Learning Regular Languages Using Non Deterministic Finite Automata *

François Denis, Aurélien Lemay, Alain Terlutte
{denis, lemay, terlutte}@lifl.fr

Bât. M3, GRAPPA-LIFL
Université de Lille I
59655 Villeneuve d'Ascq Cedex
France

Abstract. We define here the *Residual Finite State Automata* class (RFSA). This class, included in the Non deterministic Finite Automata class, strictly contains the Deterministic Finite Automata class and shares with it a fundamental property : the existence of a canonical minimal form for any regular language. We also define a notion of characteristic sample S_L for a given regular language L and a learning algorithm (DeLeTe). We show that DeLeTe can produce the canonical RFSA of a regular language L from any sample S which contains S_L. We think that working on non deterministic automata will allow, in a great amount of cases, to reduce the size of the characteristic sample. This is already true for some languages for which the sample needed by DeLete is far smaller than the one needed by classical algorithms.

Key words : regular inference, non deterministic automata.

1 Introduction

Regular language learning is still one of the major thema of the grammatical inference field. This class of language, the simplest in the Chomsky hierarchy, is neither efficiently learnable in the Gold Model [Pit89], nor in the Valiant one [KV94]. Nevertheless, this class is polynomially learnable from given data [Gol78] : one can build for each regular language L a sample S_L in polynomial time relatively to the size of the smallest Deterministic Finite Automata (DFA) recognizing L and *characterizing* L in the way that L can be found back from any sample S containing S_L. We could think that this theoritical result would have few consequences since nothing assures that a *natural* sample contains S_L. But some learning algorithms in this last model, like RPNI made by Oncina and Garcia [OG92], are already interesting from an experimental point of view : correctly adapted and with good heuristics, they can turn into powerful algorithms [LPP98]. Can we go further ? The RPNI algorithm calculates a *deterministic*

* This work was partially supported by "Motricité et Cognition : Contrat par objectif région Nord/Pas-de-Calais"

automaton compatible with the sample in polynomial time. But it is a well known fact that regular languages have representations much cheaper in term of size : the minimal DFA recognizing the language $\Sigma^*0\Sigma^n$ has $O(2^n)$ states while this language is described here by a regular expression with $O(\log n)$ symbols. Looking after a non deterministic finite automaton (NFA) could seem to be a promising way but it has been proved that NFA are not polynomially identifiable from given data [Hig97]. That is nevertheless the way we will explore here.

We define a sub-class of NFA, the Residual Finite States Automata (RFSA) that has some good properties of DFA (existence of a canonical representation) and NFA (often concise minimal representation), but also some of their drawbacks. We will show how to associate with a DFA A a *characteristic* sample S_A of polynomial *cardinal* and we will define an algorithm (DeLeTe) that builds the canonical RFSA from any sample containing S_A. The cardinal of S_A is, in any cases, of the same order than the cardinal of the sample needed by RPNI, and it may be exponentially smaller in best cases. For example, RPNI needs $O(2^n)$ examples to learn the language $\Sigma^*0\Sigma^n$ although DeLeTe can learn it from sample of $O(n^2)$ words. However, this new algorithm is not a learning algorithm from given data, we will show that it is probably not a serious problem in a PAC context (Probably Approximatively Correct [Val84]) where samples are produced using a probability distribution, where the allowed running time depends on the length of the drawn examples and where the returned hypothesis can be an approximation of the target. After some preliminaries on languages, automata and the learning model from given data presented in Sect. 2, we introduce the Residual Finite States Automata in Sect. 3. Then we show how to associate a representative set with any regular language in Sect. 4.1, we present the DeLeTe algorithm in Sect. 4.2 and we demonstrate the main result of this article in Sect. 4.4. Then, we comment this result and argue that it seems promising.

2 Preliminary

2.1 Languages and Automata

Let Σ be a finite alphabet and Σ^* be the set of words built on Σ. We note ε the empty word and $|u|$ the length of a word u of Σ^*. We assume that words on Σ^* are ordered the following way : $u < v$ iff $[|u| < |v|$ or $(|u| = |v|$ and u is before v in the lexicographical order)]. A language is a subset of Σ^*. If L is a language, we note $pref(L) = \{u \in \Sigma^* | \exists v \in \Sigma^*$ such that $uv \in L\}$. A Non deterministic Finite Automaton (NFA) is a quintuplet $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of terminal states and δ is a *(partial) transition function* defined from a subset of $Q \times \Sigma$ to 2^Q. We also note δ the extended transition function defined on (a subset of) $2^Q \times \Sigma^*$. A language L is *regular* if there exists a NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ such that $L = \{u \in \Sigma^* | \delta(Q_0, u) \cap F \neq \emptyset\}$. We note REG the set of regular languages. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA and q a state of A, we note L_q the language $L_q = \{u \in \Sigma^* | \delta(\{q\}, u) \cap F \neq \emptyset\}$. An automaton is said *Deterministic* (DFA) if Q_0 contains only one element, and if, for each state q and each letter

x, $\delta(q,x)$ contains at most one element. A finite automaton A is *trimmed* if every state is accessible and if, from each state, we can access to a terminal state. Any non empty regular language is accepted by a unique (with respect to an isomorphism) minimal deterministic trimmed automaton. If L is a regular language, and if u is a word of Σ^*, we note $u^{-1}L$ the *residual* language of L by u defined by $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$. According to the Myhill-Nerode theorem, the set of distincts residual languages of a regular language is finite. Furthermore, if $A = \langle \Sigma, Q, \{q_0\}, F, \delta \rangle$ is the minimal DFA recognizing L, $u^{-1}L \rightarrow \delta(q_0, u)$ is a bijection from the set of residual states of L to Q. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA and L the language recognized by A. We define $A^s = \langle \Sigma, Q, Q_0^s, F, \delta^s \rangle$ where $Q_0^s = \{q \in Q | L_q \subseteq L\}$ and $\delta^s(q,x) = \{q' \in Q | L_{q'} \subseteq x^{-1}L_q\}$ for any state q and any letter x. A^s is said to be the *saturated* of A and we say that an automaton A is *saturated* if it is isomorph with A^s. One can show that an automaton and its saturated recognize the same language [DLT00b].

2.2 Learning Languages

Our framework is regular language learning from examples. If L is a language defined on the alphabet Σ, an *example* of L is a pair (u,e) where $u \in \Sigma^*$ and $e = 1$ if $u \in L$ (*positive* example), and $e = 0$ otherwise (*negative* example). A *sample* S of L is a finite set of examples of L. We note $S^+ = \{u | (u,1) \in S\}$ and $S^- = \{u | (u,0) \in S\}$. The *size* of a sample S (noted $||S||$) is the sum of the length of all the words in it.Gold showed that the class of regular languages is polynomially learnable from given data [Gol78]. Goldman and Mathias introduced a learning model with teacher [GM96] that De la Higuera extended to languages and showed equivalent to the learning model from given data [Hig97]. To show that the REG class, represented by DFA, is polynomially learnable from given data is equivalent to show that there exists two algorithms $\mathcal{T}$ and $\mathcal{L}$ such that for any regular language L with a minimal DFA A :

- $\mathcal{T}$ takes A as input and gives in polynomal time a sample S_L of L (of polynomial size with respect to the size of A)
- for each sample S of L, $\mathcal{L}$ takes S as input and gives in polynomial time a DFA compatible with S, equivalent with A if S contains S_L.

The RPNI algorithm ([OG92]) is a learning algorithm for the regular language class in this model. RPNI builds from a sample S the most specific DFA recognizing S^+ (the prefix tree), then examines if it is possible to merge two states, beginning with the root, while keeping the automaton deterministic and consistant with the sample. The purpose of the characteristic sample is here to avoid that two states are merged when they do not have to be. So, that is not a surprise that RPNI can identify an automaton from any sample containing its characteristic sample : but the main interesting point of this algorithm is that it keeps working in a proper way in "deteriorated" mode, that is when the sample does not contain the characteristic sample. It is here necessary to define heuristics precising whether two states can be merged or not, even when the conditions to do it are not fulfilled.

3 RFSA

Definition 1. *A Residual Finite State Automaton (RFSA) is a finite automaton* $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ *such that,*

$$\forall q \in Q, \exists u \in \Sigma^* \text{ such that } L_q = u^{-1}L.$$

Deterministic automata are obviously RFSA, but some non deterministic finite automata are not (cf. figure 1).

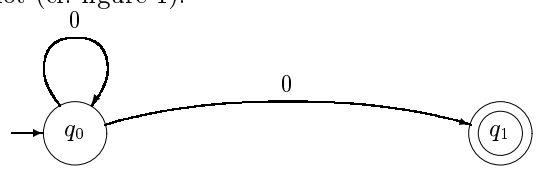

Fig. 1. This automaton is not a RFSA as $L_{q_1} = \{\varepsilon\}$ is not a residual of 0^+.

One of the major interests of the RFSA class is that we can define a notion of *canonical* RFSA associated with a language.

Definition 2. *Let L be a regular language. A residual language $u^{-1}L$ is said to be* prime *if it is not union of other residual languages of L, that is if*

$$\bigcup \{v^{-1}L \mid v^{-1}L \subsetneq u^{-1}L\} \subsetneq u^{-1}L.$$

We say that a residual language is composed *if it is not prime.*

Example 1. Let $L = a^* + b^*$. This language possesses 3 non empty residuals : L, a^* and b^*. The first one is composed, and the other two are prime.

Definition 3. *Let L be a regular language on Σ and let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be the automaton defined by :*

- *Q is the set of prime residual non-empty languages of L*
- *Q_0 is the set of prime residual languages included in L.*
- *F is the set of prime residual languages of L containing ε.*
- *δ is defined by $\delta(u^{-1}L, x) = \{v^{-1}L \in Q | v^{-1}L \subseteq ux^{-1}L\}$*

We say that A is the canonical RFSA of L.

One can show that the canonical RFSA of a regular language L is an RFSA, that it is saturated, that it recognizes L, that it is minimal in the number of states and that any other minimal saturated RFSA recognizing L is isomorph with it [DLT00b]. As a result, the RFSA class shares at least two important properties with DFA : states are defined by residual languages and there exists a canonical element. On the other hand, canonical RFSA of a regular language can be much smaller than the minimal DFA that recognizes this language : that is the case of the language $\Sigma^* 0 \Sigma^n$, often showed in the litterature to illustrate the fact that there can be an exponential gap between the size of the minimal DFA and the size of a minimal NFA recognizing a language.

Fig. 2. canonical RFSA accepting the language $a^* + b^*$.

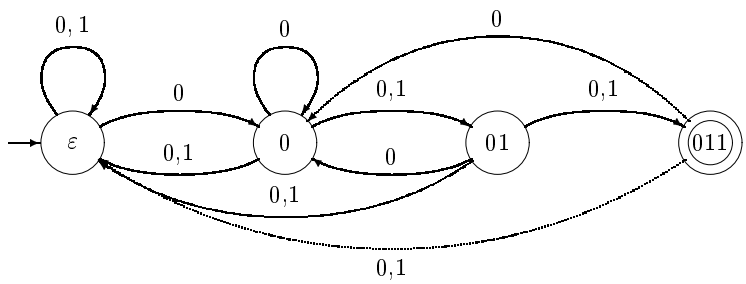

Fig. 3. The canonical RFSA of $\Sigma^*0\Sigma^2$: a minimal NFA has 4 states too, although its minimal DFA has 8 states. The 4 prime residual languages are $L = \Sigma^*0\Sigma^2$, $0^{-1}L = L \cup \Sigma^2$, $01^{-1}L = L \cup \Sigma$, $011^{-1}L = L \cup \{\varepsilon\}$. The other 4 residuals are composed.

4 Learning Regular Languages Using RFSA

4.1 A Characteristic Sample

Let $A = (\Sigma, Q, q_0, F, \delta)$ be a minimal trimmed DFA. For every state q of A, we define u_q as being the smallest word of Σ^* such that $\delta(q_0, u_q) = q$. As a consequence, we have $u_{q_0} = \varepsilon$. We assume that $Q = \{q_0, \ldots, q_n\}$ is ordered using u_q. In other words, $q_i < q_j$ iff $u_{q_i} < u_{q_j}$.

We note $PC(L) = \{u_q | q \in Q\}$ and $U(L) = PC(L) \cup \{ux | u \in PC(L),\ x \in \Sigma\}$.

Definition 4. *We say that a sample S is characteristic for the minimal DFA if*

- *for every state q of Q, there exists a word of the form $u_q v$ in S^+*
- *for every state q and q'' of Q and every letter x, if $q' = \delta(q, x)$ then*
  - *if $L_{q'} \setminus L_{q''} \neq \emptyset$ then there exists w such that $u_q xw \in S^+$ and $u_{q''} w \in S^-$*
  - *if $L_{q''} \setminus L_{q'} \neq \emptyset$ then there exists w such that $u_q xw \in S^-$ and $u_{q''} w \in S^+$*

Let S be a sample, u and v be 2 words of $pref(S^+)$. We note :

- $u \simeq v$ if no word w exists such that $uw \in S^+$ and $vw \in S^-$ or the opposite
- $u \prec v$ if no word w exists such that $uw \in S^+$ and $vw \in S^-$

As, in a learning context, we do not know residual languages but only the learning sample, we will use these relations to estimate relations between residual languages. We show here that it is reasonable to do that if S is characteristic for the minimal DFA, and if u and v are in $U(L)$.

Lemma 1. *If $u, v \in U(L)$ and if S is a sample of L then*

$$u \simeq v \Leftarrow u^{-1}L = v^{-1}L,\ u \prec v \Leftarrow u^{-1}L \subseteq v^{-1}L$$

If S is characteristic for the minimal DFA A, then the converses are true.

Proof. The if-parts are straightforward. Let us assume that S is characteristic for the DFA A and that $u^{-1}L \setminus v^{-1}L \neq \emptyset$. Then u and v are distincts and one of those two words at least is different from ε. Let us assume that $u = u_1x$. We verify that a state q_1 exists such that $u_{q_1} = u_1$. There exists a word w such that $uw = u_1xw \in S^+$ and $vw \in S^-$, thus $u \not\simeq v$ and $u \not\prec v$. The other three cases ($u^{-1}L \setminus v^{-1}L \neq \emptyset$ and $v = v_1x$, $v^{-1}L \setminus u^{-1}L \neq \emptyset$ and $v = v_1x$, $v^{-1}L \setminus u^{-1}L \neq \emptyset$ and $u = u_1x$) are treated in a same way.

Let us introduce a new notation :

- if q is a prime state, let v_q be the smallest word of L_q such that, for every state q', $L_{q'} \subsetneq L_q \Rightarrow v \notin L_{q'}$.
- if q is a composed state, there exists a highest index k such that
  - $L_q = \bigcup\{L_{q_i} | L_{q_i} \subsetneq L_q \text{ and } i \leq k\}$ and
  - $\bigcup\{L_{q_i} | L_{q_i} \subsetneq L_q \text{ and } i < k\} \subsetneq L_q$.

  We then define v_q as being the smallest word of $L_q \setminus \{L_{q_i} | L_{q_i} \subset L_q,\ i < k\}$.

Definition 5. *We say that a sample S is characteristic for the canonical RFSA of a regular language L if it is characteristic for the minimal DFA of L and if*

- *$v_{q_0} \in S^+$ and for every state q such that $L_q \subseteq L \setminus \{v_{q_0}\}$, we have $u_q v_{q_0} \in S^-$,*
- *for every state q and for every letter x, if $q' = \delta(q, x)$ then*
  - *$u_q x v_{q'} \in S^+$*
  - *and for every state q'' such that $L_{q''} \subseteq L_{q'} \setminus \{v_{q'}\}$, $u_{q''} v_{q'} \in S^-$*

Let S be a sample, $u_0, \ldots, u_n \in pref(S)$. We note $u_0 = \oplus\{u_1, \ldots, u_n\}$ if $u_i \prec u_0$ for every $i = 1 \ldots n$ and if for every word v, $u_0 v \in S^+$ implies that there exists at least one index $i > 0$ such that $u_i v \notin S^-$.

Lemma 2. *If $u_0, \ldots, u_n \in U(L)$ and if S is a sample of L then*

$$u_0^{-1}L = \cup\{u_i^{-1}L | i = 1 \ldots n\} \Rightarrow u_0 = \oplus\{u_1, \ldots, u_n\}$$

Furthermore, if we assume that S is characteristic for the canonical RFSA and that $U = \{u_1, \ldots, u_n\}$ verify the following property :

$$\forall v \in PC(L), v \prec u_0 \text{ and } \exists u \in U\ v \leq u \Rightarrow v \in U$$

Then the reciprocal is true.

Proof. The necessary part of this statement is straightforward. Let us assume that $u_0 = \oplus\{u_1, \ldots, u_n\}$. As S is characteristic for the DFA, we have a $u_i^{-1}L \subseteq u_0^{-1}L$ for each $i = 1 \ldots n$ because of the previous lemma. If we had $\cup\{u_i^{-1}L | i = 1 \ldots n\} \subsetneq u_0^{-1}L$ we would also have $u_i^{-1}L \subseteq u_0^{-1}L \setminus \{v_{\delta(q_0, u_0)}\}$ for every index i and so $u_i v_0 \in S^-$, which is wrong because of our hypothesis

Remarks

- $\prec$, $\simeq$ and $\oplus$ operators have the intended properties as long as we require the good conditions for the working sample and as long as we only use words of $U(L)$. This means that, if we have a characteristic sample, we can use those operators to evaluate relations between residual languages ; that is actually what we will do in the learning algorithm presented here,
- of course, there exists characteristic samples for the minimal DFA of polynomial size with respect to the size of this DFA ; we can also observe that the RPNI algorithm can find back this minimal DFA from such a sample,
- there also exists characteristic samples for RFSA whose *cardinal* is polynomial with respect to the size of the minimal DFA recognizing the language L ; however, it happens that the smallest characteristic sample for the canonical RFSA recognizing a language L contains words of length exponential with respect to the size of the minimal DFA of this language. Let us note $p_1, \ldots, p_n$ the first n prime numbers and let us define for every index i the language $L_i = \{\varepsilon\} \cup \{a^k | p_i \text{ does not divide } k\}$. Let us introduce $n+1$ extra letters $\{x_0, \ldots, x_n\}$ and let us consider the language $L = x_0 a^* \cup \bigcup \{x_i L_i | i = 1 \ldots n\}$. The residual $x_0^{-1} L$ is not the reunion of the residuals $x_i^{-1} L$ but the first word that can show that is $a^{p_1 \cdots p_n}$ and its length is exponential with respect to the size of the minimal DFA recognizing L.

4.2 The *DeLeTe* Algorithm

We present here a grammatical inference algorithm that builds a NFA from a sample of a target language L. We show that if this sample is characteristic, the resulting automaton is the canonical RFSA of L. This algorithm is divided in 4 main phases. Suppose that the input sample is characteristic :

- the *marking* phase marks some prefixes of the input sample : each marked word is in $PC(L)$ (lemma 3) and each state of the canonical RFSA of L will correspond to one of those words.
- in the *saturation* phase, we build the prefix tree automaton corresponding to the positive part of the input sample. We establish the $\prec$ relation between the marked prefixes and their successors. From our hypothesis and lemmas 1 and 2, these relations are correct and the resulting automaton contains the canonical RFSA of L,
- in the *cleaning* phase, we delete un-marked states. Again, we will show that having a good sample allows us to do that without changing the language recognized by the automaton,
- in the *reduction* phase, we delete marked states that are recognized as composed by the algorithm.

In these conditions, the resulting automaton is the canonical RFSA of L. If the sample is not characteristic, we can at least show that the resulting automaton is consistant with the sample.

Input : a sample S of a language L (we suppose S^+ is not empty).
We note $Q_S = \{u_0, \dots, u_n\}$ prefixes of S^+ ordered using the usual order.

Marking phase
Initialisation : let $\dot{Q} = \{\varepsilon\}(= \{u_0\})$
For i from 1 to n **Do**
  If $\exists j < i$ and $x \in \Sigma$ such that $u_i = u_j x$ and $u_j \in \dot{Q}$ **Then**
    Let $E_i = \{u_k \in \dot{Q} | k < i, u_k \prec u_i\}$
    If $u_i \neq \oplus E_i$ **Then** Add q_i to $\dot{Q}$.
End For
We note Q_M the set $\dot{Q}$ obtained at the end of this phase.

Saturation phase
Initialisation : We create an automaton $\dot{A} = \langle \Sigma, \dot{Q}, \dot{Q}_0, \dot{F}, \dot{\delta} \rangle$ by setting $\dot{Q} = Q_S$, $\dot{Q}_0 = \{u_0\}$, $\dot{F} = S^+$ and $\dot{\delta}(u, x) = \{ux \in Q_S\}$ for every word $u \in Q_S$ and every $x \in \Sigma$. We note $\ddot{Q} = Q_M \cup \{ux \in pref(S^+) \mid x \in \Sigma, u \in Q_M\}$
For i from 0 to n **Do**
  For j from 0 to n **Do**
    If $i \neq j$, $u_i \in Q_M$, $u_j \in \ddot{Q}$, $u_i \prec u_j$ **Then**
      If $u_j = ux$ **Then**
        Add u_i to $\dot{\delta}(u, x)$ if it does not imply losing consistance with S^-
      If $u_j = \varepsilon$ **Then**
        Add u_i to $\dot{Q}_0$ if it does not imply losing consistance with S^-
  End For
End For
We call $A_s = \langle \Sigma, Q_S, Q_{0S}, F_S, \delta_S \rangle$ the automaton $\dot{A}$ obtained after this phase.

Cleaning phase
Initialisation : We set $\dot{A} = \langle \Sigma, \dot{Q}, \dot{Q}_0, \dot{F}, \dot{\delta} \rangle = A_S$
For every states $u \in \dot{Q} \setminus Q_M$, **Do**
  Suppress u of $\dot{Q}$ if it does not imply losing the consistance with S^+
End For
We note $A_C = \langle \Sigma, Q_N, Q_{0N}, F_N, \delta_N \rangle$ the automaton $\dot{A}$ obtained after this phase.

Reduction phase
Initialisation : We take $\dot{A} = \langle \Sigma, \dot{Q}, \dot{Q}_0, \dot{F}, \dot{\delta} \rangle = A_C$
For every states $u_i \in \dot{Q}$ **Do**
  If $\forall v$ such that $u_i v \in S^+$, $\exists u_k \in \dot{Q}$ such that $u_k \prec u_i$, $u_k \neq u_i$ and $u_k v \notin S^-$
  Then Suppress u_i of $\dot{Q}$ if it does not imply losing the consistance with S^+
End For
We note $A_R = \langle \Sigma, Q_R, Q_{0R}, F_R, \delta_R \rangle$ the automaton $\dot{A}$ obtained after this phase.

Output : The automaton A_R.

4.3 Example

Let $L = \Sigma^* 0 \Sigma$. The minimal DFA that recognizes L has 4 states corresponding to the residual languages L, $0^{-1}L = L \cup \Sigma$, $00^{-1}L = L \cup \Sigma \cup \varepsilon$ and $01^{-1}L = L \cup \varepsilon$. All of them are prime except $00^{-1}L = L \cup 0^{-1}L \cup 01^{-1}L$.

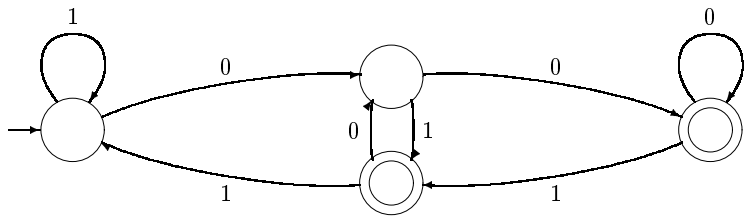

Fig. 4. minimal DFA recognizing L

We have $PC(L) = \{\varepsilon, 0, 00, 01\}$ and $U(L) = \{\varepsilon, 0, 1, 00, 01, 000, 001, 010, 011\}$.

Let us study the behaviour of Delete on the following sample :

$$S^+ = \{00, 01, 000, 100, 0000, 0100, 01000, 01100\}$$

$$S^- = \{\varepsilon, 0, 1, 10, 010, 011, 0110\}$$

We first mark the words. We can observe for instance that 1 is not marked, as we have $1 \simeq \varepsilon$. At the end of the marking phase, we have $Q_M = \{\varepsilon, 0, 00, 01\}$ and $\ddot{Q} = \{\varepsilon, 0, 1, 00, 01, 000, 010, 011\}$.

Between words of Q_M and $\ddot{Q}$, we've got the following non trivial relations : for every u, $\varepsilon \prec u$, $000 \simeq 00$, $010 \simeq 0$, $011 \simeq \varepsilon$, $1 \simeq \varepsilon$, $0 \prec 00$, $01 \prec 00$, $0 \prec 000$, $01 \prec 000$. With those relations, we can build the automaton of figure 5 at the end of the saturation phase.

At the cleaning phase, we suppress every un-marked state. As the state 00 is composed of all the others, we can suppress it during the reduction phase and we obtain the automaton of figure 6 at the end of the reduction phase. This automaton is the canonical RFSA of L.

4.4 Results

We suppose in this whole section that S is a characteristic sample of the RFSA.

Lemma 3. *For every word $u \in pref(S^+)$, we have*

- *if $u \in Q_M$ then $u \in PC(L)$*
- *if $u \in PC(L)$, then there exists $u_1, \ldots, u_l \in Q_M$ such that $u_i \leq u$ for every index i and such that $u = \oplus\{u_1, \ldots, u_l\}$.*

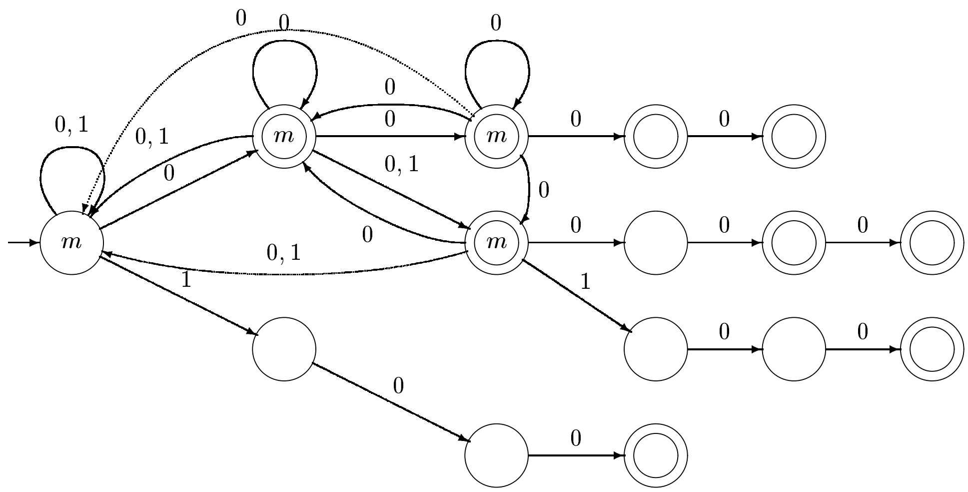

Fig. 5. Output Automaton at the end of the saturation phase

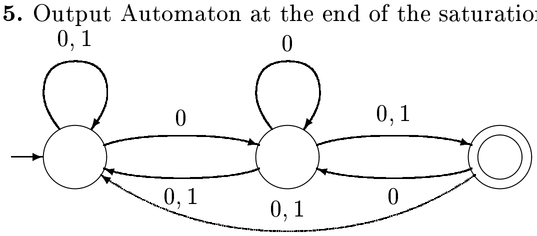

Fig. 6. Output automaton of DeLeTe

One can prove both properties at the same time using a recurrence on the length of u (see full proof in [DLT00a]).

This lemma does imply two main results : first, as every marked word u is in $PC(L)$, the $\prec$ and $\oplus$ relations between those words *do* correspond to real inclusions and union relations between their representative languages, that is, if u and v are in Q_M, $u \prec v \Rightarrow u^{-1}L \subseteq v^{-1}L$; second, every prime word of $PC(L)$ is in Q_M.

We can then show that the automaton A_S obtained after the saturation phase has an essential property : the language L_S recognized by A_S is exactly equal to the target language L. We can prove that using the following lemma.

Lemma 4. $\forall u \in Q_M$, $u^{-1}L_S = u^{-1}L$.

Sketch of proof : (see full proof in [DLT00a])

The proof is mainly based on the fact that, due to the precedent lemma, if we have two words u and v in Q_M such that $u \prec v$, then we have $u^{-1}L \subseteq v^{-1}L$.

We first have to show that consistance tests in the algorithm are useless when the sample has the conditions required here, and that we can neglect them in those proofs.

We also have to show that our transition function is correct, that is if $u \in \delta_S(v, x)$, then we have $u^{-1}L \subseteq (vx)^{-1}L$. That can be proved using the fact that

it is true in the prefix tree, and all changes in the transition function keep this property.

In particular, this implies that $\forall u \in Q_M$, $u^{-1}L_S \subseteq u^{-1}L$.

Furthermore, we can show that the automaton A_S contains all the states and transitions of the canonical RFSA of L. So, we can conclude that, at least for words $u \in Q_M$ such that $u^{-1}L$ is a prime residual of L, we have $u^{-1}L \subseteq u^{-1}L_S$. That is not too hard a step to also show it for every other states of Q_M.

□

Theorem 1. *If we give a characteristic sample for the canonical RFSA of a language L as input to the DeLeTe algorithm, the automaton given in output is the canonical RFSA of L.*

Sketch of proof : (see full proof in [DLT00a]) We already know that the automaton A_S contains the states and the transitions of the canonical RFSA of L, and it is not hard to prove that the cleaning phase and the reduction phase suppress from this automaton all the states that are not prime. Furthermore, we can show that all the remaining transitions have the properties required in the canonical RFSA and that the remaining initial states are the initial states of the canonical RFSA. □

Despite of this theorem, *DeLeTe* is not a learning algorithm for regular language from given data as it can happen that the smallest characteristic sample for the canonical RFSA of the target language contains a word whose length is exponential with respect to the size of the minimal DFA recognizing it.

5 Remarks and Conclusion

Using DFA to represent regular languages is too strong a constraint since many regular languages possess shorter representations. That is the case of languages like $\Sigma^*0\Sigma^n$ for which the minimal DFA has a number of states exponential in n. So, it is impossible for an algorithm like RPNI to have good results with those languages from samples of polynomial size in n. Nevertheless, it is quite easy to show that *DeLeTe* can find back those languages from a sample of cardinal in $O(n^2)$ and of size in $O(n^3)$. For instance, one can verify that the sample $\{01^i01^j, 101^i, 01^k, 1^i | 0 \leq i, j \leq n+1,\ 0 \leq k \leq 2n\}$ correctly labeled, of size $n^2+O(n)$ behaves like a strongly representative sample in the sense that *DeLeTe* calculates exactly the target language from all the examples it contains. The reason of this good behaviour is that those languages have few prime residuals and that they are defined by short words.

But *DeLeTe* is not built to succeed on those "academic" languages. The ideas we propose here shares the same philosophy as the one that motivated the conception of RPNI or that explains its performances. In ideal conditions, it is an algorithm of exact learning ; in less ideal conditions (but not designed to make it fail), it is still a good algorithm for approximative learning. This can surely be explained by the fact that, if an incorrect merge has been made, it is because there was no example to forbid it, and that it could probably

have been made anyway without important consequences. This remark is also true when examples are distributed using a probability distribution and when asked performances are at the same time approximative and relative to the used distribution, that is in the PAC framework. This analysis is still basic and we think a deeper analysis is still to be done.

The previous remark can be used in our case too. We said in the introduction that the representative sample of a DFA can contain words of exponential length, but in a PAC learning context, it is probably not a serious problem. Indeed, those long words are used, in an exact learning context, to be sure not to replace a residual by the reunion of the residuals it contains when we should not. But if no example forbid that reunion, it is probably not serious to suppose we could do it. All of this need to be formalized and precised : we'd like to find an algorithm that, when examples are distributed using a given probability, gives us a RFSA near to the target relatively to this distribution - that is still up to do.

We think the ideas exposed here are original and promising. The present paper is a first step. It shows that it is relevant to study grammatical inference of regular languages using RFSA representation. This work will be carry on.

References

[DLT00a] F. Denis, A. Lemay, and A. Terlutte. Learning regular languages using non deterministic finite automata. Technical Report 7, 2000.

[DLT00b] F. Denis, A. Lemay, and A. Terlutte. Les automates finis à états résiduels (afer). Technical report, ftp://ftp.grappa.univ-lille3.fr/pub/reports/afer.ps.gz, 2000.

[GM96] S. A. Goldman and H. D. Mathias. Teaching a smarter learner. *Journal of Computer and System Sciences*, 52(2):255–267, 1996.

[Gol78] E.M. Gold. Complexity of automaton identification from given data. *Inform. Control*, 37:302–320, 1978.

[Hig97] Colin De La Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27:125–137, 1997.

[KV94] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[LPP98] K. J. Lang, B. A. Pearlmutter, and R. A. Price. Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. *Lecture Notes in Computer Science*, 1433:1–12, 1998.

[OG92] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. In *Pattern Recognition and Image Analysis*, pages 49–61, 1992.

[Pit89] L. Pitt. Inductive Inference, DFAs, and Computational Complexity. In *Proceedings of AII-89 Workshop on Analogical andInductive Inference; Lecture Notes in Artificial Intelligence 397*, pages 18–44, Heidelberg, October 1989. Springer-Verlag.

[Val84] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.