Learning Tree Languages from Positive Examples and Membership Queries

Jérôme Besombes [a,b] Jean-Yves Marion [b,c]

[a] *France Telecom R&D*
2 avenue Pierre Marzin
22300 Lannion

[b] *LORIA*
615, rue du jardin botanique
54602 Villers-lès-Nancy, France

[c] *INPL-Ecole Nationale Supérieure des Mines de Nancy*
Parc de Saurupt,
54042 Nancy CEDEX, France

Jerome.Besombes@rd.francetelecom.com, Jean-Yves.Marion@loria.fr

1 Introduction

1.1 Some linguistic motivations

One of the most astonishing discovery of Chomsky [13] is the universal grammar which is a model of how human language works. The universal grammar is an innate combinatorial system from which every language, French, English, Japanese, can be derived. What is the implication of Chomsky's universal grammar for grammatical inference? We think it gives a strong intuition for the mathematical modeling of language learning. Before going further, let us focus on the linguistic aspect of the language acquisition processes.

Recent works of psycho-linguists like Pinker [27] or Christophe [14]

advocate that the universal grammar plays the role of a learning device for children. A child is able to determine whether or not a sentence is grammatically correct, even if (s)he does not know the meaning of each word. Of course, semantics speed up the learning process, but they are not necessary. And, it is fascinating to see that a child needs only few pieces of information in order to learn a language (poverty-of-stimulus hypothesis [13]).

Another important feature is our capacity to guess mental tree structured representations of phrases. How a child is able to do that is beyond the scope of this paper. However the child language acquisition process is not based on the construction of a huge finite automaton with probabilistic transitions, because there is an infinity of valid sentences and so he cannot learn it. On the other hand, we have the ability to generate an infinite number of sentences.

To sum up this brief discussion, we take as hypothesis that a child computes a grammar from tree structured sentences.

1.2 A mathematical model

Can we give a mathematical model of the language acquisition which corroborates this theory?
The grammatical inference paradigm of Gold [22] is a good candidate. Indeed, the inputs of the learning process are just examples of the target language. And there is no interaction with the environment. But this paradigm is too weak to be plausible. For this reason, we add to Gold's paradigm an oracle which answers to membership queries. Hence, the grammatical inference is based on positive examples and membership questions of computed elements, as introduced by Angluin in [2]. This learning model agrees with the poverty-of-stimulus hypothesis. Indeed the interaction with the environment is very weak. For example, a child asks something, but nobody understands. From this lack of reaction from the environment, he may deduce that the sentence

is wrong, and so not in the language. We insist on the fact that membership queries are a minimal information which can be inferred from a dialog.

On the other hand, negative examples are not necessary because for example a parent does not say incorrect sentences to a child. One might think about other kind of queries like equivalence queries as suggested by Angluin [3]. But there are not necessary as we shall see and there are unrealistic in a linguistic context. In conclusion, our learning model seems quite adequate with respect to our initial motivation, even if we are aware that our description is a bit rough.

Now, we have set the learning paradigm, we have to say what are the languages which are targeted. As we have said, we can assume that a child has a kind of parser which transforms a (linear) sentence into a tree representation. So, we learn tree languages. Regular tree languages are the bare bone of several linguistic formalisms like classical categorial grammars for which learnability has been studied in [24, 11], dependency languages [16, 7, 10] or TAG derivations.

1.3 The results

We establish that the whole set of regular tree languages is efficiently identifiable from membership queries and positive examples. The running time of the learning algorithm is polynomial in the size of the input examples. The efficiency is a necessary property of our model. The difficulty is to construct trees to ask membership queries and which give useful information to proceed in the inference process.

1.4 A web application

There are other applications of our result. For example, an XML document is a tree, if we forget links. Now, say that we try to determine a Document Type Definition (the DTD grammar which generates XML documents). For this, we can read correct XML documents from a server which forms a set of positive examples. Then, we can build an XML document and make a membership query by sending it to the server. If no error occurs then the document is in the language, otherwise it is not.

1.5 Related works

1.5.1 Learning from positive examples and membership queries

Angluin considers the same learning paradigm in [2] for the class of regular word languages. The notion of observation table which is defined in [3] is already implicitly used in [2]. However, we can not extend in a straight-forward way the algorithm of [2] as it is explained by Sakakibara in [29]. In Section 4.3, we shall compare more precisely these works with our approach.

1.5.2 Other paradigms

Sakakibara studied grammatical inference of languages of unlabeled derivation trees of context free grammars. In [30], he extends the result of [3] by learning with membership queries and equivalence queries. The possibility of asking the teacher whether a calculated hypothesis corresponds to the target language seems not to be relevant for the aim of constructing a model of natural language process. In [29, 28] Sakakibara uses positive and negative examples with membership queries in two slight different contexts. In Comparson with [28], we have shown that negative examples are not necessary and that the running time of our learning algorithm is polynomial. It is worth noticing that the

set of all unlabeled context free derivation tree languages is a strict subclass of the set of regular tree languages. Therefore, we learn more and in a weaker setting since negative examples are not necessary in our work. This is important when we are learning from structured examples. Indeed, the language of structured examples may not come from a context free grammar but still the word language is context free.

Inference of regular tree languages from positive examples only, has been studied in [20, 23, 21]. In [7], we define the notion of reversible tree languages following the concept of reversible word languages suggested by Angluin [1]. We study further reversible tree grammars in [8, 9, 6]. All these studies are in the PhD thesis of Besombes [5]. Finally, we refine this notion in order to delineate a class of reversible categorial grammars [11]. Note also the work [26] on reversible grammars. In [19], Fernau defines a learnable subclasses of tree languges, and in [12], learning is studied from a stochastic point of view.

Inference of regular tree languages has been studied in [17]; the learning algorithm is based on membership queries and equivalence queries and this result constitutes an extension of Sakakibara's works. In [18], a polynomial version of the former learning algorithm has been developed.

2 Regular tree languages

We give a short presentation of regular tree languages, which is based on [15].

2.1 Terms, contexts and subterms

A ranked alphabet $\mathcal{V}$ is a finite set of symbols with a rank function *arity* from $\mathcal{V}$ to $\mathbb{N}$. The set $\mathcal{T}(\mathcal{V})$ of terms is inductively defined as follows. A symbol of arity 0 is in $\mathcal{T}(\mathcal{V})$, and if $\mathtt{f}$ is a symbol of

arity n and $\mathtt{t}_1, \ldots, \mathtt{t}_n$ are in $\mathcal{T}(\mathcal{V})$, then $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ is in $\mathcal{T}(\mathcal{V})$. The size of a term is the number of symbols occurring in it.

A *context* is a term $\mathtt{c}[\diamond]$ containing a special variable $\diamond$ which has only one occurrence. The variable $\diamond$ marks an empty place in a term. In particular, $\diamond$ is a context called the *empty context*. The substitution of $\diamond$ by a term $\mathtt{s}$ is the term noted $\mathtt{c}[\mathtt{s}]$.

A *subterm* $\mathtt{s}$ of a term $\mathtt{t}$ is a term such that there is a context $\mathtt{c}[\diamond]$ which satisfies $\mathtt{t} = \mathtt{c}[\mathtt{s}]$. For a set of terms $\mathcal{E}$, $\mathcal{S}(\mathcal{E})$ is the set of subterms of the terms of $\mathcal{E}$.

We define the set $\mathcal{E}[\diamond]$ as the set of contexts obtained by replacing, in each term of $\mathcal{E}$, one occurrence of a subterm by $\diamond$. In particular, the empty context $\diamond$ is in $\mathcal{E}[\diamond]$. More precisely,

$$\mathcal{E}[\diamond] = \{\mathtt{c}[\diamond] \mid \exists \mathtt{s}, \mathtt{c}[\mathtt{s}] \in \mathcal{E}\}$$

2.2 Automata and regular tree languages

A *bottom up non-deterministic tree automaton* (NFTA) is a quadruple $\mathcal{A} = \langle \mathcal{V}, \mathcal{Q}, \mathcal{Q}_F, \underset{\mathcal{A}}{\longrightarrow} \rangle$ where $\mathcal{V}$ is a ranked alphabet, $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_F \subseteq \mathcal{Q}$ is the set of final states, and $\underset{\mathcal{A}}{\longrightarrow}$ is the set of transitions. A transition is a rewrite rule of the form $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{A}}{\longrightarrow} q$ where q and $q_1, \ldots, q_n$ are states of $\mathcal{Q}$, and $\mathtt{f}$ is a symbol of arity n. In particular, a transition may be just of the form $\mathtt{a} \underset{\mathcal{A}}{\longrightarrow} q$ where $\mathtt{a}$ is a symbol of arity 0.

The single derivation relation $\underset{\mathcal{A}}{\longrightarrow}$ is defined so that $\mathtt{t} \underset{\mathcal{A}}{\longrightarrow} \mathtt{s}$ if and only if there is a transition $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{A}}{\longrightarrow} q$ such that for a context $\mathtt{c}[\diamond]$, $\mathtt{t} = \mathtt{c}[\mathtt{f}(q_1, \ldots, q_n)]$ and $\mathtt{s} = \mathtt{c}[q]$. The derivation relation $\overset{*}{\underset{\mathcal{A}}{\longrightarrow}}$ is the reflexive and transitive closure of $\underset{\mathcal{A}}{\longrightarrow}$.

The automaton $\mathcal{A}$ recognizes the language $\mathcal{L}_\mathcal{A}$ where

$$\mathcal{L}_\mathcal{A} = \{\mathtt{t} \in \mathcal{T}(\mathcal{V}) \ : \ \mathtt{t} \overset{*}{\underset{\mathcal{A}}{\longrightarrow}} q_F \text{ and } q_F \in \mathcal{Q}_F\}$$

A tree language is regular if and only if it is recognized by an automaton. Throughout, we say "automaton" to refer to NFTA.

A finite tree automaton is *deterministic* (DFTA) if there are no two rules with the same left hand side. It is well known that NFTA and DFTA recognize the same class of languages.

Example 1 *Consider the DFTA* $\mathcal{A} = \langle \{a, b, c\}, \{q_F, q_1, q_2, q_3, q_4\}, \{q_F\}, \underset{\mathcal{A}}{\rightarrow} \rangle$, *where* $\underset{\mathcal{A}}{\rightarrow}$ *is the following set of transitions:*

$$a(q_2, q_3) \underset{\mathcal{A}}{\rightarrow} q_F$$

$$b(q_2) \underset{\mathcal{A}}{\rightarrow} q_1 \qquad c(q_4) \underset{\mathcal{A}}{\rightarrow} q_3$$

$$b(q_1) \underset{\mathcal{A}}{\rightarrow} q_2 \qquad c(q_3) \underset{\mathcal{A}}{\rightarrow} q_4$$

$$b \underset{\mathcal{A}}{\rightarrow} q_1 \qquad c \underset{\mathcal{A}}{\rightarrow} q_3$$

The tree $a(b(b), c(c(c)))$ *belongs to* $\mathcal{L}_\mathcal{A}$. *Indeed we have:*

$$a(b(b), c(c(c))) \underset{\mathcal{A}}{\rightarrow} a(b(q_1), c(c(c))) \underset{\mathcal{A}}{\rightarrow} a(q_2, c(c(c)))$$

$$\underset{\mathcal{A}}{\rightarrow} a(q_2, c(c(q_3))) \underset{\mathcal{A}}{\rightarrow} a(q_2, c(q_4)) \underset{\mathcal{A}}{\rightarrow} a(q_2, q_3) \underset{\mathcal{A}}{\rightarrow} q_F$$

$\mathcal{L}_\mathcal{A}$ *is the tree language* $\{a(b^{2n+2}, c^{2m+1}) \ : \ n, m \in \mathbb{N}\}$.

2.3 Canonical automaton

For a given tree language $\mathcal{L}$, the congruence $\equiv_\mathcal{L}$ is defined by $\mathsf{t} \equiv_\mathcal{L} \mathsf{s}$ iff for every context $\mathsf{c}[\diamond]$, $\mathsf{c}[\mathsf{t}] \in \mathcal{L}$ iff $\mathsf{c}[\mathsf{s}] \in \mathcal{L}$.

Theorem 1 (Myhill-Nerode Theorem for trees) *Let* $\mathcal{L}$ *be a tree language.* $\mathcal{L}$ *is a regular tree language iff the congruence* $\equiv_\mathcal{L}$ *is of finite index.*

Kozen in [25] has written an elementary proof of the Myhill-Nerode Theorem and has told the story behind.

Throughout, we denote $\delta(\mathsf{t})$ the equivalence class of the term t wrt the congruence $\equiv_\mathcal{L}$.

A finite congruence $\equiv_{\mathcal{L}}$ defines the minimal automaton for $\mathcal{L}$ (up to a renaming of states) as follows.

- the state set is the set of equivalence classes of $\equiv_{\mathcal{L}}$.
- the set of final states is the set of states such that $\delta(\mathtt{t})$ is contained in $\mathcal{L}$,
- the transition rule $\underset{\mathcal{L}}{\rightarrow}$ is the smallest relation such that

$$\mathtt{a} \underset{\mathcal{L}}{\rightarrow} \delta(\mathtt{a})$$
$$\mathtt{f}(\delta(\mathtt{t}_1), \ldots, \delta(\mathtt{t}_n)) \underset{\mathcal{L}}{\rightarrow} \delta(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n))$$

A state q of an automaton $\mathcal{A}$ *accepts* a context $\mathtt{c}[\diamond]$ if $\mathtt{c}[q] \xrightarrow[\mathcal{A}]{*} q_F$. The *canonical* automaton $\mathcal{A}_{\mathcal{L}}$ for $\mathcal{L}$ is the restriction of the minimal automaton of $\mathcal{L}$ to states, which accept at least one context. In other word, a canonical has no "trash" state, but on the other hand it is incomplete.

An *automaton homomorphism* between the `NFTA` $\mathcal{A} = \langle \mathcal{V}, \mathcal{Q}, \mathcal{Q}_F, \underset{\mathcal{A}'}{\longrightarrow} \rangle$ and $\mathcal{A}' = \langle \mathcal{V}', \mathcal{Q}', \mathcal{Q}'_F, \underset{\mathcal{A}'}{\longrightarrow} \rangle$ is a mapping ϕ from $\mathcal{Q}$ and $\mathcal{Q}'$ such that :

(1) ϕ is surjective.
(2) For each transition $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{A}}{\rightarrow} q$ of $\mathcal{A}$, there is a transition $\mathtt{f}(\phi(q_1), \ldots, \phi(q_n)) \underset{\mathcal{A}'}{\longrightarrow} \phi(q)$ in $\mathcal{A}'$.
(3) $\phi(\mathcal{Q}_F) \subseteq \mathcal{Q}'_F$.

This implies that if there is a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$, then $\mathcal{L}_{\mathcal{A}} \subseteq \mathcal{L}_{\mathcal{A}'}$. If ϕ is bijective, it consists in a renaming of the states and $\mathcal{L}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}'}$. Notice that $\mathcal{A}$ has no more states than $\mathcal{A}'$.

3 Learning regular tree languages

3.1 The learning paradigm

The goal is the identification of any unknown regular tree language $\mathcal{L}$ with help of a teacher. The teacher is an oracle which answers to membership queries. The learning process begins with a set of positive examples. Then, a dialogue is established between the learner and the teacher. The learner asks whether or not a new tree belongs to the unknown language. The teacher answers by "yes" or "no" to this query. This learning process halts after a finite number of queries.

We shall provide a necessary and sufficient condition on the examples, to guess the unknown language and so to construct a `DFTA` which recognizes it.

3.2 Representative samples

Informally, a representative sample $\mathcal{E}$ is a finite subset of a regular tree language $\mathcal{L}$ such that each transition of the canonical automaton of $\mathcal{L}$ is used to produce a term of $\mathcal{E}$.

The set $\mathcal{E}$ is a *representative sample* of $\mathcal{L}$ if for each transition $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{L}}{\rightarrow} q$, there is a term $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ in $\mathcal{S}(\mathcal{E})$ such that $\forall 1 \leq i \leq n$, $\delta(\mathtt{t}_i) = q_i$. That is, $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ matches the rule $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{L}}{\rightarrow} q$.

As a consequence, for each state q of the canonical automaton $\mathcal{A}_\mathcal{L}$ of $\mathcal{L}$, there is a subterm $\mathtt{t}$ of $\mathcal{S}(\mathcal{E})$ such that $\delta(\mathtt{t}) = q$. And inversely, for each subterm $\mathtt{t}$ in $\mathcal{S}(\mathcal{E})$, there is a state q of $\mathcal{A}_\mathcal{L}$ such that $\delta(\mathtt{t}) = q$. Indeed, there is a term $\mathtt{s} \in \mathcal{E}$ and a context $\mathtt{c}[\diamond]$ of $\mathcal{E}[\diamond]$ such that q accepts $\mathtt{c}[\diamond]$ and $\mathtt{s} = \mathtt{c}[\mathtt{t}]$.

Example 2 *This example illustrates the fact that several distinct languages can have identical representative samples. This is not*

surprising since Gold demonstrated that word regular languages are not learnable from positive examples. So, the knowledge of any set of example, and a fortiori a representative sample, is not sufficient in order to infer the right language.

Let $\mathcal{A}$ be the `DFTA` *defined in Example 1 and $\mathcal{A}'$ the* `DFTA` *defined by:*

$$a(q_1, q_2) \xrightarrow[\mathcal{A}']{} q_F$$
$$b(q_1) \xrightarrow[\mathcal{A}']{} q_1 \qquad c(q_2) \xrightarrow[\mathcal{A}']{} q_2$$
$$b \xrightarrow[\mathcal{A}']{} q_1 \qquad c \xrightarrow[\mathcal{A}']{} q_2$$

where q_F is the unique final state of $\mathcal{A}'$. From this definition, we have

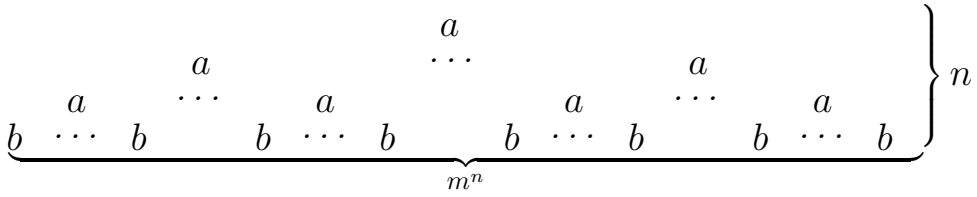
$$\mathcal{L}_{\mathcal{A}'} = \{a(b^n, c^m) : \ n, m \in \mathbb{N}^*\}$$

and the singleton set

$$\{a(b(b), c(c(c)))\}$$

is a representative sample for both $\mathcal{L}_{\mathcal{A}}$ and $\mathcal{L}_{\mathcal{A}'}$.

Remark 1 *The question of the size of a minimal representative sample relative to the size of the minimal automaton $\mathcal{A}$ of a language $\mathcal{L}$ is of interest to be discussed. In contrast with the case of word language, there is no polynomial relation between the size (the total number of nodes) of a minimal representative sample and the size (the number of states) of the canonical* `DFTA`*. For instance, for two integers n and m, the singleton language containing the following tree:*



where the arity of a is m and the arity of b is 0, is a regular tree language recognized by the above canonical `DFTA`*.*

$$\begin{aligned} a(q_n, \ldots, q_n) &\rightarrow q_F \\ &\vdots \\ a(q_1, \ldots, q_1) &\rightarrow q_2 \\ b &\rightarrow q_1 \end{aligned}$$

The size of the representative sample which the singeton language above, is $m^{n+1} - 1$.

3.3 Observation tables

Following the method of Angluin [4], information obtained from the queries is stored in a table. Let $\mathcal{L}$ be a tree language, $\mathcal{E}$ be a finite set of terms and F be a finite set of contexts.

The *observation table* $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is the table defined as follows

- Rows are indexed by subterms in $\mathcal{S}(\mathcal{E})$,
- Columns are indexed by contexts of F.
- The cell indexed by the subterm t and the context. $\mathsf{c}[\diamond]$ is noted $T_{\mathcal{L}}(\mathsf{t}, \mathsf{c}[\diamond])$. The content of $T_{\mathcal{L}}(\mathsf{t}, \mathsf{c}[\diamond])$ is defined by

$$T_{\mathcal{L}}(\mathsf{t}, \mathsf{c}[\diamond]) = \begin{cases} \mathtt{1} \text{ if } \mathsf{c}[\mathsf{t}] \in \mathcal{L} \\ \mathtt{0} \text{ otherwise} \end{cases}$$

We call row(t) the binary word of $\{0,1\}^*$ corresponding to the reading from left to right of the row labeled by t in T.

The induced automaton $\mathcal{A}_T = \langle \mathcal{V}, \mathcal{Q}, \mathcal{Q}_F, \underset{T}{\rightarrow} \rangle$ build from an observation table $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is defined thus.

- the ranked alphabet $\mathcal{V}$ is the set of symbols occurring in $\mathcal{E}$,
- the set of states is $\mathcal{Q} = \{\text{row}(\mathsf{t}), \mathsf{t} \in \mathcal{S}(\mathcal{E})\}$,
- the set of final states $\mathcal{Q}_F = \{\text{row}(\mathsf{t}), \mathsf{t} \in \mathcal{L}\}$,

- the set of transitions $\underset{T}{\rightarrow}$ is the smallest relation satisfying

$$\mathtt{a} \underset{T}{\rightarrow} \mathrm{row}(\mathtt{a})$$
$$\mathtt{f}(\mathrm{row}(\mathtt{t}_1), \ldots, \mathrm{row}(\mathtt{t}_n)) \underset{T}{\rightarrow} \mathrm{row}(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n))$$

In general, $\mathcal{A}_T$ is nondeterministic and not complete. The language recognized by $\mathcal{A}_T$ is $\mathcal{L}_{\mathcal{A}_T}$.

Lemma 1 *Let $\mathcal{L}$ be a regular tree language. Assume that $\mathcal{E}$ is a representative sample for $\mathcal{L}$, and F is a finite set of contexts. If $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$, then we have $\mathcal{L} \subseteq \mathcal{L}_{\mathcal{A}_T}$.*

Proof We define an automaton homomorphism ϕ from the canonical automaton $\mathcal{A}_{\mathcal{L}}$ onto $\mathcal{A}_T$. For each subterm $\mathtt{t}$ of $\mathcal{E}$, we set $\phi(\delta(\mathtt{t})) = \mathrm{row}(\mathtt{t})$. The mapping ϕ is well defined because if $\delta(\mathtt{t}) = \delta(\mathtt{s})$ then we necessarily have $\mathrm{row}(\mathtt{t}) = \mathrm{row}(\mathtt{s})$. Since $\mathcal{E}$ is a representative sample of $\mathcal{L}$, ϕ is defined on all states of $\mathcal{A}_{\mathcal{L}}$. It is clearly surjective. Finally, for each transition $\mathtt{f}(q_1, \ldots, q_n) \underset{\mathcal{L}}{\rightarrow} q$, there is a term $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ in $\mathcal{S}(\mathcal{E})$ which matches $\mathtt{f}(q_1, \ldots, q_n)$. That is $\forall 1 \leq i \leq n$, $\delta(\mathtt{t}_i) = q_i$ and necessarily $\delta(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) = q$. So, $\mathtt{f}(\phi(q_1), \ldots \phi(q_n)) \underset{\mathcal{A}_T}{\longrightarrow} \phi(q)$. □

The number of states of the automaton $\mathcal{A}_T$ is always less than or equal to the number of states of $\mathcal{A}_{\mathcal{L}}$. This remark and the previous Lemma lead to the following conclusion.

Corollary 2 *Let $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ be an observation table where $\mathcal{L}$ is a regular tree language, $\mathcal{E}$ a representative sample for $\mathcal{L}$, F a set of contexts.*

If $\mathcal{L} \neq \mathcal{L}_{\mathcal{A}_T}$, then there are two subterms $\mathtt{t}$ and $\mathtt{t}'$ of $\mathcal{E}$ such that $row(\mathtt{t}) = row(\mathtt{t}')$ and $\delta(\mathtt{t}) \neq \delta(\mathtt{t}')$.

Proof By contradiction, if it is not the case, then the homomorphism ϕ defined in the previous proof should be an isomorphism, and so $\mathcal{L} = \mathcal{L}_{\mathcal{A}_T}$. □

Example 3 *Consider the tree language $\mathcal{L}_{\mathcal{A}}$ defined in Example 1. Put*

$$\mathcal{E} = \{a(b(b), c(c(c)))\}$$

The set $\mathcal{E}$ is a representative sample. Next, define

$$\begin{aligned}\mathsf{F} = \mathcal{E}[\diamond] = \{&\diamond, a(\diamond, c(c(c))), a(b(\diamond), c(c(c))), a(b(b), \diamond), a(b(b), c(\diamond)),\\ &a(b(b), c(c(\diamond)))\}\end{aligned}$$

The corresponding observation table $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is the following:

	$\diamond$	$a(\diamond, c(c(c)))$	$a(b(\diamond), c(c(c)))$	$a(b(b), \diamond)$	$a(b(b), c(\diamond))$	$a(b(b), c(c(\diamond)))$
$a(b(b), c(c(c)))$	*1*	*0*	*0*	*0*	*0*	*0*
$b(b)$	*0*	*1*	*0*	*0*	*0*	*0*
b	*0*	*0*	*1*	*0*	*0*	*0*
$c(c(c))$	*0*	*0*	*0*	*1*	*0*	*1*
$c(c)$	*0*	*0*	*0*	*0*	*1*	*0*
c	*0*	*0*	*0*	*1*	*0*	*1*

The table T defines the NFTA $\mathcal{A}_T$ as:

- $\mathcal{V}_T = \{a, b, c\}$
- $\mathcal{Q}_T = \{\mathit{100000}, \mathit{010000}, \mathit{001000}, \mathit{000101}, \mathit{000010}\}$
- $\mathcal{Q}_{F,T} = \{\mathit{100000}\}$
- $\underset{T}{\longrightarrow}$ *is the following set of transitions*

$$a(\mathit{010000}, \mathit{000101}) \underset{\mathcal{A}_T}{\longrightarrow} \mathit{100000}$$

$$b(\mathit{010000}) \underset{\mathcal{A}_T}{\longrightarrow} \mathit{001000} \qquad c(\mathit{000010}) \underset{\mathcal{A}_T}{\longrightarrow} \mathit{000101}$$

$$b(\mathit{001000}) \underset{\mathcal{A}_T}{\longrightarrow} \mathit{010000} \qquad c(\mathit{000101}) \underset{\mathcal{A}_T}{\longrightarrow} \mathit{000010}$$

$$b \underset{\mathcal{A}_T}{\longrightarrow} \mathit{001000} \qquad c \underset{\mathcal{A}_T}{\longrightarrow} \mathit{000101}$$

We remark that $\mathcal{A}_T = \phi(\mathcal{A})$, where $\mathcal{A}$ is the `DFTA` *introduced in Example 1 and ϕ is the renaming defined by: $\phi(q_F) =$ `100000`,*

$\phi(q_1) = \mathit{001000}$, $\phi(q_2) = \mathit{010000}$, $\phi(q_3) = \mathit{000101}$, $\phi(q_4) = \mathit{000010}$.

An observation table $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is said to be *consistent* if for any terms $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ and $\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)$ in $\mathcal{S}(\mathcal{E})$, if $\forall 1 \leq j \leq n$ we have :

$$\mathrm{row}(\mathtt{t}_j) = \mathrm{row}(\mathtt{t}'_j)$$

then

$$\mathrm{row}(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) = \mathrm{row}(\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n))$$

Lemma 2 *Assume that $\mathcal{L}$ is a regular tree language, $\mathcal{E}$ is a finite set of terms and F is a finite set of contexts.*

Then,the observation table $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is consistent if and only if the induced automaton $\mathcal{A}_T$ is deterministic.

Proof This equivalence is straightforward from definitions of consistency of T and of the induced automaton $\mathcal{A}_T$. □

Given an observation table $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$, a context $\mathtt{c}[\diamond]$ is a *separating* context if there are two subterms $\mathtt{t}$ and $\mathtt{t}'$ of $\mathcal{E}$ such that

$$\mathrm{row}(\mathtt{t}) = \mathrm{row}(\mathtt{t}')$$

and

$$\mathtt{c}[\mathtt{t}] \in \mathcal{L} \text{ and } \mathtt{c}[\mathtt{t}] \notin \mathcal{L}$$

This implies that $\delta(\mathtt{t}) \neq \delta(\mathtt{t}')$. It is worth noticing that a separating context $\mathtt{c}[\diamond]$ is not in F because both lines row($\mathtt{t}$) and row($\mathtt{t}'$) are identical in T.

Lemma 3 *Let $\mathcal{L}$ be a regular tree language, $\mathcal{E}$ a set of terms and F a set of contexts. Assume that $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ is an observation table which is not consistent. Then, there is a separating context.*

Proof Since T is not consistent, there are two terms of $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ and $\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)$ of $\mathcal{S}(\mathcal{E})$ such that $\forall 1 \leq j \leq n, \mathrm{row}(\mathtt{t}_j) = \mathrm{row}(\mathtt{t}'_j)$

and on the other hand

$$\text{row}(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) \neq \text{row}(\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n))$$

This means that $\delta(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) \neq \delta(\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n))$ This implies that there is at least an index i such that $\delta(\mathtt{t}_i) \neq \delta(\mathtt{t}'_i)$. So, there is a separating context $\mathtt{c}[\diamond]$ such that $\mathtt{c}[\mathtt{t}] \in \mathcal{L}$ and $\mathtt{c}[\mathtt{t}] \notin \mathcal{L}$. □

From all this, it follows that an observation table induces an automaton $\mathcal{A}_T$ which approximates from below the target language $\mathcal{L}$. Indeed, Lemma 1 claims that the number of states of $\mathcal{A}_T$ is always less than the one of the canonical automaton $\mathcal{A}_\mathcal{L}$ of $\mathcal{L}$. By finding a separating context, we increase the number of states of $\mathcal{A}_T$ and we get closer to the canonical automaton of $\mathcal{L}$. The process is repeated until the observation table is consistent. This poses the question of whether the automaton $\mathcal{A}_T$ induced by a consistent observation table is the correct one. Recall that an observation table $T = T_\mathcal{L}(\mathcal{E}, \mathsf{F})$ is built from a set $\mathcal{E}$ of positive examples and from F a set of contexts. A consistent observation table $T = T_\mathcal{L}(\mathcal{E}, \mathsf{F})$ yields the canonical automaton for $\mathcal{L}$, if two conditions are satisfied, as we shall establish in the next Lemma. First, the set $\mathcal{E}$ is a representative sample. Second, for each state q of the canonical automaton $\mathcal{A}_\mathcal{L}$, there is a context of F which is accepted by q. The later condition is fulfils by requiring that F contains $\mathcal{E}[\diamond]$.

The next question to solve is how to find a separating context. The following Lemma is the cornerstone of the paper because:

- it explains how to construct a separating context from a non-consistent observation table
- it claims that it is sufficient to consider only context with one hole. That means that there is only one error and not several which would imply to consider contexts with many holes. This observation implies that we need to consider only a linear number of contexts wrt the symbol arity. Otherwise, it would be exponential in the symbol arity.

Lemma 4 *Let $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ be an observation table where $\mathcal{L}$ is a regular tree language, $\mathcal{E}$ a representative sample for $\mathcal{L}$, and F a set of contexts containing $\mathcal{E}[\diamond]$.*

Assume that T is not consistent. Then, there are two terms $f(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ and $f(\mathsf{t}'_1, \ldots, \mathsf{t}'_n)$ of $\mathcal{S}(\mathcal{E})$ such that

$$\begin{aligned} row(f(\mathsf{t}_1, \ldots, \mathsf{t}_n)) &\neq row(f(\mathsf{t}'_1, \ldots \mathsf{t}'_n)) \\ row(\mathsf{t}_i) &= row(\mathsf{t}'_i) \qquad\qquad \forall 1 \leq i \leq n \end{aligned}$$

and there is an index i such that

$$\begin{aligned} \delta(\mathsf{t}_i) &\neq \delta(\mathsf{t}'_i) \\ \delta(\mathsf{t}_j) &= \delta(\mathsf{t}'_j) \qquad\qquad \forall j \neq i \end{aligned}$$

Proof Since T is not consistent, there are separating contexts by Lemma 3. Take $\mathsf{c}[\diamond]$ to be a separating context which is minimal with respect to the size. There are two terms s and s' in $\mathcal{S}(\mathcal{E})$ such that

$$\begin{gathered} row_T(\mathsf{s}) = row_T(\mathsf{s}') \\ \mathsf{c}[\mathsf{s}] \in \mathcal{L} \text{ and } \mathsf{c}[\mathsf{s}'] \notin \mathcal{L} \end{gathered}$$

Since $\mathsf{c}[\diamond]$ is not in F and $\diamond \in \mathsf{F}$, we see that for some context $\mathsf{d}[\diamond]$ and terms $\mathsf{s}_1, \ldots, \mathsf{s}_{i-1}, \mathsf{s}_{i+1}, \ldots, \mathsf{s}_n$, we have

$$\mathsf{c}[\diamond] = \mathsf{d}[\mathtt{f}(\mathsf{s}_1, \ldots, \mathsf{s}_{i-1}, \diamond, \mathsf{s}_{i+1}, \ldots, \mathsf{s}_n)]$$

Notice that the size of $\mathsf{d}[\diamond]$ is strictly smaller than the size of $\mathsf{c}[\diamond]$. So, $\mathsf{d}[\diamond]$ is not a separating context. Otherwise, it would falsify the minimality condition on $\mathsf{c}[\diamond]$.

Since $\mathsf{c}[\mathsf{s}] = \mathsf{d}[\mathtt{f}(\mathsf{s}_1, \ldots, \mathsf{s}_{i-1}, \mathsf{s}, \mathsf{s}_{i+1}, \ldots, \mathsf{s}_n)]$ is in $\mathcal{L}$, there is a transition in $\mathcal{A}_{\mathcal{L}}$ of the form

$$\mathtt{f}(q_1, \ldots, q_n) \xrightarrow[\mathcal{L}]{} r$$

where $\delta(\mathsf{s}_j) = q_j$ for each $j \neq i$ and $\delta(\mathsf{s}) = q_i$.

By definition of a representative sample of $\mathcal{L}$, there is a term $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ in $\mathcal{S}(\mathcal{E})$ which matches the above transition. That is, $\delta(\mathtt{t}_j) = q_j$ for each j and

$$\delta(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) = r$$

Now, there is a context $\mathtt{d}'[\diamond]$ such that $\mathtt{d}'[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)] \in \mathcal{E} \subseteq \mathcal{L}$. Since F contains $\mathcal{E}[\diamond]$, the context $\mathtt{d}'[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \diamond, \mathtt{t}_{i+1}, \ldots, \mathtt{t}_n)]$ is also in F. We have $\mathtt{d}'[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \mathtt{s}, \mathtt{t}_i, \ldots, \mathtt{t}_n)] \in \mathcal{L}$ because $\delta(\mathtt{t}_i) = \delta(\mathtt{s})$.
Moreover, $\mathrm{row}(\mathtt{s}) = \mathrm{row}(\mathtt{s}')$. So $\mathtt{d}'[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \mathtt{s}', \mathtt{t}_{i+1}, \ldots, \mathtt{t}_n)] \in \mathcal{L}$. Therefore, there is a transition in $\mathcal{A}_\mathcal{L}$

$$\mathtt{f}(q_1, \ldots, q_{i-1}, \delta(\mathtt{s}'), q_{i+1}, \ldots, q_n) \underset{\mathcal{L}}{\longrightarrow} r'$$

Now, we necessarily have $r \neq r'$ because $\mathtt{c}[\diamond]$ is a separating context. Again, there is a subterm $\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)$ in $\mathcal{E}$ which matches the above transition rule.

By contradiction, suppose that

$$\mathrm{row}(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) = \mathrm{row}(\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n))$$

We see that $\mathtt{d}[\diamond]$ becomes a separating context because

$$\mathtt{d}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)] \in \mathcal{L}$$

and

$$\mathtt{d}[\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)] \notin \mathcal{L}$$

Our assumption violates the minimality of $\mathtt{c}[\diamond]$ and leads to the conclusion. □

Lemma 5 *Let $\mathcal{L}$ be a regular tree language, $\mathcal{E}$ a representative sample for $\mathcal{L}$ and F a set of contexts including $\mathcal{E}[\diamond]$. If the table $T = T_\mathcal{L}(\mathcal{E}, \mathsf{F})$ is consistent, then $\mathcal{L}_{\mathcal{A}_T} = \mathcal{L}$.*

Proof Lemma 4 yields that if a table is consistent then two equivalent terms wrt Myhill-Nerode congruence have the same row. From this, we define an automaton homomorphism from $\mathcal{A}_T$ onto $\mathcal{A}_\mathcal{L}$. So, $\mathcal{L}_{\mathcal{A}_T} \subseteq \mathcal{L}$. The conclusion follows by Lemma 1. □

4 The algorithm Altex

4.1 Definition

The algorithm Altex is described in Figure 1. Altex first receives a finite subset $\mathcal{E}$ of an unknown language $\mathcal{L}$. Altex constructs the first observation table $T = T_\mathcal{L}(\mathcal{E}, \mathcal{E}[\diamond])$ by asking membership queries. Then, it checks the consistency of the table. Each time Altex finds the table non-consistent, new contexts are constructed from terms that contradict the consistency. Those calculated contexts are added to the table which is then completed with queries. The process stops when the table is consistent and the automaton $\mathcal{A}_T$ is output.

Input: a finite set of terms $\mathcal{E}$
Initialization: $\mathsf{F} = \mathcal{E}[\diamond]$;
Construct the table $T = T_\mathcal{L}(\mathcal{E}, \mathsf{F})$;
while there is $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ and $\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)$ in $\mathcal{S}(\mathcal{E})$ such that
$\mathrm{row}(\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)) \neq \mathrm{row}(\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n))$ and $\forall 1 \leq i \leq n$, $\mathrm{row}(\mathtt{t}_i) = \mathrm{row}(\mathtt{t}'_i)$
do
  Find a context $\mathtt{c}[\diamond]$ in F such that
  $\mathtt{c}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)] \in \mathcal{L}$ and $\mathtt{c}[\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)] \notin \mathcal{L}$;
  $\mathsf{F} = \mathsf{F} \cup \{\mathtt{c}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \diamond, \mathtt{t}_{i+1}, \ldots, \mathtt{t}_n)], 1 \leq i \leq n\}$;
  $\{n$ contexts are added$\}$
  Construct $T = T_\mathcal{L}(\mathcal{E}, \mathsf{F})$;
end while;
Return the automaton $\mathcal{A}_T$.

Fig. 1. The learning algorithm Altex

From the definition of consistency, ALTEX can easily verify whether the table T constructed with help of membership queries is consistent or not. In the case where T is not consistent, the problem is that the algorithm has to find by itself (no counter-example is allowed) a new context that will separate two equivalent rows. The key point is that the input set of terms $\mathcal{E}$ is representative which provides the ability to determining such separating contexts.

Lemma 6 *Assume that the inputs of* ALTEX *is a representative sample. If the table* $T = T_{\mathcal{L}}(\mathcal{E}, \mathsf{F})$ *is not consistent,* ALTEX *calculates a separating-context.*

Proof ALTEX collects every pair of terms $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ and $\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)$ in $\mathcal{S}(\mathcal{E})$, such that there is a context $\mathtt{c}[\diamond]$ in F with

$$\forall 1 \leq i \leq n,\ \mathrm{row}(\mathtt{t}_i) = \mathrm{row}(\mathtt{t}'_i)$$
$$\mathtt{c}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)] \in \mathcal{L} \text{ and } \mathtt{c}[\mathtt{f}(\mathtt{t}'_1, \ldots, \mathtt{t}'_n)] \notin \mathcal{L}$$

Among the pairs gathered as above, Lemma 4 states that there is an index i such that

$$\forall 1 \leq j \neq i \leq n,\ \delta(\mathtt{t}_j) = \delta(\mathtt{t}'_j)$$

and

$$\delta(\mathtt{t}_i) \neq \delta(\mathtt{t}'_i)$$

So, $\mathtt{c}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \diamond, \mathtt{t}_{i+1}, \ldots, \mathtt{t}_n)]$ is a separating context which is added to F. The rows of $\mathtt{t}_i$ and $\mathtt{t}'_i$ are now different in

$$T_{\mathcal{L}}(\mathcal{E}, \mathsf{F} \cup \{\mathtt{c}[\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_{i-1}, \diamond, \mathtt{t}_{i+1}, \ldots, \mathtt{t}_n)]\})$$

□

Theorem 1 *The algorithm* ALTEX *identifies the class of regular tree languages in polynomial time.*

Proof The algorithm ALTEX starts with the construction of $T_{\mathcal{L}}(\mathcal{E}, \mathcal{E}[\diamond])$ and enters the while-loop. If the program leaves this loop, the observation table is consistent and by Lemma 5, the automaton given as output is correct. It remains to show that ALTEX terminates. From Lemma 6, each time the loop is processed, a new separating-context is added to the table. And, two faulty rows which were identical are now different. The number of states of the automaton $\mathcal{A}_T$ strictly increase. From Lemma 1, this number is always lower than or equal to the number of states of the canonical automaton for $\mathcal{L}$. This implies that the loop may be processed only a finite number of times and by consequence, ALTEX terminates.

The algorithm identifies the class of regular languages because the observation table is consistent at the end of Altex run. If the inputs contain a representative sample of the target language, then the induced automaton is a canonical automaton for the target language by Lemma 5.

The time complexity of ALTEX depends on the size n of a representative sample $\mathcal{E}$ (the sum of the size of terms in $\mathcal{E}$) and the size m of the canonical automaton (number of states) for the language $\mathcal{L}$ to identify. The first observation table has n^2 cells. Then, the number of rows doesn't change and the number of columns increases until the table is consistent. Let $p \leq n$ be the greatest arity of a symbol. At each step, there is at most p new contexts which are added to the table. And the while loop is bounded by m. Now, since we have $m \leq n$, we conclude that the runtime is bounded by $O(n^3)$. □

We noticed that any finite set containing a representative sample is also a representative sample. This implies that if we consider an incremental version of ALTEX (the table is completed as the set of positive examples increases during the process), the algorithm converges. Now, if the input set doesn't contain a representative sample yet, the algorithm calculates a sub-automaton (the recognized language is strictly contained in the target language). The

algorithm terminates on any input and the success of learning is guaranteed if a representative sample is presented as input, which constitutes a weak hypothesis.

4.3 Why does it not blow up?

In [2], Angluin studies the paradigm of learning regular (word) languages from positive examples and queries and in [4], the idea of observation table is introduced. It's interesting to see whether these results may be applied in the case of tree languages. Angluin's algorithm tries any possible transition by considering the set of words $\omega\alpha$, where ω is a prefix and α is a letter of the alphabet. (By analogy, the suffix α is a context when we deal with trees.) To apply this technique in the case of trees, we have to construct, for any subterm $\mathtt{t}$ of $\mathcal{E}$, all terms of the form $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}, \ldots, \mathtt{t}_n)$, for each subterm $\mathtt{t}_j$ of $\mathcal{E}$ and each element $\mathtt{f}$ of arity n in the alphabet. This straight generalization leads to an exponential procedure in the maximum arity of the alphabet.

ALTEX proceeds in a different way. It determines at most p contexts which are candidates to be a separating context. Here p is the maximum arity of a symbol. We are certain that among those p contexts there is one separating contexts.

5 Examples

In Example 2, we saw that the tree $a(b(b), c(c(c)))$ is a representative sample for the tree language

$$\mathcal{L}_{\mathcal{A}} = \{a(b^{2n+2}, c^{2m+1}) \ : \ n, m \in \mathbb{N}\}$$

Now suppose that the above singleton is given to ALTEX ; the table constructed with help of membership queries is the table of Example 3. This table is consistent and then, ALTEX outputs the automaton given in the same example. This automaton is a

renaming of $\mathcal{A}$ and the language is learned. If we suppose that, with the same input, the language to learn is

$$\mathcal{L}_{\mathcal{A}'} = \{a(b^n, c^m) \ : \ n, m \in \mathbb{N}^*\}$$

($\{a(b(b), c(c(c))\}$ is representative for this language too), the table is now:

	$\diamond$	$a(\diamond, c(c(c)))$	$a(b(\diamond), c(c(c)))$	$a(b(b), \diamond)$	$a(b(b), c(\diamond))$	$a(b(b), c(c(\diamond)))$
$a(b(b), c(c(c)))$	1	0	0	0	0	0
$b(b)$	0	1	1	0	0	0
b	0	1	1	0	0	0
$c(c(c))$	0	0	0	1	1	1
$c(c)$	0	0	0	1	1	1
c	0	0	0	1	1	1

ALTEX checks again that this table is directly consistent and output the automaton $\phi'(\mathcal{A}')$, where ϕ' is the automaton homomorphism defined by: $\phi'(q_F) = \texttt{100000}$, $\phi'(q_1) = \texttt{011000}$, $\phi'(q_2) = \texttt{000111}$.

ALTEX is defined as an iterative algorithm; if during the process, the observation table is found not to be consistent, the canonical automaton for the target language must have some particular rules: rules which are identical except in a single state of its left hand side. If the canonical automaton has no such rules, the first table constructed by ALTEX is consistent and the language is learned immediately. With the aim of illustrating the iterative behavior, we now propose an automaton specially constructed to have this property. Let so be $\mathcal{L}_{\mathcal{A}''}$ the language defined by the following canonical automaton $\mathcal{A}''$:

$$\begin{array}{rrrr}
a(q_1) \underset{\mathcal{A}}{\rightarrow} q_F & d(q_3, q_5) \underset{\mathcal{A}}{\rightarrow} q_1 & e(q_7) \underset{\mathcal{A}}{\rightarrow} q_3 & g \underset{\mathcal{A}}{\rightarrow} q_7 \\
a(q_2) \underset{\mathcal{A}}{\rightarrow} q_F & d(q_4, q_5) \underset{\mathcal{A}}{\rightarrow} q_1 & e(q_8) \underset{\mathcal{A}}{\rightarrow} q_4 & h \underset{\mathcal{A}}{\rightarrow} q_8 \\
b(q_1) \underset{\mathcal{A}}{\rightarrow} q_F & d(q_3, q_6) \underset{\mathcal{A}}{\rightarrow} q_1 & f(q_9) \underset{\mathcal{A}}{\rightarrow} q_5 & i \underset{\mathcal{A}}{\rightarrow} q_9 \\
c \underset{\mathcal{A}}{\rightarrow} q_1 & d(q_4, q_6) \underset{\mathcal{A}}{\rightarrow} q_2 & f(q_{10}) \underset{\mathcal{A}}{\rightarrow} q_6 & j \underset{\mathcal{A}}{\rightarrow} q_{10}
\end{array}$$

where q_F is the unique final state. From this definition, we establish that $\mathcal{L}_{\mathcal{A}''}$ is the finite language corresponding to the following set of terms:

$$\mathcal{L}_{\mathcal{A}''} = \{a(c), b(c), a(d(e(g), f(i))), a(d(e(h), f(i))), a(d(e(g), f(j))),$$

$$a(d(e(h), f(j))), b(d(e(g), f(i))), b(d(e(h), f(i))), b(d(e(g), f(j)))\}.$$

Let suppose that a teacher constructs the representative sample:

$$\mathcal{E} = \{b(c), a(d(e(g), f(i))), a(d(e(h), f(i))),$$

$$a(d(e(g), f(j))), a(d(e(h), f(j)))\}.$$

The learner Altex starts with the construction of $\mathsf{F} = \mathcal{E}[\diamond]$ and $T = T_{\mathcal{L}_{\mathcal{A}''}}(\mathcal{S}(\mathcal{E}), \mathsf{F})$.

Altex now notices the three problematics pairs of terms

$$d(e(h), f(i)) \text{ and } d(e(h), f(j)),$$

$$d(e(g), f(i)) \text{ and } d(e(h), f(j))$$

and

$$d(e(g), f(j)) \text{ and } d(e(h), f(j)).$$

Indeed

$$\text{row}(e(g)) = \text{row}(e(h)) \text{ and } \text{row}(f(i)) = \text{row}(f(j))$$

but

$$b(d(e(g), f(i))) \in \mathcal{L}, b(d(e(g), f(j))) \in \mathcal{L}, b(d(e(h), f(i))) \in \mathcal{L}$$

and

$$b(d(e(h), f(j))) \notin \mathcal{L} \text{ (Figure 2).}$$

Altex adds

$$b(d(e(g), \diamond)),$$
$$b(d(e(h), \diamond)),$$
$$b(d(\diamond, f(i)))$$

and

$$b(d(\diamond, f(j)))$$

	$\ldots$ $b(\diamond)$ $\ldots$
$\vdots$	
$d(e(g), f(i))$	010011 $\ldots$ 1 $\ldots$ 100110
$d(e(g), f(j))$	010011 $\ldots$ 1 $\ldots$ 100110
$d(e(h), f(i))$	010011 $\ldots$ 1 $\ldots$ 100110
$d(e(h), f(j))$	010011 $\ldots$ 0 $\ldots$ 100110
$\vdots$	
$e(g)$	001000 $\ldots$ 0 $\ldots$ 000101
$e(h)$	001000 $\ldots$ 0 $\ldots$ 000101
$\vdots$	
$f(i)$	000100 $\ldots$ 0 $\ldots$ 001010
$f(j)$	000100 $\ldots$ 0 $\ldots$ 001010
$\vdots$	

Fig. 2.

to F and complete the table T with help of the teacher. The new contexts $b(d(\diamond, f(j)))$ and $b(d(e(h), \diamond))$ respectively, separate the rows of $e(g)$ and $e(h)$ and the rows of ($f(i)$ and $f(j)$(Figure 3).

ALTEX now finds out that

$$b(d(e(h), f(i))) \in \mathcal{L}, b(d(e(h), f(j))) \notin \mathcal{L} \text{ but } \text{row}(i) = \text{row}(j)$$

and that

$b(d(e(g), f(j))) \in \mathcal{L}, b(d(e(h), f(j))) \notin \mathcal{L}$ but $\text{row}(g) = \text{row}(h)$ (Figure 4).

The new contexts

$$b(d(e(\diamond), f(j)))$$

and

$$b(d(e(h), f(\diamond)))$$

are added into F and the table T is completed one last time with the help of the teacher (Figure 5). The rows of i and j and the rows of g and h are separated.

	$\ldots$	$b(\diamond)$	$\ldots$	$b(d(e(g),\diamond))$	$b(d(e(h),\diamond))$	$b(d(\diamond,f(i)))$	$b(d(\diamond,f(j)))$
$\vdots$							
$d(e(g),f(i))$	$010011\ldots$	$1\ldots$	$100110\ldots$	0	0	0	0
$d(e(g),f(j))$	$010011\ldots$	$1\ldots$	$100110\ldots$	0	0	0	0
$d(e(h),f(i))$	$010011\ldots$	$1\ldots$	$100110\ldots$	0	0	0	0
$d(e(h),f(j))$	$010011\ldots$	$0\ldots$	$100110\ldots$	0	0	0	0
$\vdots$							
$e(g)$	$001000\ldots$	$0\ldots$	$000101\ldots$	0	0	1	1
$e(h)$	$001000\ldots$	$0\ldots$	$000101\ldots$	0	0	1	0
$\vdots$							
$f(i)$	$000100\ldots$	$0\ldots$	$001010\ldots$	1	1	0	0
$f(j)$	$000100\ldots$	$0\ldots$	$001010\ldots$	1	0	0	0
$\vdots$							

Fig. 3.

	$\ldots$	$b(d(e(h),\diamond))$	$\ldots$	$b(d(\diamond,f(j)))$
$\vdots$				
$e(g)$	011100	$\ldots 0$	$\ldots$	$1\ldots$
$e(h)$	011100	$\ldots 0$	$\ldots$	$0\ldots$
$f(i)$	011100	$\ldots 1$	$\ldots$	$0\ldots$
$f(j)$	011100	$\ldots 0$	$\ldots$	$0\ldots$
$\vdots$				
g	000010	$\ldots 0$	$\ldots$	$0\ldots$
h	000010	$\ldots 0$	$\ldots$	$0\ldots$
$\vdots$				
i	000001	$\ldots 0$	$\ldots$	$0\ldots$
j	000001	$\ldots 0$	$\ldots$	$0\ldots$
$\vdots$				

Fig. 4.

	...	$b(d(e(h),\diamond))$	...	$b(d(\diamond,f(j)))$	$b(d(e(\diamond),f(j)))$	$b(d(e(h),f(\diamond)))$
$\vdots$						
$e(g)$	011100	...0	...	1...	0	0
$e(h)$	011100	...0	...	0...	0	0
$f(i)$	011100	...1	...	0...	0	0
$f(j)$	011100	...0	...	0...	0	0
$\vdots$						
g	000010	...0	...	0...	1	0
h	000010	...0	...	0...	0	0
$\vdots$						
i	000001	...0	...	0...	0	1
j	000001	...0	...	0...	0	0
$\vdots$						

Fig. 5.

T is now consistent and ALTEX output $\mathcal{A}_T$ which verify $\mathcal{L}_{\mathcal{A}_T} = \mathcal{L}_{\mathcal{A}''}$.

References

[1] D. Angluin. Inductive inference of formal langage from positive data. *Information and Control*, 45:117–135, 1980.

[2] D. Angluin. A note on the number of queries needed to identify regular languages. *Information and Control*, 51:76–87, 1981.

[3] D. Angluin. Learning regular sets from queries and counter examples. *Information and Control*, 75:87–106, 1987.

[4] D. Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988.

[5] J. Besombes. *Un modèle algorithmique de la généralisation de structures dans le processus d'acquisition du langage*. PhD thesis, Université Nancy 1 - UHP, 2003.

[6] J. Besombes and J-Y Marion. Apprentissage de langages réguliers d'arbres et applications. *Traitement Automatique des Langues, Hermès*, 44(1):121–153, 2003.

[7] J. Besombes and J.Y. Marion. Identification of reversible dependency tree languages. *Proceedings of the third Learning Language in Logic workshop*, pages 11–22, 2001.

[8] J. Besombes and J.Y. Marion. Apprentissage des langages réguliers d'arbres et applications. *Conférence d'Apprentissage, Orléans 17, 18 et 19 juin 2002*, pages 55–70, 2002.

[9] J. Besombes and J.Y. Marion. Learning regular tree languages and applications. *Quatrièmes Rencontres de l'Informatique Messine*, pages 169–180, 2003.

[10] J. Besombes and J.Y. Marion. Learning dependency languages from a teacher. In *Proceedings of Formal Grammar 2004*, pages 17–28, 2004.

[11] J. Besombes and J.Y. Marion. Learning reversible categorial grammars from structures. In *Proceedings of the international IIS:IIPWM'04*, Advances in Soft Computing, Springer Verlag, pages 181–190, 2004.

[12] R.C. Carrasco, J. Oncina, and J. Calera. Stochastic inference of regular tree languages. *Lecture Notes in Computer Science*, 1433:185–197, 1998.

[13] N. Chomsky. *Knowlege of Language*. Praeger, New York, 1986.

[14] A. Christophe. L'apprentissage du langage. In *Université de tous les savoirs*, volume 2, pages 41–51. Odile Jacob, 2000.

[15] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.

[16] A. Dikovsky and L. Modina. Dependencies on the other side of the curtain. *Traitement automatique des langues*, 41(1):67–96, 2000.

[17] F. Drewes and J. Högberg. Learning a regular tree language from a teacher. In Z. Ésik and Z. Fülöp, editors, *Proc. Devel-*

opments in Language Theory 2003, volume 2710 of *Lecture Notes in Computer Science*, pages 279–291. Springer, 2003.

[18] F. Drewes and J. Högberg. Learning a regular tree language from a teacher even more efficiently. Technical Report 03.11, UmeaUniversity, 2003.

[19] H. Fernau. Learning tree languages from text. 2375:153–168, 2002.

[20] H. Fukuda and K. Kamata. Inference of tree automata from sample set of trees. *International Journal of Computer and Information Sciences*, 13(3):177–196, 1984.

[21] P. García. Learning k-testable tree sets from positive data. *Technical Report DSIC/II/46/1993, Departamento de Sistemas Informáticos i Computación, Universidad Politécnica de Valencia*, 1993.

[22] M.E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[23] K. Kamata. Inference methods for tree automata from sample set of trees. *IEEE International Conference on Systems, Man and Cybernetics*, pages 490–493, 1988.

[24] M. Kanazawa. *Learnable classes of Categorial Grammars.* CSLI, 1998.

[25] D. Kozen. On the myhill-nerode theorem for trees. *EATCS Bulletin*, 47:170–173, June 1992.

[26] D. López, J.M. Sempere, and P. García. Inference of reversible tree languages. *IEEE Transactions on Systems, Man and Cybernetics*, 34(4):1658–1665, 2004.

[27] S. Pinker. *The language instinct.* Harper, 1994.

[28] Y. Sakakibara. Inductive inference of logic programs based on algebraic semantics. Technical Report ICOT, 79, TR-260, 1987.

[29] Y. Sakakibara. Inferring parsers of context-free languages from structural examples. Technical Report ICOT, 81, TR-330, 1987.

[30] Y. Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76:223–242, 1990.