



## Universal automata and NFA learning<sup>☆</sup>

Pedro García<sup>a</sup>, Manuel Vázquez de Parga<sup>a</sup>, Gloria I. Álvarez<sup>b</sup>, José Ruiz<sup>a,\*</sup>

<sup>a</sup> Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, Spain

<sup>b</sup> Pontificia Universidad Javeriana - Seccional Cali, Calle 18 118-250 Cali, Colombia

### ARTICLE INFO

#### Article history:

Received 27 November 2006  
 Received in revised form 6 May 2008  
 Accepted 16 May 2008  
 Communicated by D. Perrin

#### Keywords:

Grammatical inference  
 Finite automata  
 Universal automaton

### ABSTRACT

The aim of this paper is to develop a new algorithm that, with a complete sample as input, identifies the family of regular languages by means of nondeterministic finite automata. It is a state-merging algorithm. One of its main features is that the convergence (which is proved) is achieved independently from the order in which the states are merged, that is, the merging of states may be done “randomly”.

© 2008 Elsevier B.V. All rights reserved.

## 1. Introduction

Grammatical inference is the discipline that deals with learning finite models that represent a formal language  $L$  either from positive data (a sequence of the words of  $L$ ) or from a complete presentation (a sequence of words that are labeled according to their membership in  $L$ ).

We restrict the learning task to the family of regular languages. As stated by Gold [11,12], there are two problems: this family cannot be learned using positive samples only, and finding the minimal deterministic finite automaton (*DFA*) that is consistent with a complete presentation is a NP-complete problem.

Gold [11] made the first attempt to deal with the task of learning regular languages. His algorithm used a table to determine *distinguishable states* and to establish the transitions between them. The first inference algorithm that used the merging of states as a way of generalizing the input sample is the *RPNI* [17]. The possible merges are done in lexicographical order of the states of the prefix tree acceptor (*PTA*) of the sample. The *PTA* is a tree-shaped automaton that only recognizes the exact sample. Under certain conditions of the training sample, *RPNI* converges to the minimum *DFA* that is consistent with that sample. More recently, using the concept of inclusion between the residuals of states, an extension of *RPNI* that enlarges the training set while learning has been proposed in [9].

Since then, several state-merging algorithms have been proposed, in which the merging order can be established by the training data. The first of these was developed in [13] in which the candidate states to be merged are ordered by the number of training samples crossing each state of the *PTA*. This idea was improved by the *EDSM* [15], which uses a control strategy called *blue-fringe* in which one of the candidate states to be merged is in the root of a tree. Another attempt to improve the efficiency of this method was made in [5], where the set of candidates to be merged was limited by a given distance. Since the output of this algorithm was very sensitive to the first merges done, a new measure called *shared evidence* was proposed in [1] to overcome this difficulty.

<sup>☆</sup> Work partially supported by Spanish Ministry of Education and Science research project TIN2007-60769.

\* Corresponding author.

E-mail addresses: [pgarcia@dsic.upv.es](mailto:pgarcia@dsic.upv.es) (P. García), [mvazquez@dsic.upv.es](mailto:mvazquez@dsic.upv.es) (M. Vázquez de Parga), [galvarez@dsic.upv.es](mailto:galvarez@dsic.upv.es) (G.I. Álvarez), [jruiz@dsic.upv.es](mailto:jruiz@dsic.upv.es) (J. Ruiz).

All the above-mentioned algorithms have in common that the output hypotheses are DFAs. More recently, starting with the idea that nondeterministic finite automata are generally smaller descriptions for a language than their equivalent deterministic ones, algorithms that output non deterministic finite automata have been proposed. One of them is the *DeLeTe2* algorithm [8]. It outputs a special type of automaton called *RFSA* (Residual finite-state automaton). A finite automaton  $\mathcal{A} = (Q, \Sigma, \delta, I, F)$  is a *RFSA* if for every  $q \in Q$  the language  $\{x \mid \delta(q, x) \cap F \neq \emptyset\}$  is a residual language of  $L(\mathcal{A})$ . We recall that the residual of a language  $L$  with regard to a word  $u$ , denoted by  $u^{-1}L$ , is  $u^{-1}L = \{v \in \Sigma^* : uv \in L\}$ . In [7], a subclass of the class of NFAs called unambiguous finite automata (*UFA*) is defined. One of the properties of *UFA* is that the same target language will be achieved independently of the order of the merging of the states. Some algorithms that use the same strategy as *RPNI* but that output NFAs have also been proposed [2].

While most of the inference algorithms start from the prefix tree acceptor of the sample, an algorithm has recently been proposed in [19] that begins constructing the subautomaton associated with every word of the positive sample.

In this work we propose a state-merging algorithm that, with a *universal sample* as input, converges to a nondeterministic finite automaton that recognizes the target language, independently of the order in which the states are merged. The concept of universal sample is defined and its finiteness is proved. This algorithm, as is described here is theoretical and is not intended to be used in practice as it is stated. However the concept of universal automaton for a language and some other concepts that have been used to prove its convergence will clarify and simplify ideas about the convergence of previous inference algorithms, as well as of those that may be proposed in the future. Note that even though different orders of merging states may lead to different hypotheses (automata), on convergence, the language accepted by those automata will be the target language.

The article is structured as follows: Section 2 presents some preliminary definitions and the notations used throughout the paper; Section 3 presents the definition of universal sample and some propositions to prove its finiteness. Section 4 presents the algorithm, a proof of its convergence, an example of a run and an experiment that shows the different hypotheses that can be obtained when the algorithm has converged, depending on the order of merging states. Finally, Section 5 presents the conclusions.

## 2. Definitions and notations

In this section, we will describe some facts about formal languages in order to make the notation understandable to the reader. For further details about the definitions, the reader is referred to [14].

### 2.1. Languages and automata

Let  $A$  be a finite alphabet and  $A^*$  the free monoid generated by  $A$  with concatenation as the binary operation and  $\lambda$  as neutral element. A language  $L$  is any subset of  $A^*$ , the elements  $x \in A^*$  are called *words*.

A (non deterministic) *finite automaton* (*NFA*) is a 5-tuple  $\mathcal{A} = (Q, A, \delta, I, F)$ , where  $Q$  is a finite set of states,  $A$  is an alphabet,  $I, F \subseteq Q$  are respectively the set of initial and final states and  $\delta : Q \times A \rightarrow 2^Q$  is the transition function, which will also be denoted as  $\delta \subseteq Q \times A \times Q$ .

Given  $P \subseteq Q$  and  $a \in A$ ,  $\delta(P, a) = \cup_{q \in P} \delta(q, a)$ . The function  $\delta$  is extended to words writing  $\delta(P, \lambda) = P$  and  $\delta(P, xa) = \delta(\delta(P, x), a)$ , for every  $a \in A$ ,  $x \in A^*$ . The language accepted by  $\mathcal{A}$  will be denoted as  $L(\mathcal{A})$ , that is,  $L(\mathcal{A}) = \{x \in A^* : \delta(I, x) \cap F \neq \emptyset\}$ . Two automata are equivalent if they accept the same language. The *left language* of a state  $q$  with respect to  $\mathcal{A}$  is  $L_q = \{x \in A^* : q \in \delta(I, x)\}$ .

A finite automaton  $\mathcal{A}$  is *deterministic* if  $\text{Card}(I) = 1$  and for every state  $q$  and every symbol  $a$ ,  $\text{Card}(\delta(q, a)) \leq 1$ .

A *subautomaton* of a non deterministic finite automaton  $\mathcal{A} = (Q, A, \delta, I, F)$  is any finite automaton  $\mathcal{A}' = (Q', A, \delta', I', F')$  where  $Q' \subseteq Q$ ,  $I' \subseteq I \cap Q'$ ,  $F' \subseteq F \cap Q'$  and  $\delta' \subseteq \delta \cap Q' \times A \times Q'$ .

It is easily seen that if  $\mathcal{A}'$  is a subautomaton of  $\mathcal{A}$ , then  $L(\mathcal{A}') \subseteq L(\mathcal{A})$ .

Given  $\mathcal{A} = (Q, A, \delta, I, F)$  and  $\mathcal{B} = (Q', A, \delta', I', F')$ , the function  $\varphi : Q \rightarrow Q'$  is a *homomorphism* from  $\mathcal{A}$  to  $\mathcal{B}$  if  $\varphi(I) \subseteq I'$ ,  $\varphi(F) \subseteq F'$  and  $\varphi(\delta(q, a)) \subseteq \delta'(\varphi(q), a)$  for any  $q$  in  $Q$  and  $a$  in  $A$ . The subautomaton of  $\mathcal{B}$  induced by  $\varphi(Q)$  is denoted as  $\varphi(\mathcal{A})$ . It follows that  $L(\mathcal{A}) \subseteq L(\varphi(\mathcal{A})) \subseteq L(\mathcal{B})$ .

Let  $D \subset A^*$  finite. The *maximal automaton* for  $D$  is the NFA  $MA(D) = (Q, A, \delta, I, F)$  where  $Q = \cup_{x \in D} \{(u, v) \in A^* \times A^* : uv = x\}$ ,  $I = \{(\lambda, x) : x \in D\}$ ,  $F = \{(x, \lambda) : x \in D\}$  and for  $(u, av) \in Q$ ,  $\delta((u, av), a) = (ua, v)$ . So defined  $L(MA(D)) = D$ .

The *merge* of states  $p$  and  $q$  in a finite automaton  $\mathcal{A} = (Q, A, \delta, I, F)$  is defined as follows:  $\text{merge}(\mathcal{A}, p, q) = (\varphi(Q), A, \delta', I', F')$  where  $\varphi(q) = p$  and  $\forall r \neq q$ ,  $\varphi(r) = r$ , also  $I' = \varphi(I)$ ,  $F' = \varphi(F)$  and  $(r, a, s) \in \delta$  if and only if  $(\varphi(r), a, \varphi(s)) \in \delta'$ . It follows that  $L(\mathcal{A}) \subseteq L(\text{merge}(\mathcal{A}, p, q))$ . Two states  $p$  and  $q$  are *mergible* if  $L(\text{merge}(\mathcal{A}, p, q)) = L(\mathcal{A})$ .

Let  $\mathcal{A} = (Q, A, \delta, I, F)$  be an automaton and let  $\pi$  be a partition of  $Q$ . Let  $B(q, \pi)$  be the class of  $\pi$  that contains  $q$ . The *quotient automaton* is  $\mathcal{A}/\pi = (Q', A, \delta', I', F')$ , where  $Q' = Q/\pi = \{B(q, \pi) : q \in Q\}$ ,  $I' = \{B \in Q' : B \cap I \neq \emptyset\}$ ,  $F' = \{B \in Q' : B \cap F \neq \emptyset\}$  and the transition function is  $B' \in \delta'(B, a)$  if and only if  $\exists q \in B$ ,  $\exists q' \in B'$  with  $q' \in \delta(q, a)$ .

An automaton  $\mathcal{A}$  is *irreducible* if there is no non-trivial partition  $\pi$  of its set of states such that  $L(\mathcal{A}/\pi) = L(\mathcal{A})$ , that is, if the merging of any pair of states leads to an automaton that recognizes a different language.

Given a language  $L$ , let  $\mathcal{U}$  be the set of all the possible intersections of residuals of  $L$  with respect to the words over a certain alphabet  $A$ , that is,  $\mathcal{U} = \{u_1^{-1}L \cap \dots \cap u_k^{-1}L : k \geq 0, u_1, \dots, u_k \in A^*\}$ . If  $L$  is a regular language,  $\mathcal{U}$  is finite. The *universal automaton* (*UA*) [3,4,6,16,18] for  $L$  is defined as  $\mathcal{U} = (\mathcal{U}, A, \delta, I, F)$  with:

- $I = \{q \in U : q \subseteq L\}$ .
- $F = \{q \in U : \lambda \in q\}$ .
- The transition function is such that  $q \in \delta(p, a)$  iff  $q \subseteq a^{-1}p$ .

If the number of states of the minimal DFA recognizing  $L$  is  $n$ , the number of states of the UA for  $L$  lies between  $n$  and  $2^n$ . Finally, we recall that the UA for a language  $L$  does not have any mergible states [10].

With regard to the universal automaton, we have the following theorem that states that every automaton that recognizes a subset of a language  $L$  can be projected into the UA for  $L$ .

**Theorem 1** ([4,6]). Let  $\mathcal{U} = (U, A, \delta, I, F)$  be the universal automaton for  $L \subseteq A^*$ . Then:

- (1)  $L(\mathcal{U}) = L$ .
- (2) For any automaton  $\mathcal{A} = (Q, A, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}})$  such that  $L(\mathcal{A}) \subseteq L$ , the function  $\varphi : Q \rightarrow U$  defined as  $\varphi(q) = \bigcap_{u \in L_q} u^{-1}L$  is an automata homomorphism.

## 2.2. Grammatical inference

Regular language learning is an important issue of grammatical inference, which is defined as the process of learning an unknown formal language from a finite set of labeled examples.

A positive (resp. negative) sample of  $L$  is any finite set  $D_+ \subseteq L$  (resp.  $D_- \subseteq \bar{L}$ ). In the case that it contains positive and negative words, it will be denoted as  $(D_+, D_-)$  and called a *complete sample*. A *complete presentation* of  $L \subseteq \Sigma^*$  is a sequence of all the words of  $\Sigma^*$  labeled according to their membership to  $L$ .

An *inference algorithm* is an algorithm that, on input of any sample, outputs a representation of a language. The algorithm is *consistent* if the output contains  $D_+$  and is disjoint with  $D_-$ .

The type of convergence that we will use in our algorithms was defined by Gold [11,12] and is called *identification in the limit*. It is a mathematical framework that was proposed to analyze the behavior of different learning tasks in a computational way.

Given a family  $\mathcal{L}$  of languages,  $\mathcal{H}$  is a set of hypotheses for  $\mathcal{L}$  if, for every  $L \in \mathcal{L}$ , there is  $h \in \mathcal{H}$  that describes  $L$ . For the family of regular languages,  $\mathcal{H}$  can be the set of NFAs.

An algorithm  $\mathcal{I}\mathcal{A}$  identifies a class of languages  $\mathcal{L}$  by means of hypotheses in  $\mathcal{H}$  *in the limit* if and only if, for any  $L \in \mathcal{L}$ , and any presentation of  $L$  (i.e., a sequence of words of  $\Sigma^*$  classified according to their membership to  $L$ ), the infinite sequence of hypotheses output by  $\mathcal{I}\mathcal{A}$  converges to  $h \in \mathcal{H}$  such that  $L(h) = L$ . In other words, there exists  $t_0$  such that  $(t \geq t_0 \Rightarrow h_t = h_{t_0} \wedge L(h_{t_0}) = L)$ , where  $h_t$  denotes the hypothesis output by  $\mathcal{I}\mathcal{A}$  after processing  $t$  examples.

Equivalently,  $\mathcal{A}$  identifies a class of languages  $\mathcal{L}$  by means of hypotheses in  $\mathcal{H}$  *in the limit* if there exists, for every  $L \in \mathcal{L}$ , a pair  $(D_+, D_-)$  such that when  $\mathcal{A}$  is supplied with any pair  $(D'_+, D'_-)$  with the condition that  $D_+ \subseteq D'_+$  and  $D_- \subseteq D'_-$ , it outputs the same hypothesis  $h \in \mathcal{H}$  for  $L$ .

The learning algorithm that we propose in this paper is an algorithm in which the generalizing process is based on merging states from a starting automaton. The starting point is the maximal automaton for the positive samples, and the merges are performed under the control of the negative samples.

## 3. Universal sample

The aim of this section is to define and prove the existence of a finite set of words  $D_+ \subseteq L$ , for every regular language  $L$ . This set of words will be called *universal sample* for  $L$ . This set has the property that any partition  $\pi$  that makes  $MA(D_+)/\pi$  irreducible in  $L$  defines a subautomaton of  $\mathcal{U}$  that accepts exactly  $L$  (see Theorem 15).

**Definition 2.** An automaton  $\mathcal{A}$  is irreducible in a regular language  $L$  if and only if  $L(\mathcal{A}) \subseteq L$  and  $L(\mathcal{A}/\pi) - L \neq \emptyset$ , for any non trivial partition  $\pi$  of the states of  $\mathcal{A}$ .

Therefore, an automaton  $\mathcal{A}$  is irreducible if and only if it is irreducible in  $L(\mathcal{A})$ . In fact,  $\mathcal{A}$  is irreducible if and only if there exists a language  $L$  such that  $\mathcal{A}$  is irreducible in  $L$ .

**Proposition 3** ([10]). Let  $\mathcal{A}$  be an automaton accepting  $L$ , and let  $\mathcal{U}$  be the universal automaton of  $L$ . Let  $\varphi$  be the morphism of Theorem 1 that maps  $\mathcal{A}$  in  $\mathcal{U}$ . If there exist  $k$  states of  $\mathcal{A}$ ,  $q_1, \dots, q_k$ , such that  $\varphi(q_1) = \dots = \varphi(q_k)$ , then the states  $q_1, \dots, q_k$  are mergible.

**Proposition 4.** Let  $D_+ \subseteq L$  be finite and let  $\pi$  be a partition of the states of  $MA(D_+)$  such that  $MA(D_+)/\pi$  is irreducible in  $L$ . Then,  $MA(D_+)/\pi$  is isomorphic to a subautomaton of  $\mathcal{U}$  (the universal automaton of  $L$ ).

**Proof.** As  $MA(D_+)/\pi$  is irreducible in  $L$ , then  $L(MA(D_+)/\pi) \subseteq L$ , and this quotient automaton does not have mergible states. Then, by Proposition 3, the morphism  $\varphi$  of Theorem 1 is injective. ■

**Definition 5.** Let  $\mathcal{A} = (Q, A, \delta, I, F)$  be a nondeterministic finite automaton and let  $x = a_1 a_2 \dots a_n \in L(\mathcal{A})$ . An *acceptance path* for  $x$  in  $\mathcal{A}$  is a sequence of arcs  $\langle (q_1, a_1, q_2), (q_2, a_2, q_3), \dots, (q_n, a_n, q_{n+1}) \rangle$  with  $q_1 \in I, q_{n+1} \in F$ .

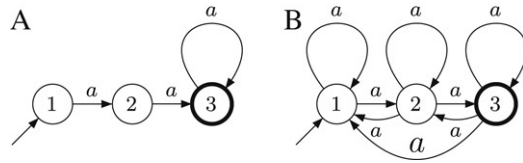


Fig. 1. (A) Minimal deterministic automaton for language  $L = a^2 a^*$ . (B) Universal automaton for  $L$ .

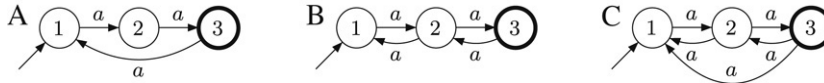


Fig. 2. Subautomaton induced by one of the accepting paths for  $a^5$  (A) and for  $a^6$  in  $\mathcal{U}$ . Juxtaposition of the paths (C).

**Definition 6.** Given a path  $\mathcal{C} = \langle (q_1, a_1, q_2), (q_2, a_2, q_3), \dots, (q_n, a_n, q_{n+1}) \rangle$ , the subautomaton of  $\mathcal{A}$  induced by  $\mathcal{C}$  is  $\mathcal{A}_{\mathcal{C}} = (Q', A, \delta', \{q_1\}, \{q_{n+1}\})$ , where  $Q'$  is the set of distinct states of  $q_1, q_2, \dots, q_{n+1}$  and  $\delta'$  is the set of transitions in  $\mathcal{C}$ .

**Definition 7.** Given a NFA  $\mathcal{A} = (Q, A, \delta, I, F)$  and the collection of subautomata  $\{\mathcal{A}_i\}_{i=1}^n$ , where  $\mathcal{A}_i = (Q_i, A, \delta_i, I_i, F_i)$  the juxtaposition of the subautomata is the subautomaton  $\mathcal{A}' = (Q', A, \delta', I', F')$  with  $Q' = \bigcup_{i=1}^n Q_i, I' = \bigcup_{i=1}^n I_i, F' = \bigcup_{i=1}^n F_i$  and  $\delta'(q, a) = \bigcup_{i=1}^n \delta_i(q, a)$ , for every  $q \in Q'$  and every  $a \in A'$ .

From this definition, it is clear that  $\bigcup_{i=1}^n L(\mathcal{A}_i) \subseteq L(\mathcal{A}') \subseteq L(\mathcal{A})$ .

Let  $\mathcal{A}$  be an automaton, let  $x \in L(\mathcal{A})$ , and let  $\mathcal{C}_x$  be the set of acceptance paths for  $x$  in  $\mathcal{A}$ . Given  $\{x_1, x_2, \dots, x_n\} \subseteq L(\mathcal{A})$ , we can associate a subautomaton  $\mathcal{A}_{\{c_1, \dots, c_n\}}$  to every set  $\{c_1, c_2, \dots, c_n\}$ , with  $c_i \in \mathcal{C}_{x_i}$ . This subautomaton  $\mathcal{A}_{\{c_1, \dots, c_n\}}$  is obtained as juxtaposition of the subautomata  $\mathcal{A}_{c_i}$  (each of which is associated to one of the paths).

The following example will clarify the previous concepts.

**Example 8.** Let  $L = a^2 a^*$ . A deterministic automaton for  $L$  is depicted in Fig. 1(A), and the universal automaton for  $L$  is depicted in Fig. 1(B).

For example, given the words  $a^5$  and  $a^6$  of  $L$ , possible accepting paths in the UA of  $L$  are:  $\langle (1, a, 2), (2, a, 3), (3, a, 1), (1, a, 2), (2, a, 3) \rangle$  for  $a^5$ , and  $\langle (1, a, 2), (2, a, 3), (3, a, 2), (2, a, 1), (1, a, 2), (2, a, 3) \rangle$  for  $a^6$ . The subautomata of  $\mathcal{U}$  induced by those paths and their juxtaposition are shown in Fig. 2.

**Definition 9.** Let  $D_+ = \{x_1, \dots, x_n\} \subseteq L$ . For every  $i, 1 \leq i \leq n$ , let  $\{c_1^{(i)}, \dots, c_{n_i}^{(i)}\}$  be the set of all accepting paths for  $x_i$ , and let  $\{\mathcal{A}_1^{(i)}, \dots, \mathcal{A}_{n_i}^{(i)}\}$  be the set of their induced subautomata.  $D_+$  is a universal sample for  $L$  if for every choice  $j_i = 1, \dots, n_i$  with  $i = 1, \dots, n$  the subautomaton of  $\mathcal{U}$  obtained by the juxtaposition of the subautomata  $\mathcal{A}_{j_1}^{(1)}, \mathcal{A}_{j_2}^{(2)}, \dots, \mathcal{A}_{j_n}^{(n)}$  recognizes  $L$ .

**Proposition 10.** Let  $\mathcal{U}$  be the universal automaton for  $L$ . Let  $\mathcal{A}$  be a subautomaton of  $\mathcal{U}$  such that  $L(\mathcal{A}) = L$ , let  $x \in L$ , let  $\mathcal{C} = \{c_1^{(x)}, \dots, c_{n_x}^{(x)}\}$  be the set of all accepting paths for  $x$  in  $\mathcal{U}$ , and let  $\{\mathcal{A}_1^{(x)}, \dots, \mathcal{A}_{n_x}^{(x)}\}$  be the set of subautomata of  $\mathcal{U}$  induced by  $\mathcal{C}$ . The juxtaposition of  $\mathcal{A}$  and  $\mathcal{A}_i^{(x)}$  for every  $i, 1 \leq i \leq n_x$  accepts  $L$ .

**Proof.** The juxtaposition of  $\mathcal{A}$  and any  $\mathcal{A}_i^{(x)}$  is a subautomaton of  $\mathcal{U}$ , and thus it accepts a subset of  $L$ . Since  $\mathcal{A}$  accepts  $L$ , the juxtaposition of  $\mathcal{A}$  and  $\mathcal{A}_i^{(x)}$  also accepts  $L$ . ■

**Proposition 11.** For every regular language  $L$ , there exists a finite universal sample.

**Proof.** To give a constructive proof, we build a tree whose nodes are subautomata of the UA of  $L$  as follows:

- The root of the tree is the empty subautomaton.
- If a node accepts  $L$ , it has no successors.
- To obtain the successors of a node  $\mathcal{A}$  with  $L(\mathcal{A}) \neq L$ , we choose a word  $x$  of  $L$  that is not accepted by  $\mathcal{A}$ . Let  $\mathcal{C} = \{c_1^{(x)}, \dots, c_{n_x}^{(x)}\}$  be the set of accepting paths for  $x$  in  $\mathcal{U}$ , and let  $\{\mathcal{A}_1^{(x)}, \dots, \mathcal{A}_{n_x}^{(x)}\}$  be the set of subautomata of  $\mathcal{U}$  induced by  $\mathcal{C}$ . Every automaton obtained as the juxtaposition of  $\mathcal{A}$  and  $\mathcal{A}_i^{(x)}, 1 \leq i \leq n_x$  is a successor of the node  $\mathcal{A}$ .

The successors of  $\mathcal{A}$  in the tree are “bigger” than  $\mathcal{A}$ , as they may have more arcs and more initial or final states. As  $\mathcal{U}$  is finite and the nodes are subautomata of the UA of  $L$ , it follows that the depth of the tree is finite, and therefore, the number of words necessary to build the tree is a finite set that we call  $D_+$ .

Every leaf-node of the tree that we have built represents a subautomaton of  $\mathcal{U}$  that accepts  $L$ , and every node is obtained using a subset of  $D_+$ . By Proposition 10,  $D_+$  is a universal sample for  $L$ . ■

**Proposition 12.** Let  $D_+$  be a universal sample for a language  $L$  and let  $x \in L - D_+$ . Then  $D_+ \cup \{x\}$  is also a universal sample for  $L$ .

**Proof.** As  $D_+$  is a universal sample, any subautomaton induced by the words of  $D_+$  recognizes  $L$ . By Proposition 10, if we add any word of  $L$  the induced subautomata also recognize  $L$ . Thus,  $D_+ \cup \{x\}$  is also a universal sample for  $L$ . ■

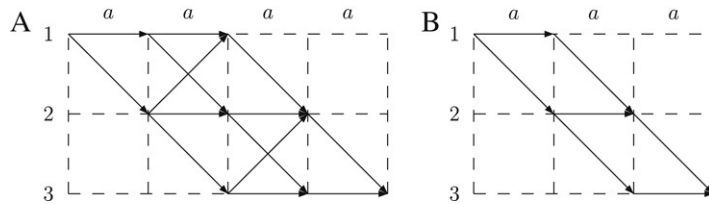


Fig. 3. Trellis of all accepting paths in  $\mathcal{U}$ : (A) For the word  $a^4$ , (B) For  $a^3$ .



Fig. 4. Subautomata induced by the paths 12123 and 12323, which do not accept  $L$ .

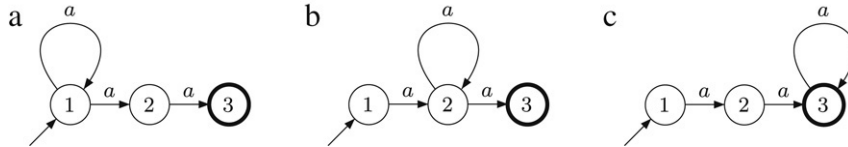


Fig. 5. Subautomata of  $\mathcal{U}$  induced by the different accepting paths for  $a^3$ : (a) 1123 (b) 1223 and (c) 1233.

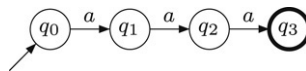


Fig. 6.  $MA(a^3)$ .

**Corollary 13.** If  $D'_+ \subseteq L$  includes a universal sample for a language  $L$  then  $D'_+$  is also a universal sample for  $L$ .

To better understand of the above definitions and propositions, we present the following example:

**Example 14.** Let  $L$  be the language  $a^2a^*$  used in the previous example (Example 8). The set  $\{a^4\}$  uses every transition in  $\mathcal{A}$ . The trellis of all accepting paths of  $\{a^4\}$  in  $\mathcal{U}$  is shown in Fig. 3(A). As the alphabet contains one symbol, we will describe the paths using only the visited states, then the set of accepting paths is  $C_{a^4} = \{11123, 11223, 11233, 12123, 12223, 12233, 12323, 12333\}$ .

All the subautomata of  $\mathcal{U}$  induced by  $C_{a^4}$  (except those induced by 12123 and by 12323) accept  $L$ . The subautomata induced by those paths are shown in Fig. 4. Note that these automata do not accept, for example,  $a^3$ .

The accepting paths for the word  $a^3$  are shown in Fig. 3; they are  $C_{a^3} = \{1123, 1223, 1233\}$ . The subautomata associated to these paths recognize  $L$  (see Fig. 5) and their juxtaposition with those subautomata depicted in Fig. 4 also recognize  $L$ . Therefore, the sample  $\{a^3, a^4\}$  is universal for  $L$ .

In fact,  $\{a^3\}$  is universal for  $L$ . The automaton  $MA(a^3)$  is shown in Fig. 6. It is easily seen that the only partitions of its set of states that make the quotient automata irreducible and consistent with  $\bar{L}$  are  $\{\{q_0, q_1\}, q_2, q_3\}$ ,  $\{\{q_1, q_2\}, q_0, q_3\}$ , and  $\{\{q_2, q_3\}, q_0, q_1\}$ .

Note that  $(q_0, q_2)$ ,  $(q_0, q_3)$  and  $(q_1, q_3)$  cannot be in the same block of a partition of  $MA(a^3)$  if the quotient automaton by that partition is irreducible in  $L$ . This comes from the fact that the accepting paths of  $a^3$  in  $\mathcal{U}$  and the pairs  $(\lambda, a^2)$ ,  $(\lambda, a^3)$ ,  $(a, a^3)$  do not reach the same state in  $\mathcal{U}$ . Thus, the only irreducible automata for the sample  $\{a^3, a^4\}$  are shown in Fig. 5, and all of them accept the target language  $a^2a^*$ .

**Theorem 15.** Let  $L$  be a regular language and let  $D_+$  be a universal sample for  $L$ . Let  $MA(D_+)$  be the maximal automaton for  $D_+$ , and let  $\pi$  be any partition of the states of  $MA(D_+)$  such that  $MA(D_+)/\pi$  is irreducible in  $L$ . Then  $L(MA(D_+)/\pi) = L$ .

**Proof.** As  $MA(D_+)/\pi$  is irreducible in  $L$ , by Proposition 4,  $MA(D_+)/\pi$  is isomorphic to a subautomaton of  $\mathcal{U}$ . As  $D_+$  is universal for  $L$ , then  $L(MA(D_+)/\pi) = L$ . ■

Given a regular language  $L$  and a finite positive sample  $D_+ \subseteq L$ , determining whether or not a partition  $\pi$  of the states of  $MA(D_+)$  makes the quotient automaton  $MA(D_+)/\pi$  be an irreducible automaton in  $L$  requires prior knowledge of  $\bar{L}$ . This language might be infinite. In the following, we will show that a finite subset of  $\bar{L}$  will be enough to determine the irreducibility of  $MA(D_+)/\pi$ .

**Proposition 16.** Given a finite set  $D_+ \subseteq L$ , there exists a finite set  $D_- \subseteq \bar{L}$  such that, if  $\pi$  is a partition of the set of states of  $MA(D_+)$  so that  $MA(D_+)/\pi$  is irreducible in  $\bar{D}_-$ , then  $MA(D_+)/\pi$  is irreducible in  $L$ .

**Proof.** Merging two states  $p$  and  $q$  of any automaton means establishing a partition in its set of states where one of the blocks contains exactly  $p$  and  $q$  and the rest of the blocks are singletons. Let  $p$  and  $q$  be any two states of  $MA(D_+)$  with the condition that  $L(\text{merge}(MA(D_+), p, q)) - L \neq \emptyset$ . To avoid this merge, it is sufficient to add any word belonging to

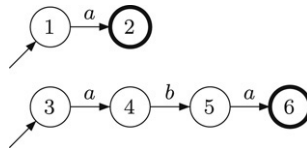


Fig. 7. Maximal automata for  $D_+ = \{a, aba\}$ .

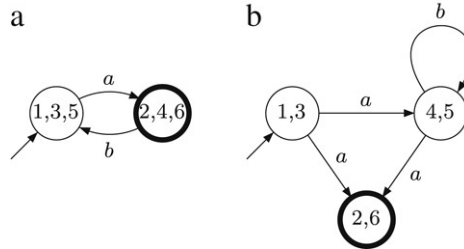


Fig. 8. (a)  $MA(D_+)/\pi_1$  and (b)  $MA(D_+)/\pi_2$ .

$L(\text{merge}(MA(D_+), p, q)) - L$  to  $D_-$ . The finiteness of  $D_-$  follows from the fact that the number of states whose merging has to be avoided is finite. ■

**Corollary 17.** Let  $D_+$  a universal sample for  $L$ . There exists a finite set  $D_-$  such that any partition  $\pi$  of the set of states of  $MA(D_+)$  that is irreducible in  $\overline{D_-}$  verifies that  $L(MA(D_+)/\pi) = L$ .

The following example will help to understand the previous statement:

**Example 18.** Let  $L$  be the language denoted by the regular expression  $a(bb^*a)^*$  and let  $D_+ = \{a, aba\}$ . The maximal automaton  $MA(D_+)$  is shown in Fig. 7.

States 1 and 2 cannot be merged since the resulting automaton would recognize words in  $\overline{L}$ . For example, the word  $\lambda$  added to  $D_-$  would prevent that merge. Proceeding this way with all the pairs of states, the (non unique) set  $D_- = \{\lambda, aa, ab, ba\}$  can be built.

With this set,  $\lambda$  avoids the merging of (1, 2), (1, 6), (2, 3) and (3, 6);  $aa$  avoids the merging of (1, 4) and (2, 5);  $ab$  avoids the merging of (5, 6) and  $ba$  avoids the merging of (3, 4). Any partition  $\pi$  that contains both states in one of these pairs in the same block would make  $MA(D_+)/\pi$  not to be irreducible in  $L$ .

Partitions  $\pi_1 = \{\{1, 3, 5\}, \{2, 4, 6\}\}$  and  $\pi_2 = \{\{1, 3\}, \{2, 6\}\{4, 5\}\}$  are examples of partitions that are irreducible in  $\overline{D_-}$  and therefore in  $\overline{L}$ . Fig. 8 shows the quotient automata obtained for  $\pi_1$  and  $\pi_2$ . Note that they do not recognize the target language. This is due to the fact that  $D_+$  is not a universal sample (which is easily seen from the universal automaton for the language, see Fig. 17(b)). Therefore, the merging of states in  $MA(D_+)$  only guarantees obtaining a subset of the target language and not the target language itself.

Let  $D_+$  be a universal sample for  $L$ , and let  $D_-$  be a finite set that guarantees that any partition in the states of  $MA(D_+)$  gives an automaton that recognizes  $L$ . If we add new words of  $L$  to the set  $D_+$ , the new set continues to be a universal sample for  $L$ , but  $D_-$  may not guarantee that any partition in the maximal automaton of the new  $D_+$  still accepts  $L$ . This fact will be reflected in the OIL Algorithm that is presented in Section 4. To better understand this fact, we present the following example:

**Example 19.** Let  $L = a^* + b^*$ . The set  $D_+ = \{\lambda, a, b, aa, bb, aaa, bbb\}$  is universal for  $L$  and  $D_- = \{ab, ba, abb, bba, abbb, bbba\}$  is such that for any partition in  $MA(D_+)$  that is irreducible in  $\overline{D_-}$ , the quotient automaton with respect to that partition accepts  $L$ . Let us suppose that the chosen partition gives the automaton shown in Fig. 9(a).

If we add the word  $bbbb$  to  $D_+$ , keeping  $D_-$  as it was, there should be an irreducible partition in  $\overline{D_-}$  that could give the automaton shown in Fig. 9(b).

To avoid this possibility, a word, let us say  $abbbb$ , must be added to  $D_-$ . Note that for any  $b^n$  added to  $D_+$ , we need to add at least  $ab^n$  to the set of negative samples.

#### 4. A family of order-independent merging-state NFA learning algorithms

Based on the above definitions and theorems we propose the **OIL** (Order Independent Learning) family of algorithms, that is described in Algorithm 1. We will prove that this algorithm identifies the family of regular languages in the limit.

When a set of blocks of positive and negative samples for the target language  $L$  is input to the algorithm, an automaton that recognizes  $L$  in the limit is obtained. Note that a block may contain just a single word, so the algorithm is presented here in a very general way.

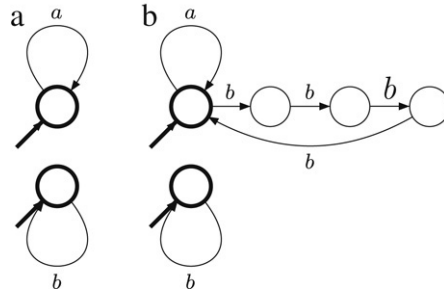


Fig. 9.

**Algorithm 1 OIL**

**Require:** A sequence of blocks  $\langle (D_+^{(1)}, D_-^{(1)}), (D_+^{(2)}, D_-^{(2)}), \dots, (D_+^{(n)}, D_-^{(n)}) \rangle$ .

**Ensure:** An irreducible automaton consistent with the sample (recognizes the target language in the limit).

- 1: **STEP 1:**
- 2: Build  $MA(D_+^{(1)})$ ;
- 3:  $D_- = D_-^{(1)}$ ;
- 4: Find a partition  $\pi$  of the states of  $MA(D_+^{(1)})$  such that  $MA(D_+^{(1)})/\pi$  is irreducible in  $\overline{D_-}$ .
- 5: **STEP  $i + 1$ :**
- 6: Let  $\mathcal{A} = (Q, A, \delta, I, F)$  be the output of the algorithm after processing the first  $i$  blocks, for  $i \geq 1$ .
- 7:  $D_- = D_- \cup D_-^{(i+1)}$ .
- 8: **if**  $\mathcal{A}$  is consistent with  $(D_+^{(i+1)}, D_-^{(i+1)})$  **then**
- 9:   Go to **Step**  $i + 2$ .
- 10: **end if**
- 11: **if**  $\mathcal{A}$  is consistent with  $D_-^{(i+1)}$  **then**
- 12:    $D_+^{(i+1)'} = D_+^{(i+1)} - L(\mathcal{A})$ ;
- 13:   Build  $MA(D_+^{(i+1)'})$ ;  $//MA(D_+^{(i+1)'}) = (Q', A, \delta', I', F')//$
- 14:    $\mathcal{A}' = (Q \cup Q', A, \delta \cup \delta', I \cup I', F \cup F')$ ;
- 15:   Find a partition  $\pi$  of  $Q \cup Q'$  such that  $\mathcal{A}'/\pi$  is irreducible in  $\overline{D_-}$ .
- 16:    $\mathcal{A} = \mathcal{A}'/\pi$ ; Go to **Step**  $i + 2$ .
- 17: **end if**
- 18: **if**  $\mathcal{A}$  is not consistent with  $D_-^{(i+1)}$  **then**
- 19:   **Run OIL** with input  $\langle (D_+^{(1)}, D_-), (D_+^{(2)}, D_-), \dots, (D_+^{(i+1)}, D_-) \rangle$
- 20:   Go to **Step**  $i + 2$ .
- 21: **end if**
- 22: Return  $\mathcal{A}$

The method starts building the maximal automaton for  $D_+^{(1)}$  and merges the states in a random order until the algorithm obtains an irreducible automaton in  $\overline{D_-^{(1)}}$ .

The algorithm performs the following steps for every new block:

- (1) If the existing automaton is consistent with the new block, nothing has to be done.
- (2) If it is consistent with the new set of negative samples, the algorithm deletes the superfluous positive words (i.e., those that are accepted by the previous hypothesis). Then it builds the maximal automaton for the new set of positive words, adds the new negative words to  $D_-$  and finds a partition of the states of the automaton until an irreducible automaton in  $\overline{D_-}$  is obtained.
- (3) Otherwise the algorithm runs itself taking into account the whole set of negative samples at every step. This part of the algorithm (lines 18–21) overcomes the fact that, even though we may have a universal sample, the negative samples may not lead to consistency.

#### 4.1. Example of a run

The algorithm has been presented in a very general way, as there are many possible ways of obtaining a partition of a set of states (lines 4 and 15 of the algorithm). One implementation of this algorithm can be done by ordering the states in  $MA(D_+^{(1)})$  randomly (from 1 to  $N_1$ , with  $N_1$  being the number of states of  $MA(D_+^{(1)})$ ) and merging the states in that order so

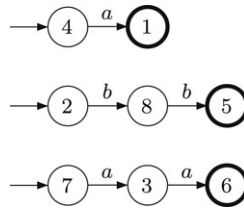


Fig. 10.  $MA(D_+^{(1)})$  with randomly numbered states.

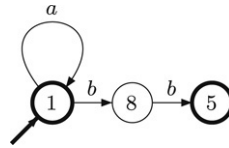


Fig. 11. Irreducible (in  $\overline{D_-^{(1)}}$ ) automaton after merging states in  $D_+^{(1)}$ .

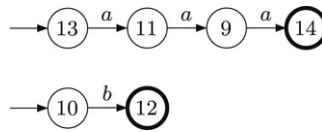


Fig. 12.  $MA(D_+^{(2)})$  with randomly numbered states.

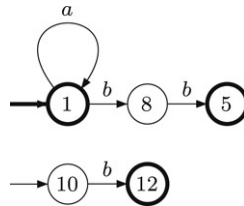


Fig. 13. Automaton after joining the current hypothesis  $M$  with the maximal automaton of the samples from the second block that are not accepted by  $M$ .

that an irreducible automaton in  $\overline{D_-}$  is obtained. At every step of the algorithm, it has to erase the words of the new block that are already accepted by the current hypothesis and merge the states of the maximal automaton of the rest of the words according to a random ordering. If the current hypothesis is not consistent with the new block of positive samples, it has to be run again taking into account the whole set of negative samples seen so far. The following example illustrates this implementation.

Let  $L = a^* + b^*$  and let the input sample be divided into the following blocks:  $D_+^{(1)} = \{a, b^2, a^2\}$ ,  $D_-^{(1)} = \{ab, b^2a\}$ ,  $D_+^{(2)} = \{b, a^3\}$ ,  $D_-^{(2)} = \{a^3b, a^2b\}$ ,  $D_+^{(3)} = \{\varepsilon, b^3, a^4\}$  and  $D_-^{(3)} = \{ab^2, ba\}$ .

The OIL algorithm starts considering the first block ( $D_+^{(1)}, D_-^{(1)}$ ). The maximal automaton  $MA(D_+^{(1)})$  for this block is shown in Fig. 10, in which we have randomly sorted the states.

Next, OIL makes all the possible merges following the established order under the control of  $D_-^{(1)}$ . It merges 1 with 2, 1 with 3, 1 with 4, 1 with 6 and 1 with 7. Note that 1 cannot be merged with 5 (it would accept  $b^2a \in D_-^{(1)}$ ) and neither 1 and 8, nor 5 and 8 can be merged for a similar reason. The algorithm obtains the automaton shown in Fig. 11, which is irreducible with respect to  $D_+^{(1)}$  and  $\overline{D_-^{(1)}}$ .

Once the first hypothesis has been obtained, the algorithm starts processing the second block. The maximal automaton for  $D_+^{(2)}$  with the randomly numbered states (starting with  $N_1 + 1$ , where  $N_1$  is the number of states of  $MA(D_+^{(1)})$ ) is shown in Fig. 12.

Once the new maximal automaton has been constructed, it has to be checked for consistency with the current hypothesis. At this point, the automaton shown in Fig. 11 is consistent with the negative samples  $D_-^{(2)} = \{a^3b, a^2b\}$  and since this automaton accepts  $a^3$ , this portion of the maximal automaton must be eliminated and the rest is incorporated to the hypothesis, which is  $D_+^{(2)'} = \{b\}$ . The resulting automaton is shown in Fig. 13.

Now, the algorithm will try to merge the states in this order, controlled by the set of negative samples  $D_- = D_-^{(1)} \cup D_-^{(2)}$ , that is  $D_- = \{ab, b^2, a^2b, a^3b\}$ . Therefore the merges made at this point are 1 with 12 and 5 with 10. Note that merging 1 with 10 would cause the negative sample  $ab$  to be accepted. These merges give the second hypothesis, which is depicted in Fig. 14.



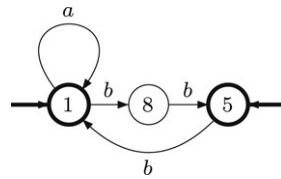


Fig. 14. The second hypothesis given by OIL, after considering the first two blocks.

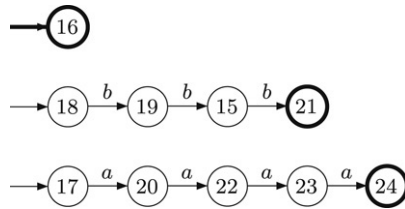


Fig. 15.  $MA(D_+^{(3)})$  with randomly numbered states.

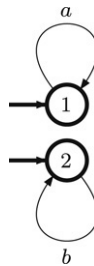


Fig. 16. Final hypothesis output by OIL.

Now, to start processing the third block of samples, OIL builds the maximal automaton and randomly extends the enumeration of the states. The automaton is shown in Fig. 15. It then checks the third block of the sample for consistency with the current hypothesis (Fig. 14).

As there are now negative samples (for example  $ab^2$ ) accepted by the hypothesis, the algorithm must process the positive samples  $D_+^{(1)} \cup D_+^{(2)} \cup D_+^{(3)}$  again. This process is controlled by the whole set of negative samples, that is,  $D_- = \{ab, ab^2, a^2b, a^3b, ab^2, ba\}$ . At this point, the algorithm uses the former enumeration of the states in the maximal automaton.

Now, states 1 and 2 cannot be merged since the resulting automaton would accept  $ab^2$ , 1 and 5 cannot be merged either since  $b^2a$  would be accepted. States 1 and 3, 1 and 4, 1 and 7, 2 and 5, and finally 2 with 8 are merged, outputting the automaton shown in Fig. 16, which recognizes the target language.

#### 4.2. Convergence and complexity of the OIL algorithm

**Proposition 20.** *The OIL algorithm identifies the family of regular languages in the limit.*

**Proof.** Let  $L$  be a language. Let  $(D_+^{(1)}, D_-^{(1)})$ ,  $(D_+^{(2)}, D_-^{(2)})$ ,  $\dots$  be a complete presentation of  $L$  in blocks. There exists  $n \geq 0$  such that  $D_+ = \bigcup_{i=1}^n D_+^{(i)}$ . As  $D_+$  is a universal sample, there exists (Proposition 16) a finite subset of negative samples of  $L$  that avoids any undesired merging of states of  $MA(D_+)$ .

As the sequence of blocks is, in fact, a complete presentation of  $L$ , the set of negative samples that is needed to prevent undesired mergings will appear after processing the  $m$  blocks, for some  $m \geq n$ . Let  $D_-$  be the set  $\bigcup_{i=1}^m D_-^{(i)}$ . Note that, after processing the first  $n$  blocks, the algorithm will output an automaton that recognizes  $L$  if it has been supplied with enough negative samples.

If convergence has not been reached before processing the block  $(D_+^{(m)}, D_-^{(m)})$ , the current automaton will accept words that are not in  $L$  and thus, it is not consistent with  $D_-$ . The algorithm then processes all the blocks up to the  $m$ -th block, but using  $D_-$  instead of  $D_-^{(i)}$ . In this case, the algorithm will converge after processing the  $n$ -th block. ■

If  $n$  is the number of blocks that the input is divided into and  $|D_+|$  (resp.  $|D_-|$ ) is the sum of the lengths of the positive (resp. negative) samples, the algorithm runs in  $O(|D_+|^2|D_-|n^2)$ .

The following example shows that different orderings of the states in the maximal automaton of the sample may lead to different final hypotheses.

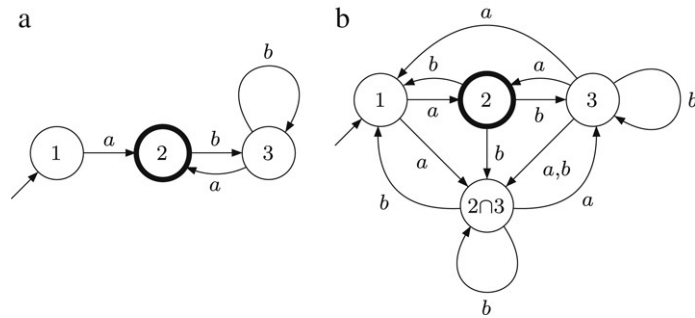


Fig. 17. Minimal deterministic finite automaton (a) and universal automaton (b) for  $L = a(bb^*a)^*$ .

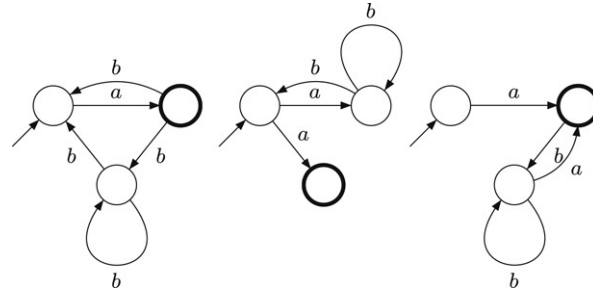


Fig. 18. Three different outputs of the algorithm with different random orderings of the states to be merged.

### 4.3. Example

Let  $L = a(bb^*a)^*$ , that is, the set of words over  $\{a, b\}$  that begin and end with  $a$  and do not contain the segment  $aa$ . The minimal deterministic finite automaton that recognizes  $L$  is shown in Fig. 17(a), whereas the universal automaton for  $L$  is shown in Fig. 17(b).

When running the algorithm with different random orderings of the states to be merged, we have obtained the automata shown in Fig. 18. Note that all the outputs are subautomata of the universal automaton for the target language. The experiments were done with enough words (of a maximum length of 18) with increments of 1000 words in each step. The experiment was run 5 times.

## 5. Conclusions

We have developed an algorithm that, on input of a complete sample, infers the class of regular languages in the limit. The generalization is obtained by merging states in the maximal automata of the sample. One of the main features of the algorithm is that the convergence takes place independently from the order in which the states are merged.

This fact permits the use of additional information to establish a particular order for the merges. It also permits several parallel runs of the algorithm to be done, so that external criteria (like the size of the automaton, or prior knowledge of the task) can be used to determine the output of the algorithm.

To prove the convergence of the algorithm we have used the concept of universal automaton of a language, which may become a useful tool in this area.

We hope that the general way in which this algorithm has been described will allow researchers to prove the convergence of future heuristics and similar algorithms to be developed for specific tasks. We are working to extend our results to prove the convergence of algorithms of this type independently of the method used to generalize the sample and output the hypotheses.

## Acknowledgements

The authors wish to thank the anonymous referees, whose useful comments have help us to make the final version of this paper more clear and understandable.

## References

- [1] J. Abela, F. Coste, S. Spina, Mutually Compatible and Incompatible Merges for the Search of the Smallest Consistent DFA, in: LNAI, vol. 3264, Springer-Verlag, 2004, pp. 28–39.
- [2] G.I. Alvarez, J. Ruiz, A. Cano, P. García, Nondeterministic Regular Positive Negative Inference NRPNI, in: J.F. Díaz, C. Rueda, A. Buss (Eds.), XXXI CLEI 05, 2005, pp. 239–249.

- [3] A. Arnold, A. Dicky, M. Nivat, A note about minimal non-deterministic automata, *Bulletin of the EATCS* 47 (1970) 166–169.
- [4] C. Carrez, On the minimalization of non-deterministic automata, *Laboratoire de Calcul de la Faculté des Sciences de L'Université de Lille*, 1970.
- [5] O. Cicehillo, S. Kremer, Beyond EDSM, in: *LNAI*, vol. 2484, Springer-Verlag, 2002, pp. 37–48.
- [6] J.H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, 1971.
- [7] F. Coste, D. Fredouille, Unambiguous automata inference by means of state merging methods, in: *LNAI*, vol. 2837, Springer-Verlag, 2003, pp. 60–71.
- [8] F. Denis, A. Lemay, A. Terlutte, Learning regular languages using RFSAs, *Theoretical Computer Science* 313 (2) (2004) 267–294.
- [9] P. García, J. Ruiz, A. Cano, G. Alvarez, Inference Improvement by Enlarging the Training Set while Learning DFAs, in: *LNAI*, vol. 3773, Springer-Verlag, 2005, pp. 59–70.
- [10] P. García, M. Vazquez de Parga, A note about mergible states in large NFA, *Bulletin of the EATCS* 87 (2005) 181–184.
- [11] E.M. Gold, Language identification in the limit, *Information and Control* 10 (1967) 447–474.
- [12] M. Gold, Complexity of automaton identification from given data, *Information and Control* 37 (1978) 302–320.
- [13] C. de la Higuera, J. Oncina, E. Vidal, Data dependent vs data independent algorithms, in: L. Miclet, C. de la Higuera (Eds.), *Grammatical Inference: Learning Syntax from Sentences*, in: *LNAI*, vol. 1147, Springer-Verlag, 1996, pp. 313–325.
- [14] J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.
- [15] K. Lang, B. Perarlmutter, R. Price, Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm, in: *LNAI*, vol. 1433, Springer-Verlag, 1998, pp. 1–12.
- [16] S. Lombardy, *Approche structurelle de quelques problèmes de la théorie des automates*, Ph.D. Thesis, Ecole N.S. des Télécommunications, 2001.
- [17] J. Oncina, P. García, Inferring regular languages in polynomial updated time, in: Pérez de la Blanca, Sanfeliú, Vidal (Eds.), *Pattern Recognition and Image Analysis*, World Scientific, 1992.
- [18] L. Polák, Minimalizations of NFA using the universal automaton, in: *LNCS*, vol. 3317, 2005, pp. 325–326.
- [19] M. Vázquez de Parga, P. García, J. Ruiz, A family of algorithms for non deterministic regular languages inference, in: *LNCS*, vol. 4049, 2006, pp. 265–275.