Identification of biRFSA languages ⋆

Michel Latteux [a], Aurélien Lemay [b], Yves Roos [a] and Alain Terlutte [b]

[a] *Laboratoire d'Informatique Fondamentale de Lille, U.M.R. C.N.R.S. 8022*
Université de Lille 1, 59655 Villeneuve d'Ascq CEDEX, FRANCE
{*michel.latteux,yves.roos*}*@lifl.fr*

[b] *Équipe Grappa–EA 3588, Université de Lille 3,*
Domaine universitaire du "Pont de bois",
BP 149, 59653 Villeneuve d'Ascq CEDEX, FRANCE
{*aurelien.lemay,alain.terlutte*}*@univ-lille3.fr*

Abstract

The task of identifying a language from a set of its words is not an easy one. For instance, it is not feasible to identify regular languages in the general case. Therefore, looking for subclasses of regular languages that can be identified in this framework is an interesting problem. One of the most classical identifiable classes is the class of reversible languages, introduced by D. Angluin, also called bideterministic languages as they can be represented by deterministic automata (DFA) whose reverse is also deterministic. Residual Finite State Automata (RFSA) on the other hand is a class of non deterministic automata that shares some properties with DFA. In particular, DFA are RFSA and RFSA can be much smaller. We study here learnability of the class of languages that can be represented by biRFSA: RFSA whose reverse are RFSA. We prove that this class is not identifiable in general but we present two subclasses that are learnable, the second one being identifiable in polynomial time.

1 Introduction

We consider here learning algorithms in the identification in the limit from positive examples framework [8]. Informally, a family of languages is identifiable in the limit from positive examples if there exists an algorithm which is able to *identify* any language of this class provided *enough* examples of the language have been observed. To prove that a family of language is identifiable

⋆ This research was partially supported by Inria (MOSTRARE team).

in the limit from positive example, we use the following stronger characterization, where the term *finite representation of a language* refers to any classical representation used in language theory like grammars or automata.

Definition 1 *A family of languages $\mathcal{L} \subset 2^{\Sigma^*}$ is identifiable in the limit if there exists an algorithm M that takes as an input a finite set of words $S \subset \Sigma^*$ and produces as output a finite representation of a language such that for every language $L \in \mathcal{L}$, there exists a sample $S_L \subseteq L$, called the characteristic sample of L for M such that for every S with $S_L \subseteq S \subseteq L$, $M(S)$ is a finite representation of L.*

As defined here, identifiability from positive examples is a property that is hard to obtain. Any family of languages that contains every finite language and at least one infinite language is not identifiable in this framework [8], which implies the non-identifiability of regular languages as a whole. Nevertheless, some non-trivial families of regular languages are identifiable this way: reversible languages (called here bideterministic [1], as they can be represented by a deterministic automaton whose reverse is also deterministic) and k-testable languages [7] most notably. Researches were undertaken to extend those results to greater classes of regular languages: function distinguishable languages [6], disjoint prime residual languages [4], strictly deterministic languages [15] among others. Also some classes of context-free languages [13,9] have been proved learnable in this context.

If we consider an automaton as a representation for a target regular language, determinism seems to be an essential property for learning algorithms here. To identify a language, a study of suffixes that can follow a prefix is usually done, which in fact is equivalent to a study of residual languages of words in the sample. The main property that allows bideterministic languages to be identifiable is a property satisfied by their residual languages: bideterministic languages are languages whose residual languages are disjoint. The algorithm can therefore state that two prefixes are equivalent if their residual languages (in the sample) are not disjoint. To identify classes of regular languages, we will therefore focus our attention on residual languages.

Residual Finite State Automata (RFSA) have been introduced in [3]. An automaton is an RFSA if each of its states corresponds to a residual of the language it recognizes. This is a property of deterministic automata, but also shared by non deterministic ones. Some properties of RFSA (in the learning context) have been investigated in [5].

BiRFSA, introduced in [11], are RFSA whose reverse is also a RFSA. Since biRFSA are a natural generalization of bideterministic automata, one could hope to produce a learning algorithm for the family of biRFSA languages. Unfortunately, biRFSA languages are not identifiable as they contain a class

of languages which is proved to be not identifiable in [4]. Our purpose here is to find restrictions to the class of biRFSA languages that would define learnable classes.

Bideterministic languages have disjoint residual languages. Natural restrictions to biRFSA languages could by consequence be RFSA languages whose prime residual languages are disjoint, biRFSA languages whose residuals are without inclusions or biRFSA languages without composite. RFSA languages with disjoint prime residuals are proved to be non learnable in [4]. BiRFSA languages that have no inclusions of residuals have been introduced in [11] and are called *biseparable languages*; they have the property that their canonical RFSAs are the unique minimal NFAs that recognize them. But this class of languages also contains the class of languages with disjoint prime residuals, and is therefore not learnable. To finish, languages $L_n = \Sigma^{\leq n}$ and $L_* = \Sigma^*$ are biRFSA languages without composite, so this class is not identifiable either (due to a property of [8]).

The aim of this article is to present two families of biRFSA identifiable from positive examples. The second one is also identifiable polynomially in the sense that the learning algorithm answers in polynomial time and that the size of the characteristic sample of a language is a polynomial in the size of its canonical RFSA. In both cases, the learning algorithms aim to identify the canonical RFSA, which is a non deterministic automaton that can be much smaller than the corresponding minimal DFA.

2 Preliminaries

Let us recall the definition of residuals of a language: let Σ be an alphabet and $L \subseteq \Sigma^*$ be a language. A language $\mathsf{Post} \subseteq \Sigma^*$ is a *right residual* of L if there exists a word $u \in \Sigma^*$ such that $\mathsf{Post} = \{v \in \Sigma^* \mid uv \in L\}$, that is denoted $\mathsf{Post} = u^{-1}L$. Symmetrically is defined the notion of *left residual*: a language Pre is a left residual of L if there exists a word $v \in \Sigma^*$ such that $\mathsf{Pre} = \{u \in \Sigma^* \mid uv \in L\}$, that is denoted $\mathsf{Pre} = Lv^{-1}$. It is well known that a language is recognizable if and only if it has a finite number of residuals. In order to precise the link between residuals of a recognizable language and the states of automata which recognize it, let us introduce the following notation: let $A = \langle \Sigma, Q, I, F, \delta \rangle$ be a finite non deterministic automaton (NFA). For any state $q \in Q$, we define $\mathsf{Post}_{A,q}$, the *right* language of q, by $\mathsf{Post}_{A,q} = \{v \in \Sigma^* \mid \delta(q, v) \cap F \neq \emptyset\}$, and we define $\mathsf{Pre}_{A,q}$, the *left* language of q, by $\mathsf{Pre}_{A,q} = \{u \in \Sigma^* \mid q \in \delta(I, u)\}$. When there is no ambiguity on the used automaton, we shall just write Post_q for $\mathsf{Post}_{A,q}$ and Pre_q for $\mathsf{Pre}_{A,q}$.

The *reverse* of a word $u \in \Sigma^*$ is denoted u^R and is defined inductively by: $\varepsilon^R =$

ε, and $\forall u \in \Sigma^*, \forall x \in \Sigma, (ux)^R = x(u^R)$. Then this definition is extended to languages: if L is a language, then $L^R = \{u^R \mid u \in L\}$. Let $A = \langle \Sigma, Q, I, F, \delta \rangle$ be an automaton. Then the reverse of A is the automaton $A^R = \langle \Sigma, Q, F, I, \delta^R \rangle$ where $\delta^R = \{(q, x, q') \mid (q', x, q) \in \delta\}$. It is well known that an automaton A recognizes a language L if and only if its reverse, A^R, recognizes L^R, the reverse of L.

In order to enumerate words, we shall use in the following a fixed ordered alphabet Σ (see for instance [2]), then we can use the alphabetic order over Σ^* defined by: $\forall u, v \in \Sigma^*, u < v$ if and only if $|u| < |v|$ or there exist $w, u', v' \in \Sigma^*$ and two letters $x < y \in \Sigma$ such that $|u'| = |v'|$ and $u = wxu', v = wyv'$. We denote $<^R$ the order in the reverse, i.e. $u <^R v \Leftrightarrow u^R < v^R$. Then it directly follows the notion of smallest word (for $<$) of a language L that we denote by $\min(L)$ and the smallest word for $<^R$ denoted by $\min^R(L)$

We also define an ordering over $\Sigma^* \times \Sigma^*$: $[u, v] < [\bar{u}, \bar{v}]$ iff

$$(|uv| < |\bar{u}\bar{v}|) \text{ or } (|uv| = |\bar{u}\bar{v}| \text{ and } u < \bar{u}) \text{ or } (u = \bar{u} \text{ and } v^R < \bar{v}^R)$$

3 RFSA and biRFSA languages

If we consider any trim deterministic automaton $A = \langle \Sigma, Q, \{q_0\}, F, \delta \rangle$, it is clear that, for any state q in Q, the language Post_q is a residual of the language recognized by A. Moreover it is well known that the set of states of the minimal deterministic automaton of any recognizable language L is isomorphic to the set of right residuals of L. This fine property is not satisfied by non deterministic automata: if $A = \langle \Sigma, Q, I, F, \delta \rangle$ is a non deterministic automaton, then for any state q in Q, the language Post_q is included in a right residual of the language recognized by A, but not always equal to it. This is the reason why the following notion has been introduced in [3]:

Definition 2 *A (non deterministic) automaton $A = \langle \Sigma, Q, I, F, \delta \rangle$ is a residual finite state automaton (RFSA for short) if for every state $q \in Q$, the language Post_q is a right residual of the language recognized by A.*

The notion of unique minimal deterministic automaton is essential, unfortunately there does not exist a similar notion for NFA. Nevertheless, such a canonical representation exists for the class of RFSA. Indeed it has been proved in [3] that every recognizable language can be recognized by a unique non deterministic *reduced* RFSA, called the canonical RFSA of the language. In order to give its definition, let us first introduce the notion of *prime* residual of a language.

Definition 3 *Let L be a language. A right residual of L is* prime *if it is non*

empty and if it cannot be obtained as the union of other right residuals of L.

In a similar way, one can define the notion of prime left residual.

Definition 4 *Let Σ be an alphabet and $L \subseteq \Sigma^*$ be a recognizable language. The canonical RFSA A of L is the automaton $A = \langle \Sigma, Q, I, F, \delta \rangle$ where*

- *Q is the set of right prime residuals of L,*
- *$I = \{s \in Q \mid s \subseteq L\}$,*
- *$F = \{s \in Q \mid \varepsilon \in s\}$,*
- *$\forall s \in Q, \forall x \in \Sigma, \delta(s, x) = \{s' \in Q \mid xs' \subseteq s\}$.*

The case when the reverse of a deterministic automaton is still deterministic leads to the class of 0-reversible languages ([1]) or bideterministic languages ([12],[14]) which have been studied in the context of machine learning, or in terms of minimal representation of recognizable languages. When RFSAs are considered, we define the notion of biRFSAs:

Definition 5 *An automaton A is a biRFSA if A is an RFSA and the reverse of A is also an RFSA. A language is a biRFSA language if there exists a biRFSA which recognizes it.*

Note that, as an equivalent definition, we can say that $A = \langle \Sigma, Q, I, F, \delta \rangle$, recognizing a language L, is a biRFSA if, for any state $q \in Q$, Post_q is a right residual of L and Pre_q is a left residual of L.

Let us give some results concerning biRFSA languages (for more informations and examples, see [11])

Proposition 6 *A recognizable language is a biRFSA language if and only if its canonical RFSA is a biRFSA.*

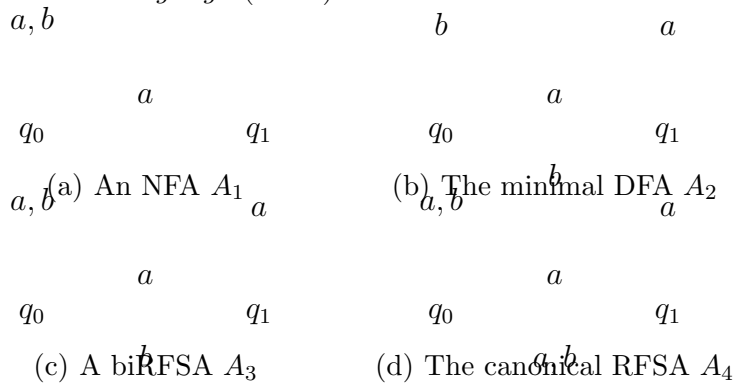
Thus the *canonical biRFSA* denotes the canonical RFSA of a biRFSA language.

Proposition 7 *A recognizable language L is a biRFSA language if and only if the reverse of its canonical RFSA is the canonical RFSA of the reverse of L.*

Lemma 8 *Let $A = \langle \Sigma, Q, I, F, \delta \rangle$ be a canonical biRFSA recognizing a language L. Then for any state $q \in Q$, there exist words $u_q \in \mathsf{Pre}_q$ and $v_q \in \mathsf{Post}_q$ such that $\mathsf{Post}_q = u_q^{-1}L$ and $\mathsf{Pre}_q = Lv_q^{-1}$. The word u_q (resp. v_q) is called an* incoming *(resp.* outgoing*)* characteristic *word of state q.*

Proposition 9 *The canonical biRFSA of a biRFSA language L is a minimal NFA for L.*

Example 10 *Let us consider the four automata of figure 1, each of them recognizing the same language* $(a+b)^*a$*:*



(a) An NFA A_1 (b) The minimal DFA A_2

(c) A biRFSA A_3 (d) The canonical RFSA A_4

Fig. 1. Some automata for $(a+b)^*a$.

The right residuals of $(a+b)^*a$ *are the two languages* $(a+b)^*a$ *and* $(a+b)^*a+\varepsilon$*. The non empty left residuals of* $(a+b)^*a$ *are the two languages* $(a+b)^*a$ *and* $(a+b)^*$*.*

Automaton A_1 *is not an RFSA because* $\mathsf{Post}_{A_1,q_1} = \{\varepsilon\}$ *which is not a right residual of* $(a+b)^*a$*. Automaton* A_2 *is an RFSA since it is the minimal DFA of* $(a+b)^*a$ *but it is not a biRFSA since* $\mathsf{Pre}_{A_2,q_0} = (a+b)^*b+\varepsilon$ *which is not a left residual of* $(a+b)^*a$*. Automaton* A_3 *is a biRFSA since now* $\mathsf{Pre}_{A_3,q_0} = (a+b)^*$*, but it is not the canonical RFSA of* $(a+b)^*a$*. Indeed,* $a\,\mathsf{Post}_{A_3,q_0} \subseteq \mathsf{Post}_{A_3,q_1}$ *but there is no transition* (q_1, a, q_0) *in automaton* A_3*. Finally, automaton* A_4 *is a biRFSA which is the canonical RFSA of* $(a+b)^*a$*.*

4 Definition of representative couples

The two families of biRFSA languages we shall define rely on the notion of representative couples which are the words used to represent the states in the canonical biRFSA. In a biRFSA, the states are associated with left and right residuals. Thus, we say that $[u, v]$ is a state of a biRFSA A if there exists a state q in automaton A such that $\mathsf{Post}_q = u^{-1}L$ and $\mathsf{Pre}_q = Lv^{-1}$. The lemma 8 will be used to prove that this notion coincides with the following definition.

Definition 11 *Let* L *be a biRFSA language. A couple* $[u, v]$ *is called a* characteristic couple *of* L *if* $(Lv^{-1})(u^{-1}L) \subseteq L$ *and* $uv \in L$*.*

Lemma 12 *Let* $A = \langle \Sigma, Q, I, F, \delta \rangle$ *be a canonical biRFSA recognizing a language* L*, then* $[u, v]$ *is a state of* A *iff* $[u, v]$ *is a characteristic couple of* L*.*

PROOF. Let $[u, v]$ be a state of A, then there exists $q \in Q$ such that $\mathsf{Post}_q = u^{-1}L$ and $\mathsf{Pre}_q = Lv^{-1}$. It directly follows that $(Lv^{-1})(u^{-1}L) \subseteq L$. Moreover, from lemma 8, there exists a word v_q such that $v_q \in \mathsf{Post}_q$ with $\mathsf{Pre}_q = Lv_q^{-1} = Lv^{-1}$. Then $uv_q \in L$ and, since $Lv_q^{-1} = Lv^{-1}$, we get that $uv \in L$.

Conversely, if $[u, v]$ is a characteristic couple of L, then $(Lv^{-1})(u^{-1}L) \subseteq L$ and $uv \in L$. Since $uv \in L$, there exists a state $q \in Q$ such that $u \in \mathsf{Pre}_q$ and $v \in \mathsf{Post}_q$. It follows that $\mathsf{Post}_q \subseteq u^{-1}L$ and $\mathsf{Pre}_q \subseteq Lv^{-1}$. For the reverse inclusions, we get from lemma 8 that there exist two words u_q and v_q such that $u_q \in \mathsf{Pre}_q \subseteq Lv^{-1}$ with $\mathsf{Post}_q = u_q^{-1}L$ and $v_q \in \mathsf{Post}_q \subseteq u^{-1}L$ with $\mathsf{Pre}_q = Lv_q^{-1}$. Since $(Lv^{-1})(u^{-1}L) \subseteq L$, it follows that $u_q(u^{-1}L) \subseteq L$, then $u^{-1}L \subseteq u_q^{-1}L = \mathsf{Post}_q$. In a similar way, we get $Lv^{-1} \subseteq Lv_q^{-1} = \mathsf{Pre}_q$. □

Using this property, one can represent each state q of a canonical biRFSA by a couple of words, that will be called the representative couple of state q.

Definition 13 *Let L be a biRFSA language. A couple $[u, v]$ is called a* representative couple *of L if*

(1) $[u, v]$ is a characteristic couple of L
(2) for any word $u' \neq u$, if $u'^{-1}L = u^{-1}L$ then $u < u'$
(3) for any word $v' \neq v$, if $Lv'^{-1} = Lv^{-1}$ then $v <^R v'$

Clearly, if L is a biRFSA language and $[u, v]$ is a characteristic couple of L, it follows from the definition of characteristic couples that for any word u' such that $u'^{-1}L = u^{-1}L$ and for any word v' such that $Lv'^{-1} = Lv^{-1}$, $[u', v']$ is a characteristic couple of L. Thus, for a biRFSA language L, we have a one-to-one correspondance between states of the canonical biRFSA recognizing L and the representative couples of L.

5 Identification from positive data of k-characteristic biRFSA languages

In order to have an identifiable subclass of biRFSA languages, one has to be able to find the representative couples of states of the biRFSA. One has to distinguish which couples are characteristic while examining other couples. We also have to confirm these characteristic couples when the sample is large enough. We define a property of biRFSA languages based on lengths of characteristic couples.

Definition 14 *A biRFSA language is* k-characteristic *if, for any couple $[u, v]$ such that $uv \in L$ and $[u, v]$ is not characteristic, there exist two words u_1 and v_2 such that $u_1 \in Lv^{-1}$, $v_2 \in u^{-1}L$, $u_1v_2 \notin L$ and $|u_1v_2| \leq |uv| + k$.*

Definition 14 does not bound the lengths of characteristic couples; that would mean to bound the size of the automaton. It defines a distance which allows us to confirm the choice of a characteristic couple.

The class of 0-characteristic languages strictly contains that of bideterministic languages. Indeed, if uv belongs to a bideterministic language, then $[u, v]$ is a characteristic couple. The language of figure 2 is a 0-characteristic language which is not a bideterministic one. Let us also remark that, for any biRFSA language L, there exists k such that L is k-characteristic.

Proposition 15 *Let $k \in \mathbb{N}$. The k-characteristic biRFSA languages are identifiable in the limit from positive data.*

PROOF. Let $A = \langle \Sigma, Q, I, F, \delta \rangle$ be the canonical RFSA of a k-characteristic biRFSA language L, Q being the set of representative couples. Let $m = 2 * p + k + 1$ where p is the length of the longest component of the representative couples. We define the characteristic sample by $S_L = L \cap \Sigma^{\leq m}$.

The algorithm is simple : it suffices to enumerate, with respect to the alphabetic ordering, the couples of factors deduced from the sample S and to test, according to the property of the language, whether the couple is characteristic. When a couple is characteristic, we have to test if there exists a previous characteristic couple which is equivalent. Two couples $[u, v]$ and $[u', v']$ are equivalent if they are characteristic and $uv' \in S$ and $u'v \in S$. If there does not exist an equivalent couple, this one is representative. Finally, we build the transitions : if $[u, v]$ and $[u', v']$ are two representative couples, there exists a transition $([u, v], x, [u', v'])$ if $uxv' \in S$.

Let PS be the set of couples deduced from S, that is $PS = \{[u, v] \mid uv \in S\}$. The algorithm is the following :

$[u, v]$:= *first couple in* PS
Q_{repr} := $\emptyset$
repeat
  if (($\nexists$ u_1 *and* v_2 *such that* $u_1 v \in S$,
        $uv_2 \in S$, $u_1 v_2 \notin S$ *and* $|u_1 v_2| \leq |uv| + k$)
    and ($\nexists$ $[u', v'] \in Q_{repr}$ *such that* $uv' \in S$ *and* $u'v \in S$)) **then**
      add $[u, v]$ to Q_{repr}
      build B *with sets of states* Q_{repr},
        $I = \{[u, v] \in Q_{repr} \mid v \in S\}$,
        $F = \{[u, v] \in Q_{repr} \mid u \in S\}$
        and transitions $\delta([u, v], a) = \{[u', v'] \in Q_{repr} \mid uav' \in S\}$
  end if
  $[u, v]$:= *following couple in* PS *in the alphabetic order*
until (B *consistent with the sample* S)

If the sample S contains the characteristic sample S_L, the property $Q_{repr} = \{[u', v'] \mid [u', v']$ is a representative couple and $[u', v'] < [u, v]\}$ is a loop invariant. It is then obvious that only representative couples and correct transitions are added until the automaton is consistent. Thus at any step of the algorithm, B is a subautomaton of A. And it is obvious that a subautomaton cannot be consistent if a state and its transitions are missing. $\square$

Of course, the characteristic sample is very large. It is defined so that Proposition 15 is obvious. The following examples show that a smaller sample is sufficient to identify k-characteristic biRFSA languages.

Example 16 *The language of Figure 2 is a* 0*-characteristic biRFSA language. But this language is not a bideterministic language. Its prime residual languages are not disjoint (neither the left nor the right ones). Additionally, one can verify that, for any* n*, the language* $\Sigma^* a^n \Sigma^* \setminus \Sigma^* a^{n+1} \Sigma^*$ *is a* 0*-characteristic biRFSA language that is not a bideterministic language. In this example,* $[\varepsilon, a]$ *and* $[a, \varepsilon]$ *are the two representative couples. Since* ε *does not belong to the language, these two couples are the first ones studied by the algorithm.*

The characteristic sample of this language is $S_L = L \cap \Sigma^{\leq 3}$ *but the algorithm only needs* $S'_L = a + ab + ba + aba$ *to build the canonical RFSA.*

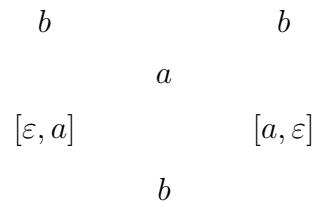


Fig. 2. The canonical RFSA of a 0-characteristic biRFSA language.

Example 17 *The automaton of Figure 3 is a canonical RFSA which recognizes a* 1*-characteristic biRFSA language. The characteristic sample of this language is* $S_L = L \cap \Sigma^{\leq 4}$ *but the algorithm only needs* $S'_L = a + aa + ab + b + bb + bba$ *to build the canonical RFSA.*

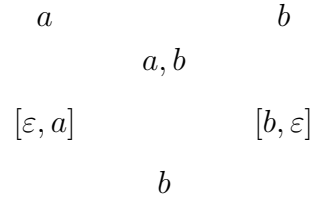


Fig. 3. The canonical RFSA of a 1-characteristic biRFSA language.

$[u, v] \in PS$	*state*	u_1, v_2 *such that* $u_1 v \in S$, $uv_2 \in S$, $u_1 v_2 \notin S$
$[\varepsilon, a]$	$[\varepsilon, a]$	
$[\varepsilon, b]$		$u_1 = b$, $v_2 = a$
$[a, \varepsilon]$		$u_1 = b$, $v_2 = a$
$[b, \varepsilon]$	$[b, \varepsilon]$	

When $Q = \{[\varepsilon, a], [b, \varepsilon]\}$, *we have the transitions*

$u_1 x v_2 \in S$	*transition* $([u_1, v_1], x, [u_2, v_2])$
aa	$([\varepsilon, a], a, [\varepsilon, a])$
a	$([\varepsilon, a], a, [b, \varepsilon])$
b	$([\varepsilon, a], b, [b, \varepsilon])$
bba	$([b, \varepsilon], b, [\varepsilon, a])$
bb	$([b, \varepsilon], b, [b, \varepsilon])$

Example 18 *Let* $\Sigma = \{a, b\}$. *We could verify that every language* $L_n = \Sigma^* a \Sigma^n$, $n \in \mathbb{N}$, *is an n-characteristic biRFSA language. The characteristic sample of the language* $L_2 = \Sigma^* a \Sigma^2$ *is* $S_{L_2} = L_2 \cap \Sigma^{\leq 9}$ *but the algorithm only needs* $S'_{L_2} = L_2 \cap \Sigma^{\leq 5}$.

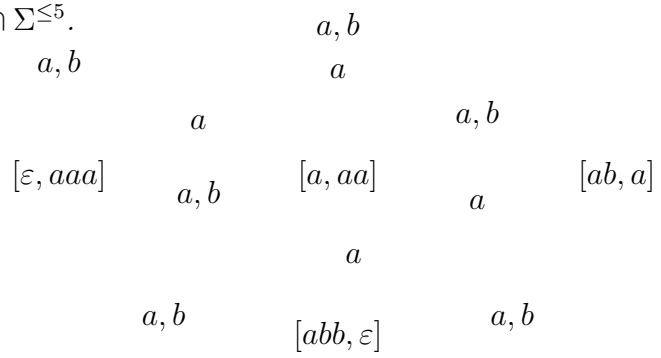


Fig. 4. The canonical RFSA of $\Sigma^* a \Sigma^2$ which is 2-characteristic.

6 Polynomial identification of ordered biRFSA languages

In this section we define a family of biRFSA languages which is polynomially identifiable in the limit from positive data: a family of languages is polynomially identifiable if the learning algorithm is in PTIME and the size of the characteristic sample of a language in the family is polynomially bounded on the size of the automaton computed by the learning algorithm. This family,

which has a quite simple definition, appreciably generalizes the family of bideterministic languages. For bideterministic languages, every couple $[u, v]$ such that $uv \in L$ is a characteristic couple. For the family we shall define, we require that the smallest words of prime residuals (*w.r.t.* alphabetic order) can be associated to constitute characteristic couples. This leads to the following notion of ordered biRFSA languages:

Definition 19 *A biRFSA language L is* ordered *if, for each representative couple $[u, v]$, we have $u = \min(Lv^{-1})$ and $v = \min^R(u^{-1}L)$.*

Example 20 *Let $\Sigma = \{a, b\}$ with the usual order $a < b$ and let us consider the language $L = \{a^2b, aba, ab^2, b^3\}$. It is a biRFSA which is recognized by the canonical RFSA of the figure 5. It is not an ordered biRFSA since $[b, b^2]$ is a representative couple of L but $b \neq \min(L(b^2)^{-1}) = a$.*

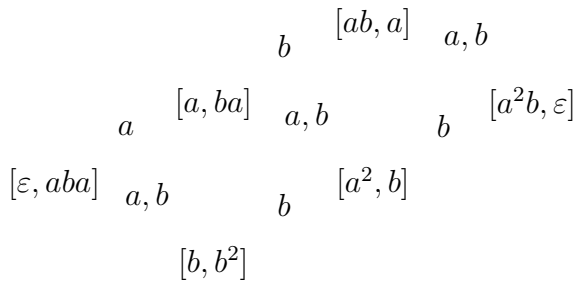


Fig. 5. The canonical RFSA recognizing $L = \{a^2b, aba, ab^2, b^3\}$.

Let us first precise some properties of the canonical RFSA of an ordered biRFSA language.

Definition 21 *For any language $L \subseteq \Sigma^*$, we denote by $Q(L)$ the set $Q(L) = \{[u, v] \mid uv \in L, v = \min^R(u^{-1}L), u = \min(Lv^{-1})\}$.*

Then we can prove:

Proposition 22 *Let $A = \langle \Sigma, Q, I, F, \delta \rangle$ be the canonical RFSA of an ordered biRFSA language L. Then*

- $Q = Q(L)$
- $I = \{[\varepsilon, v_0] \mid v_0 = \min^R(L)\}$
- $F = \{[u_0, \varepsilon] \mid u_0 = \min(L)\}$
- $\delta([u, v], x) = \{[u', v'] \in Q \mid uxv' \in L\}$.

PROOF. It is easily seen that $Q = Q(L)$. Let $[u, v] \in I$. Then $u^{-1}L \subseteq L$ and so $v \in L$. Hence $u = \min(Lv^{-1}) = \varepsilon$ and $v = \min^R(u^{-1}L) = \min^R(L)$. Let $[u, v] \in F$. Then $\varepsilon \in u^{-1}L$ and so $v = \varepsilon$. Hence $u = \min(Lv^{-1}) = \min(L)$. At last, let us consider $q, q' \in Q$ with $q = [u, v], q' = [u', v']$ such that $q' \in \delta(q, x)$. Then $x\,\mathsf{Post}_{q'} \subseteq \mathsf{Post}_q$ and so $xv' \in \mathsf{Post}_q = u^{-1}L$. Hence $uxv' \in L$. For the

reverse inclusion, let us take $q = [u, v], q' = [u', v'] \in Q$ with $uxv' \in L$. Then $ux \in Lv'^{-1} = \mathsf{Pre}_{q'}$. At last, since $\mathsf{Pre}_{q'}\,\mathsf{Post}_{q'} \subseteq L$, we get $ux\,\mathsf{Post}_{q'} \subseteq L$, $x\,\mathsf{Post}_{q'} \subseteq u^{-1}L = \mathsf{Post}_q$ hence $q' \in \delta(q, x)$. □

Example 23 *Let $\Sigma = \{a, b\}$ with $a < b$ and let us consider the language $L = (a + b)^*b(a + b)$. It is an ordered biRFSA which is recognized by the canonical RFSA of the figure 6. We have $F = \{[ba, \varepsilon]\}$ since $ba = \min(L)$, $I = \{[\varepsilon, ba]\}$ since $ba = \min^R(L)$, and $Q(L) = \{[\varepsilon, ba], [b, a], [ba, \varepsilon]\}$. For instance, $[b, a]$ is in $Q(L)$ because $ba \in L$, $b = \min(La^{-1})$ and $a = \min^R(b^{-1}L)$. It is a representative couple of L: $b^{-1}L$ is a prime right residual and La^{-1} is a prime left residual. There exists a transition from $[b, a]$ to $[\varepsilon, ba]$ labeled by b because $bbba \in L$.*

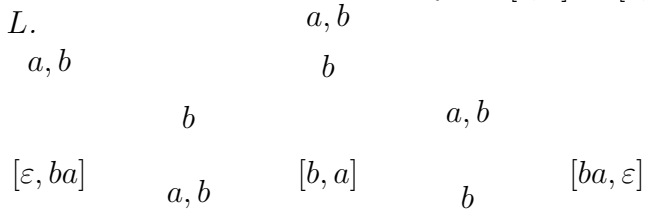


Fig. 6. The canonical RFSA recognizing $L = (a + b)^*b(a + b)$

.

Observe that, in this example, the set $\varepsilon+b+ba$ of first components of elements of $Q(L)$ is prefix-closed. This is always true for ordered biRFSA languages:

Lemma 24 *Let L be an ordered biRFSA language over Σ. Then $Q(L)$ satisfies*

$$\forall [u'u'', v] \in Q(L), \exists v' \text{ s.t. } [u', v'] \in Q(L) \quad (1)$$
$$\forall [u, v''v'] \in Q(L), \exists u' \text{ s.t. } [u', v'] \in Q(L) \quad (2)$$

PROOF. We shall prove property (1), the proof is symmetric for property (2). Clearly it is sufficient to consider only the case when u'' is a letter x. Let A be the canonical biRFSA recognizing L. Let $[u'x, v]$ be a representative couple of A. The word $u'x$ belongs to $\mathsf{Pre}_{[u'x,v]}$. There then exists a state $[\bar{u}', v']$ such that $u' \in \mathsf{Pre}_{[\bar{u}',v']}$ and there exists a transition $([\bar{u}', v'], x, [u'x, v])$. Let us suppose that $\bar{u}' \neq u'$. Since the language is ordered, we must have $\bar{u}' < u'$ and so $\bar{u}'x < u'x$. But $\bar{u}'x \in \mathsf{Pre}_{[u'x,v]}$ and that contradicts $u'x = \min(Lv^{-1})$. □

We are now able to give the definition of the characteristic sample of an ordered biRFSA language L. An important property is that the cardinality of this characteristic sample is polynomially bounded on the cardinality of $Q(L)$.

Definition 25 *Let L be an ordered biRFSA language. We define the* characteristic sample *of L, $S_L = \{uv \mid [u, v] \in Q(L)\} \cup \{uxv' \mid [u, v] \in Q(L), [u', v'] \in Q(L), uxv' \in L\}$.*

For any set S such that $S_L \subseteq S \subseteq L$, it is easy to prove $Q(L) \subseteq Q(S)$ but one does not always have equality. Indeed property (1) of lemma 24 does not necessarily hold for S as shown in the following example:

Example 26 *Let $\Sigma = \{a, b\}$ with $a < b$ and let us consider $L = \Sigma^4$. Then $Q(L) = \{[\varepsilon, a^4], [a, a^3], [a^2, a^2], [a^3, a], [a^4, \varepsilon]\}$ and L is an ordered biRFSA. The characteristic sample of L is $S_L = a^4+ba^3+aba^2+a^2ba+a^3b$ but if we consider $S = S_L + b^4$ we get $Q(S) = Q(L) + [b^2, b^2]$.*

Nevertheless we can state:

Lemma 27 *Let L be an ordered biRFSA language and S_L be its characteristic sample. Then for any set S such that $S_L \subseteq S \subseteq L$, $Q(L)$ is equal to $Q'(S)$, the largest subset of $Q(S)$ satisfying property (1) of lemma 24 and $A = \langle \Sigma, Q, I, F, \delta \rangle$, the canonical RFSA of L satisfies :*

- $Q = Q'(S)$
- $I = \{[\varepsilon, v_0] \mid v_0 = \min^R(S)\}$
- $F = \{[u_0, \varepsilon] \mid u_0 = \min(S)\}$
- $\delta([u, v], x) = \{[u', v'] \in Q \mid uxv' \in S\}$.

PROOF. Clearly, from proposition 22 and since $S_L \subseteq S$, we have only to prove $Q(L) = Q'(S)$. Let us first prove the inclusion $Q(L) \subseteq Q'(S)$. If $[u, v] \in Q(L)$ then $uv \in S_L \subseteq S$ and, since $S \subseteq L$, we get $[u, v] \in Q(S)$. Since $Q(L)$ satisfies property (1) of lemma 24, we obtain that $Q(L) \subseteq Q'(S)$. For the reverse inclusion, let us take $[u, v] \in Q'(S)$. We shall reason by induction on $|u|$: if $u = \varepsilon$, then $\min^R(u^{-1}S) = v_0 = \min^R(L)$, and $[\varepsilon, v_0] \in Q(L)$. Let us now consider $[u'x, v] \in Q'(S)$ with $x \in \Sigma$. From induction hypothesis, we have $[u', v'] \in Q(L)$ for some $v' \in \Sigma^*$. Since $u'xv \in L$, there exists $[u'', v''] \in Q(L)$ such that $u'x \in \mathsf{Pre}_{[u'', v'']}$ and $v \in \mathsf{Post}_{[u'', v'']}$. Since $[u'', v''] \in Q(L)$, we get $u''v'' \in S_L \subseteq S$. Since $u'xv'' \in L$ with $[u', v'] \in Q(L)$ and $[u'', v''] \in Q(L)$, we get $u'xv'' \in S_L \subseteq S$. It follows that $v \leq^R v''$ because $v = \min^R((u'x)^{-1}S)$, moreover since $u''v'' \in L$ and $u''v \in L$, we have $v'' \leq^R v$, hence $v = v''$. Similarly, $u'xv'' \in L$ and $u'' = \min(Lv''^{-1})$ imply $u'' \leq u'x$, moreover $u''v \in S$ and $u'x = \min(Sv^{-1})$ imply $u'x \leq u''$. So $u'' = u'x$ and $[u'x, v] = [u'', v''] \in Q(L)$. $\square$

We are now able to state the main result of this section:

Proposition 28 *The ordered biRFSA languages are identifiable in the limit from positive data with a PTIME algorithm* `M` *which builds the canonical RFSA of its target. Moreover, the size of the characteristic sample of every ordered biRFSA language L for* `M` *is polynomially bounded on the size of the canonical RFSA of L.*

PROOF. It is clear that the automaton given in lemma 27 can be built in PTIME from the sets S and $Q'(S)$. It remains to give a polynomial algorithm which takes as input a finite set S and gives $Q'(S)$ as output. Let $P = \text{pref}(S)$ be the set of prefixes of S. If we enumerate P with respect to the alphabetic order, we will find the first components of elements of $Q(S)$. If we store the corresponding second component in a set R, we can check if an element of P is really a first component of $Q'(S)$: it must not correspond to a second component that has already been stored in R. Moreover if we find such an element u of P which is not the first component of an element of $Q'(S)$, it follows from property (1) of lemma 24 that we can remove from P all the words of $u\Sigma^+$. This leads to the following algorithm:

P := pref(S)
$Q := \emptyset$
$R := \emptyset$
`while` $P \neq \emptyset$ `loop /* invariant:` $Q = Q'(S) \setminus \{[u, v] \mid u \in P\}$`*/`
  $u :=$ min(P)
  $v :=$ $\min^R(u^{-1}S)$
  `if` $v \in R$ `then /*` u `is not a first component */`
    $P := P \setminus u\Sigma^+$
  `else /*` $u = \min(Sv^{-1})$ `*/`
    $Q := Q \cup \{[u, v]\}$
    $R := R \cup \{v\}$
  `end if`
  $P := P \setminus \{u\}$
`end loop`
`return` Q

Let us prove that if there exists an ordered biRFSA language L such that $S_L \subseteq S \subseteq L$ then this algorithm computes $Q'(S)$. We shall denote by I the property $Q = Q'(S) \setminus \{[u, v] \mid u \in P\}$. This property is clearly true at the beginning of the loop. Moreover, property I is a loop invariant: if $u_0 = \min(P)$ is not a first component of an element of $Q'(S)$ then $Q'(S) \cap (u_0\Sigma^+ \times \Sigma^*) = \emptyset$ and $Q = Q'(S) \setminus \{[u, v] \mid u \in P \setminus \{u_0\}\}$, otherwise let us verify that $u_0 = \min(Sv_0^{-1})$ where $v_0 = \min^R(u_0^{-1}S)$: if $u_0 = \varepsilon$ it is clearly true, if $u_0 = u'x$ with $x \in \Sigma$ then there exists $[u', v'] \in Q$ else u_0 were removed from P. Since $u'xv_0 \in S \subseteq L$, there exists $[\alpha, \beta] \in Q(L)$ such that $u'x \in \mathsf{Pre}_{[\alpha,\beta]}$ and $v_0 \in \mathsf{Post}_{[\alpha,\beta]}$ with $\alpha \leq u'x$ and $\beta \leq^R v_0$. As $[u', v'] \in Q \subseteq Q'(S) = Q(L)$, it follows $u'x\beta \in S_L \subseteq S$ then $v_0 = \beta$. Now since $v_0 \notin R$, it follows $u_0 = u'x = \alpha = \min(Sv_0^{-1})$ and $Q \cup \{[u_0, \min^R(u_0^{-1}S)]\} = Q'(S) \setminus \{[u, v] \mid u \in (P \setminus \{[u_0, \min^R(u_0^{-1}S)]\})\}$. Then at the end of the loop we get I and $(P = \emptyset)$ which implies that $Q = Q'(S)$. Finally, since the cardinality of P strictly decreases at each step of the loop, this algorithm stops and computes $Q'(S)$.

This algorithm is polynomial in $||S||$, the size of S, that is the sum of the length of members of S : it uses at most $|\text{pref}(S)|$ steps for the iteration and for each step, each computation is in PTIME *w.r.t.* $||S||$.

At last, recall that the size of the characteristic sample S_L of an ordered biRFSA language L is polynomial according to the number of states of the canonical RFSA of L, that is $|Q(L)|$. More precisely, S_L contains at most $|Q(L)| + |\Sigma| \times |Q(L)|^2$ words. It is worth noting that the canonical RFSA of a language can be exponentially smaller than a DFA recognizing the same language. □

Let us finish this section with an example of computation:

Example 29 *Let $\Sigma = \{a, b\}$. Every language $L_n = \Sigma^* b \Sigma^n$, $n \in \mathbb{N}$, is an ordered biRFSA language. Let us recall that the size of minimal DFA of these languages grows exponentially* w.r.t. *n.*

Let us consider $L_2 = \Sigma^* b \Sigma^2$: its characteristic sample is

$$S_{L_2} = ba^2(\varepsilon + ba^2 + aba^2 + b^2a^2) + b^2a(\varepsilon + a) + bab(\varepsilon + a^2 + ba^2) + aba^2 + b^3a^2$$

If S_{L_2} is given as input then, at the beginning of the computation, we get $P = \text{pref}(S_{L_2})$.

$u = \min(P)$	$\min^R(u^{-1} S_{L_2})$	Q	*what is added to R*	*what is removed from P*
ε	ba^2	$[\varepsilon, ba^2]$	ba^2	ε
a	ba^2			$a + ab + aba + aba^2$
b	a^2	$[b, a^2]$	a^2	b
ba	a	$[ba, a]$	a	ba
b^2	a			$b^2(\varepsilon + a + b + a^2 + ba + ba^2)$
ba^2	ε	$[ba^2, \varepsilon]$	ε	ba^2
bab	ε			$bab(\varepsilon + a + b + a^2 + ba + ba^2)$
ba^3	ba^2			$ba^3(\varepsilon + b + ba + ba^2)$
ba^2b	a^2			$ba^2b(\varepsilon + a + a^2 + b + ba + ba^2)$

7 Conclusion and Prospects

This paper presented learning algorithms for two families of biRFSA languages which are identifiable from positive data. The second one, the family of ordered biRFSA languages, is polynomially identifiable in the limit from positive data. Nevertheless some improvements could be made for this last algorithm. For example, the characteristic sample of the biRFSA language L in example 29 is larger than necessary, indeed if we give the set $ba^2 + b^2a + bab + aba^2 + b^2a^2$ as input, the learning algorithm builds the equivalent automaton recognizing L given figure 7 but it is not the canonical RFSA of L. The canonical RFSA of a language is a *saturated* automaton in which it is not possible to add any transition without changing the language recognized by the automaton, then many transitions are in fact not needed in term of recognition but only in term of unicity of the canonical RFSA.

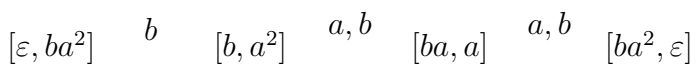


Fig. 7. An automaton for $\Sigma^* b \Sigma^2$.

Let us remark that, for a given biRFSA language L, the property of being an ordered biRFSA language depends on the choice of the fixed order of the alphabet. For instance the language $\Sigma^* a \Sigma^2$ is not an ordered biRFSA if we consider the natural order $a < b$: as $\Sigma^* b \Sigma^2$, used in example 29, it is a biRFSA language but in its canonical RFSA, given figure 4, $[ab, a]$ labels a state which is reached from the initial state by the word aa smaller than ab. A new family of biRFSA languages should be investigated, the family of *weakly* ordered biRFSA languages: a biRFSA language L is weakly ordered if it is ordered for some order of the alphabet. Concerning the algorithm `M` presented in previous section, the choice of an order does not matter in some cases like for the set $\varepsilon + a + b^2$ where `M` computes $(a + bb)^*$ for $a < b$ as for $b < a$ but, most often, it is not the case. For instance, let us consider the set $\varepsilon + a + ab + b^2$. It is easy to see that, for $a < b$, the unique ordered biRFSA language L containing $\varepsilon + a + ab$ is $(a+b)^*$: indeed since ε is in L, we have $[\varepsilon, \varepsilon]$ in $Q(L)$ and, since a is in L, we get in the canonical RFSA of L that $[\varepsilon, \varepsilon] \in \delta([\varepsilon, \varepsilon], a)$. Now, since $ab \in L$, there exists a state $[u, v]$ such that $[u, v] \in \delta([\varepsilon, \varepsilon], a)$ and $[\varepsilon, \varepsilon] \in \delta([u, v], b)$. Then $u = \min(Lv^{-1}) \leq a$ and $v = \min^R(u^{-1}L) \leq^R b$. It follows that $u = v = \varepsilon$ and $L = (a+b)^*$. If we consider now the order $b < a$, our algorithm `M` computes the automaton given figure 8 which recognizes $(a + ab + b^2)^* \subsetneq (a + b)^*$. Clearly, a learning algorithm for the family of weakly ordered biRFSA languages should compute $(a + ab + b^2)^*$ from $\varepsilon + a + ab + b^2$.

Unfortunately, it is not always so clear: let us consider the set $a + b + ab$. With the order $a < b$, `M` computes $(a + b)b^*$ whereas it computes $a^*(a + b)$ for $b < a$. We do not have any reason to choose one language rather than the other.

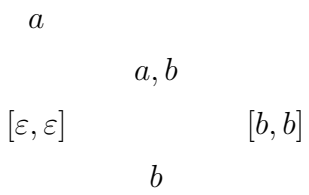


Fig. 8. The automaton computed from $\varepsilon + a + ab + b^2$ with $b < a$.

Nevertheless, if we add the word b^2 to the set, the algorithm computes Σb^* for $a < b$ and Σ^+ for $b < a$. So, to get an algorithm for the family of *weakly ordered biRFSA*, it will be necessary to enlarge the characteristic sample.

Another problem is the following: the algorithm `M` assumes that the input set S satisfies $S_L \subseteq S \subseteq L$ for some ordered biRFSA L. It is not the case for example with the set $S = a^3 + aba + ab^2 + b^3$ for which `M` builds a biRFSA $\mathtt{M}(S)$ which is not ordered. Moreover, some input sets may lead to a computation that is not consistent: let us consider the set $\varepsilon + a + ab$ with $a < b$. It is easy to verify that $\mathtt{M}(\varepsilon + a + ab)$ computes an automaton which recognizes a^*, that is an ordered biRFSA language which does not contain the word ab, a word of the input set, while we have seen before that the unique ordered biRFSA language, for $a < b$, containing $\varepsilon + a + ab$ is $(a + b)^*$. In fact, in order to solve this problem with the definition of S_L given in section 6, we should produce an algorithm `M` which satisfies the following properties:

(1) for any input set S, $\mathtt{M}(S)$ is an ordered biRFSA [1];
(2) for any ordered biRFSA language L, for any set S such that $S_L \subseteq S \subseteq L$, then $\mathtt{M}(S) = L$;
(3) for any sets S, S' such that $S \subseteq S' \subseteq \mathtt{M}(S)$ then $\mathtt{M}(S') = \mathtt{M}(S)$.

References

[1] D. Angluin. Inference of reversible languages. *J. ACM*, 29(3):741–765, July 1982.

[2] C. Choffrut and J. Karhumäki. Combinatorics of words. In G. Rozenberg and A. Salomaa, editors, *Handbook on Formal Languages*, volume I. Springer, Berlin-Heidelberg-New York, 1997.

[3] F. Denis, A. Lemay, and A. Terlutte. Residual finite state automata. In *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science*, number 2010 in Lecture Notes in Computer Science, pages 144–157. Springer Verlag, 2001.

[4] F. Denis, A. Lemay, and A. Terlutte. Some language classes identifiable in the limit from positive data. In *ICGI 2002*, number 2484 in Lecture Notes in Artificial Intelligence, pages 63–76. Springer Verlag, 2002.

[1] This property is called *prudence property* in [10].

[5] F. Denis, A. Lemay, and A. Terlutte. Learning regular languages using rfsas. *Theoretical Computer Science*, 313(2):267–294, 2004.

[6] H. Fernau. Identification of function distinguishable languages. In *ALT 2000*, volume 1968 of *Lecture Notes in Artificial Intelligence*, pages 116–130, 2000.

[7] P. Garcia and E. Vidal. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *"IEEE Transactions on Pattern Analysis and Machine Intell."*, 12(9):920–925, 1990.

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

[9] M. Kanazawa. Identification in the limit of categorial grammars. *Journal of Logic, Language, and Information*, 5(2):115–155, 1996.

[10] M. Kanazawa. *Learnable Classes of Categorial Grammars*. The European Association for Logic, Language and Information. CLSI Publications, 1998.

[11] M. Latteux, Y. Roos, and A. Terlutte. Birfsa languages and minimal nfas. Technical Report GRAPPA-0205, GRAppA, 2005.

[12] J.-E. Pin. On reversible automata. In *Proceedings of Latin American Symposium on Theoretical Informatics (LATIN '92)*, volume 583, pages 401–416, Berlin, Germany, 1992. Springer.

[13] Y. Sakakibara. Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, 97(1):23–60, 1992.

[14] H. Tamm and E. Ukkonen. Bideterministic automata and minimal representations of regular languages. *Theoretical Computer Science*, 328:135–149, 2004.

[15] T. Yokomori. On polynomial-time learnability in the limit of strictly deterministic automata. *Machine Learning*, 2:153–179, 1995.