row(t)(e) = 0$ for all $e \in E$ and $\mathcal{P}_t = \emptyset$, and hence there is no suitable prime row $q \in Q_T$ that can be assigned as a state to s. Thus, $\delta_T^*(s) = \emptyset$. ■

We can now conclude the proof of Theorem 1 as follows. The FTA $\mathcal{R}_T$ returned by RESI meets the definition of an RFTA, i.e., for each state q the set $\mathcal{C}_q$ is a residual language of L_R: Let $s \in S$ with $row(s) \in Q_T$ and $q = row(s)$. We show $\mathcal{C}_q = s^{-1}L_R$. We have $q \in \delta_T^*(s)$ due to Corollary 2. This implies $\mathcal{C}_q \subseteq s^{-1}L_R$. Conversely, for all $q' \in \delta_T^*(s)$ we have $q' \sqsubseteq q$ by Corollary 1. With Lemma 8 this implies $\mathcal{C}_{q'} \subseteq \mathcal{C}_q$ for all $q' \in \delta_T^*(s)$. This yields $s^{-1}L_R \subseteq \mathcal{C}_q$ and the equality. Lemma 2 implies that $s^{-1}L_R$ is prime and hence $\mathcal{R}_T$ is a state-minimal RFTA. The fact that $L_R = L$ follows by Lemmata 7 and 9. Finally, as X_+ is representative for L, $\mathcal{R}_T$ is saturated and thus $\mathcal{R}_T$ is isomorphic to $\mathcal{R}_L$.

3.4 Complexity

Naturally, the complexity of RESI crucially depends on the given sample X_+. Let $n_+ := \sum_{t \in X_+} |t|$ be the number of nodes of all trees in X_+ taken together, which is also the maximal cardinality of $Subt(X_+)$ and $Cont(X_+)$. In a worst case, every tree in $Subt(X_+)$ represents a different residual language of L. The initial table has at most n_+ rows and at most n_+ columns. In a worst case, we have to exclude the inclusion of the residual languages for each pair of trees $s_1, s_2 \in S$, and the number of while-loop executions needed to do so is bounded by n_+^2. Let ρ be the maximal rank in Σ. In each execution of the while-loop at most ρ contexts are added to the set of experiments E, and hence the total number of MQs that have to be asked in order to fill in the cells of the final table is bounded by $n_+ \cdot (n_+ + n_+^2\rho) = n_+^2 + n_+^3\rho$ or $O(n_+^3)$.

We observe that while the algorithm by Bollig et al. [3] learning from membership and equivalence queries can rely on the fact that it will receive a counterexample as long as the table does not represent the state-minimal saturated residual automaton for the target, i.e., it will eventually receive a counterexample containing an instance of exactly those contexts that mark a certain residual language as prime, the complexity of retrieving such a context in the present setting is not polynomial. As a first intuitive explanation, consider that in order to ensure that a certain row r in T representing a prime residual language of L is also prime two conditions have to be fulfilled: (a) r does not wrongly cover another prime row because otherwise r may appear composed, and (b1) not only for each row r' with $r' \sqsubseteq r$ there is a context making the covering relation strict but also (b2) there is a context making the covering relation between r and $\bigsqcup_{r' \sqsubseteq r} r'$ strict. Conditions (a) and (b1) are ensured by the fact that the final table constructed by RESI is T-exclusive but for (b2) we would have to try out all possible contexts up to depth I_L since this may be the minimal distance in $\mathcal{A}_L$ to an accepting state, which is an exponential procedure only bounded by $O(\rho^{I_L})$. Therefore we have chosen to introduce this information via the data a priori. The outlined argument is also the reason for the fact that in many cases the algorithm by Bollig et al. [3] identifies the target after much less queries than suggested by its theoretical complexity since while not all equivalence classes of L have to be represented in T in order to represent $\mathcal{R}_L$ the learner may even benefit from the absence of such a representative, and since their setting includes the very powerful device of equivalence queries the learner is notified immediately in case of success.

However, also note that while Bollig et al. [3] simply add all substrings of a received counterexample to E and a straightforward adaptation of their algorithm to trees would accordingly add all *subtrees* of a given counterexample to E, the number of additional contexts considered by RESI in each step depends only linearly on the maximal rank ρ in Σ, and all contexts are constructed from contexts and subtrees that are already present in the table. Moreover, in contrast to ALTEX [5] we do not add a context to E unless it actually eliminates an inclusion which makes the table even more compact and spares us the membership queries that would have been needed to fill in the additional cells.

4 Conclusion

We have chosen the setting of learning from membership queries and given positive data also because it has often been considered as a most natural one – for example, children receive correct sentences as input and may test their hypotheses by formulating sentences of their own. However, as mentioned in Subsection 3.4, the algorithm by Bollig et al. [3] for the inference of RFSA from membership and equivalence queries can be adapted to trees in a straightforward manner using the notions defined in Subsection 3.1 as well. A third setting to be investigated is learning from finite positive and negative samples (see [2] for the string case) as suggested in [4]. The author is currently writing down a meta-algorithm for the inference of RFTA that covers all three settings in the style of [13].

Concerning other kinds of structures, as can be deduced from [13], RESI can be directly adapted to so-called *multi-dimensional trees*. Multi-dimensional tree languages are associated to mildly context-sensitive string languages, a notion of great importance in the field of formal linguistics since it is supposed to cover a vast majority of non-contextfree phenomena in natural (human) language.

Recently, Brzozowski and Tamm [14] have presented another special case of NFA for strings which is a generalization of an RFSA and is unique for every regular language. Since the latter is an important property for convergence in learning theory it would be interesting to explore if (a) those automata can be inferred in the settings considered here by the current methods without major modifications, and (b) if the notion and learnability is extendable to trees.

Potential applications for learning via residual languages are all those that particularly benefit from a succinct description. One application named in [3] is automatic verification. Other areas specifically related to trees include the extraction of information from semi-structured data that may for example be given in XML (also consult `http://mostrare.lille.inria.fr`), or various applications in computational linguistics again since linguistic structure is often represented in tree form and one can imagine a range of situations where one might want to derive a succinct description for the language under consideration from huge amounts of data contained in treebanks or similar databases.

References

1. Denis, F., Lemay, A., Terlutte, A.: Residual finite state automata. Fundamentae Informaticae **51** (2002) 339–368
2. Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using RFSA. In: Proceedings of ALT 2001. Volume 2225 of LNCS., Springer (2001) 348–363
3. Bollig, B., Habermehl, P., Kern, C., Leucker, M.: Angluin-style learning of NFA. In: Online Proceedings of IJCAI 21. (2009)
4. Carme, J., Gilleron, R., Lemay, A., Terlutte, A., Tommasi, M.: Residual finite tree automata. In: Proceedings of DLT 2003. Volume 2710 of LNCS., Springer (2003) 171–182
5. Besombes, J., Marion, J.Y.: Learning tree languages from positive examples and membership queries. Theoretical Computer Science **382** (2007) 183–197
6. Angluin, D.: Learning regular sets from queries and counterexamples. Information and Computation **75**(2) (1987) 87–106
7. Gold, E.: Language identification in the limit. Information and Control **10**(5) (1967) 447–474
8. Hopcroft, J., Ullmann, J.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley Longman (1990)
9. Angluin, D.: A note on the number of queries needed to identify regular languages. Information and Control **51**(1) (1981) 76–87
10. Sakakibara, Y.: Learning context-free grammars from structural data in polynomial time. Theoretical Computer Science **76**(2–3) (1990) 223–242
11. Drewes, F., Högberg, J.: Learning a regular tree language from a teacher. In: Proceedings of DLT 2003. Volume 2710 of LNCS., Springer (2003) 279–291
12. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree Automata Techniques and Applications. Online publication (2007)
13. Kasprzik, A.: Polynomial learning of regular multi-dimensional tree languages in different settings: A meta-algorithm. Technical report, 10-1, Univ. Trier (2010)
14. Brzozowski, J., Tamm, H.: Theory of Átomata. In: Proceedings of DLT (to appear). LNCS, Springer (2011)