Residual Finite State Automata

François Denis, Aurélien Lemay, and Alain Terlutte

Bât. M3, GRAPPA-LIFL, Université de Lille I
59655 Villeneuve d'Ascq Cedex, France
{denis,lemay,terlutte}@lifl.fr
http://www.grappa.univ-lille3.fr

Abstract. We introduce a subclass of non deterministic finite automata (NFA) that we call Residual Finite State Automata (RFSA): a RFSA is a NFA all the states of which define residual languages of the language it recognizes. We prove that for every regular language L, there exists a unique RFSA that recognizes L and which has both a minimal number of states and a maximal number of transitions. Moreover, this canonical RFSA may be exponentially smaller than the equivalent minimal DFA but it also may have the same number of states as the equivalent minimal DFA, even if minimal equivalent NFA are exponentially smaller. We provide an algorithm that computes the canonical RFSA equivalent to a given NFA. We study the complexity of several decision and construction problems linked to the class of RFSA: most of them are PSPACE-complete.

1 Introduction

Regular languages and finite automata have been extensively studied since the beginning of formal language theory. Representation of regular languages by means of Deterministic Finite Automata (DFA) has many nice properties: there exists a unique minimal DFA that recognizes a given regular language (minimal in number of states and unique up to an isomorphism); each state q of a DFA A defines a language (composed of the words which lead to a final state from q) which is a natural component of the language L recognized by A, namely a *residual language* of L. One of the major drawbacks of DFA is that they provide representations of regular languages whose size is far to be optimal. For example, the regular language $\Sigma^*0\Sigma^n$ is represented here by a regular expression whose size is $O(\log n)$ while its minimal DFA has about 2^n states. Using Non deterministic Finite Automata (NFA) rather than DFA can drastically improve the size of the representation: the minimal NFA which recognizes $\Sigma^*0\Sigma^n$ has $n+2$ states. However, NFA have none of the two above-mentioned properties: languages associated with states have no natural interpretation and two minimal NFA can be not isomorphic.

In this paper, we study a subclass of non deterministic finite automata that we call Residual Finite State Automata (RFSA). By definition, a RFSA is a NFA all the states of which define residual languages of the language it recognizes. More precisely, a NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ is a RFSA if for every state q in Q there exists a word u such that uv is recognized by A if and only if reading v, a final state can be reached from q. Clearly, all DFA are RFSA but the converse is false.

We prove that among all the RFSA which recognize a given regular language, there exists a unique element which has both a minimal number of states and a maximal

A. Ferreira and H. Reichel (Eds.): STACS 2001, LNCS 2010, pp. 144–157, 2001.

number of transitions. This canonical RFSA may be exponentially smaller than the equivalent minimal DFA (for example, the canonical RFSA which recognizes $\Sigma^*0\Sigma^n$ has $n+2$ states); but it may also have the same number of states as the equivalent minimal DFA, even if minimal equivalent NFA are exponentially smaller. Another approach of canonical NFA can be found in [Car70] and [ADN92].

It is well known that for a given DFA A recognizing a language L, if we first construct the mirror automaton $\overline{A}$ and then, the deterministic automaton equivalent to $\overline{A}$ using the standard subset construction technique, we obtain the minimal DFA for $\overline{L}$. We prove a similar property for RFSA. This property provides an algorithm which computes the canonical RFSA equivalent to a given NFA. Unfortunately, we also prove that this construction problem is PSPACE-complete, as most of the constructions we define in this paper.

In section 2, we recall classical definitions and notations about regular languages and automata. We define RFSA in section 3 and we study their properties in section 4. In particular, we introduce the notion of canonical RFSA. We provide a construction of the canonical RFSA from a given NFA in section 5. In section 6, we study some particular (and pathological) RFSA. Section 7 is devoted to the study of the complexity of our constructions. Finally, we conclude by indicating where this work originates from and by describing some of its applications in the field of grammatical inference.

2 Preliminaries

In this section, we recall some definitions concerning finite automata. For more information, we invite the reader to consult [HU79,Yu97].

2.1 Automata and Languages

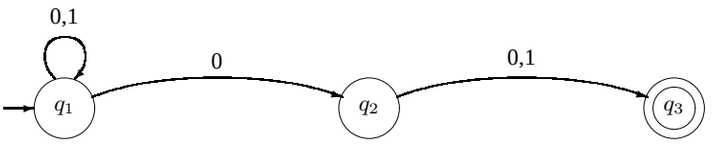

Fig. 1. A_1 Automaton Recognizes $\Sigma^*0\Sigma$ but Is neither a DFA nor a RFSA

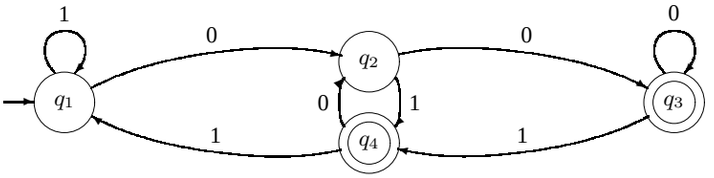

Fig. 2. A_2 Is the Minimal DFA Recognizing $\Sigma^*0\Sigma$.

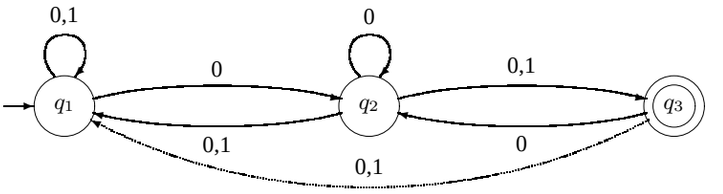

Fig. 3. A_3 Is a RFSA Recognizing $\Sigma^* 0 \Sigma$.

Let Σ be a finite alphabet, and let Σ^* be the set of words on Σ. We note ε the empty string and $|u|$ the length of a word u in Σ^*. A language is a subset of Σ^*.

A *non deterministic finite automaton* (NFA) is a quintuple $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ 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. δ is the *transition function* of the automaton defined from a subset of $Q \times \Sigma$ to 2^Q. We also note δ the extended transition function defined from a subset of $2^Q \times \Sigma^*$ to 2^Q by:

$\delta(\{q\}, \varepsilon) = \{q\}$,
$\delta(\{q\}, x) = \delta(q, x)$,
$\delta(Q', u) = \cup\{\delta(\{q\}, u) | q \in Q'\}$ and
$\delta(\{q\}, ux) = \delta(\delta(q, u), x)$

where $Q' \subseteq Q, x \in \Sigma, q \in Q$ and $u \in \Sigma^*$.

A NFA is *deterministic* (DFA) if Q_0 contains exactly one element q_0 and if $\forall q \in Q$, $\forall x \in \Sigma$, $Card(\delta(q, x)) \leq 1$. A NFA is *trimmed* if $\forall q \in Q$, $\exists w_1 \in \Sigma^*$, $q \in \delta(Q_0, w_1)$ and $\exists w_2 \in \Sigma^*$, $\delta(q, w_2) \cap F \neq \emptyset$. A state q is *reachable* by the word u if $q \in \delta(Q_0, u)$.

A word $u \in \Sigma^*$ is recognized by a NFA if $\delta(Q_0, u) \cap F \neq \emptyset$ and the language L_A recognized by A is the set of words recognized by A. We denote by $Rec(\Sigma^*)$ the class of recognizable languages. It can be proved that every recognizable language can be recognized by a DFA. There exists a unique minimal DFA that recognizes a given recognizable language (minimal with regard to the number of states and unique up to an isomorphism). Finally, the Kleene theorem [Kle56] proves that the class of regular languages $Reg(\Sigma^*)$ is identical to $Rec(\Sigma^*)$.

The *mirror* of a word $u = x_1 \dots x_n$ ($x_i \in \Sigma$) is defined by $\overline{u} = x_n \dots x_1$. The mirror of a language L is $\overline{L} = \{\overline{u} | u \in L\}$. The mirror of an automaton $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ is $\overline{A} = \langle \Sigma, Q, F, Q_0, \overline{\delta} \rangle$ where $q \in \overline{\delta}(q', x)$ if and only if $q' \in \delta(q, x)$. It is clear that $\overline{L_A} = L_{\overline{A}}$.

Let L be a regular language. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA that recognizes L and let $Q' \subseteq Q$. We note $L_{Q'}$ the language defined by $L_{Q'} = \{v | \delta(Q', v) \cap F \neq \emptyset\}$. When Q' contains exactly one state q, we simply denote $L_{Q'}$ by L_q.

2.2 Residual Languages

Let L be a language over Σ^* and let $u \in \Sigma^*$. The *residual* language of L with regard to u is defined by $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$. If L is recognized by a NFA $\langle \Sigma, Q, Q_0, F, \delta \rangle$, then $q \in \delta(Q_0, u) \Rightarrow L_q \subseteq u^{-1}L$.

The Myhill-Nerode theorem [Myh57,Ner58] proves that the set of distinct residual languages of any regular language is finite. Furthermore, if $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ is the minimal DFA recognizing L, we have:

- for every non empty residual language $u^{-1}L$, there exists a unique $q \in Q$ such that $L_q = u^{-1}L$,
- $\forall q \in Q$, there exists a unique residual language $u^{-1}L$ such that $u^{-1}L = L_q$.

3 Definition of Residual Finite State Automaton

Definition 1. *A Residual Finite State Automaton (RFSA) is a NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ such that, for each state $q \in Q$, L_q is a residual language of L_A. More formally, $\forall q \in Q$, $\exists u \in \Sigma^*$ such that $L_q = u^{-1}L_A$.*

Remark: Trimmed DFA have this property, and therefore are RFSA.

Definition 2. *Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a RFSA and let q be a state of A. We say that u is a* characterizing *word for q if $L_q = u^{-1}L_A$.*

Example 1. We study here the regular language $L = \Sigma^*0\Sigma$ where $\Sigma = \{0, 1\}$. One can prove that this language is recognized by the following automata A_1, A_2 and A_3 (fig. 1, 2, 3):

- A_1 is a NFA recognizing L. One can notice that A_1 is neither a DFA, nor a RFSA. Languages associated with states are $L_{q_1} = \Sigma^*0\Sigma$, $L_{q_2} = \Sigma$, $L_{q_3} = \{\varepsilon\}$. As for every u in Σ^*, we have $uL \subseteq L$ and so, $L \subseteq u^{-1}L$, we can see that neither L_2 nor L_3 are residual languages.
- A_2 is the minimal DFA that recognizes L. This automaton is also a RFSA , we have $L_{q_1} = \Sigma^*0\Sigma$, $L_{q_2} = \Sigma^*0\Sigma + \Sigma$, $L_{q_3} = \Sigma^*0\Sigma + \Sigma + \varepsilon$, $L_{q_4} = \Sigma^*0\Sigma + \varepsilon$, so, $L_{q_1} = \varepsilon^{-1}L$, $L_{q_2} = 0^{-1}L$, $L_{q_3} = 00^{-1}L$, $L_{q_4} = 01^{-1}L$.
- A_3 is a RFSA recognizing L. Indeed, we have $L_{q_1} = \varepsilon^{-1}L$, $L_{q_2} = 0^{-1}L$, $L_{q_3} = 01^{-1}L$. One can notice that this automaton is not a DFA. This automaton is the canonical RFSA of L, which is one of the smallest RFSA (regarding the number of states) recognizing L (the notion of canonical RFSA will be described later).

Example 2. To look for a characterizing word for a state q is often equivalent to look for a word u_q that only leads to q (i.e. such that $\delta(Q_0, u_q) = \{q\}$). Nevertheless, such a word does not always exist. For example, let $L = a^*b^* + b^*a^*$.

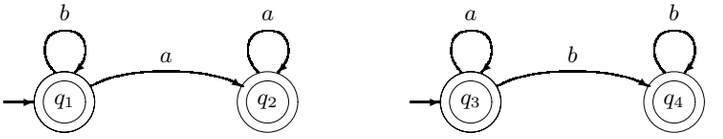

Fig. 4. A RFSA Recognizing the Language $a^*b^* + b^*a^*$.

The automaton described in figure 4 recognizes L. We have $L_{q_1} = b^*a^*$, $L_{q_2} = a^*$, $L_{q_3} = a^*b^*$, $L_{q_4} = b^*$. This automaton is a RFSA, as $L_{q_1} = b^{-1}L$, $L_{q_2} = (ba)^{-1}L$, $L_{q_3} = a^{-1}L$, $L_{q_4} = (ab)^{-1}L$. But there exists no word u such that $\delta(Q_0, u) = \{q_3\}$.

4 Properties of Residual Finite State Automata

4.1 General Properties

Definition 3. *Let L be a regular language. We say that a residual language $u^{-1}L$ is* prime *if it is not equal to the union of residual languages it strictly contains:*

$$u^{-1}L \text{ is prime if}$$

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

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

Notice that a prime residual language is not empty and that the set of distinct prime residual languages of a regular language is finite.

Proposition 1. *Let $A = \langle \Sigma, Q, Q_0, F, \delta\rangle$ be a RFSA. For each prime residual $u^{-1}L_A$, there exists a state $q \in Q$ such that $L_q = u^{-1}L_A$.*

Proof: Let $\delta(Q_0, u) = \{q_1, \ldots, q_s\}$ and let $v_1, \ldots, v_s$ be words such that $L_{q_i} = {v_i}^{-1}L_A$ for every $1 \leq i \leq s$. We have

$$u^{-1}L_A = \bigcup_{i=1 \text{ to } s} v_i^{-1}L_A.$$

As $u^{-1}L_A$ is prime, there exists some v_i such that $u^{-1}L_A = v_i^{-1}L_A = L_{q_i}$. □

As a corollary, a RFSA A has at least as many states as the number of prime residuals of L_A.

4.2 Saturation Operator

We define a *saturation* operator that allows to add transitions to an automaton without modifying the language it recognizes.

Definition 4. *Let $A = \langle \Sigma, Q, Q_0, F, \delta\rangle$ be a NFA. We call* saturated *of A the automaton $S(A) = \langle \Sigma, Q, \widetilde{Q_0}, F, \widetilde{\delta}\rangle$ with $\widetilde{Q_0} = \{q \in Q \mid L_q \subseteq L_A\}$ and $\widetilde{\delta}(q, x) = \{q' \in Q \mid xL_{q'} \subseteq L_q\}$. We say that an automaton A is saturated if $A = S(A)$.*

Lemma 1. *Let A and A' be two NFA sharing the same set of states Q. If $L_A = L_{A'}$ and if for every state $q \in Q$, $L_q = L'_q$ (L_q and L'_q being the languages corresponding to q in both automata), then $S(A) = S(A')$.*

Proof: The state q is an initial state of $S(A)$ if and only if $L_q \subseteq L_A$, that is if and only if q is an initial state of $S(A')$.

In the same way, $q' \in \widetilde{\delta}(q,x)$ in $S(A)$ if and only if $xL_{q'} \subseteq L_q$, i.e. if and only if $q' \in \widetilde{\delta}'(q,x)$ in $S(A')$. □

We note $\widetilde{L}_q = \{u \mid \widetilde{\delta}(q,u) \cap F \neq \emptyset\}$.

Proposition 2. *Let A be a NFA and let $S(A)$ be its saturated. For each state q of A, we have $L_q = \widetilde{L}_q$.*

Proof: Clearly, $L_q \subseteq \widetilde{L}_q$ as the saturated of an automaton is obtained by adding transitions and initial states. To prove the converse inclusion, we prove by induction that for every integer n and every state q

$$\widetilde{L}_q \cap \Sigma^{\leq n} \subseteq L_q.$$

If $n = 0$, the property is true as A and $S(A)$ have the same terminal states. Let $u = xv \in \widetilde{L}_q \cap \Sigma^{\leq n}$ with $n \geq 1$ and let $q' \in \widetilde{\delta}(q,x)$ such that $v \in \widetilde{L}_{q'}$. Because of our induction hypothesis, $v \in L_{q'}$. As $q' \in \widetilde{\delta}(q,x)$, we have $xL_{q'} \subseteq L_q$ and therefore $xv \in L_q$. □

Corollary 1. *Let A be a NFA and $S(A)$ be its saturated. Then A and $S(A)$ recognize the same language and $S(A) = S(S(A))$.*

Proof:

- We have $L = \cup\{L_q | q \in Q_0\} = \cup\{L_q | q \in \widetilde{Q}_0\} = \cup\{\widetilde{L}_q | q \in \widetilde{Q}_0\}$ which is equal to the language recognized by $S(A)$.
- Due to the previous point and to the proposition 2, lemma 1 can be applied on A and $S(A)$ to prove that $S(S(A)) = S(A)$; the saturated of a saturated automaton is itself.

□

Corollary 2. *If A is a RFSA then $S(A)$ is also a RFSA.*

Proof: The saturated of a RFSA is a RFSA as the saturation changes neither the languages associated with the states nor the language recognized by the automaton. □

4.3 Reduction Operator ϕ

We define a *reduction* operator ϕ that deletes states in an automaton without changing the language it recognizes.

Definition 5. *Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA, and let q be a state of Q. We note $R(q) = \{q' \in Q \setminus \{q\} \mid L_{q'} \subseteq L_q\}$. We say that q is erasable in A if $L_q = \bigcup\{L_{q'} \setminus q' \in R(q)\}$.*

If q is erasable, we define $\phi(A,q) = A' = \langle \Sigma, Q', Q'_0, F', \delta' \rangle$ where:

- $Q' = Q \backslash \{q\}$,
- $Q'_0 = Q_0$ *if* $q \notin Q_0$, *and* $Q'_0 = (Q_0 \setminus \{q\}) \cup R(q)$ *otherwise,*
- $F' = F \bigcap Q'$,
- *for every* $q' \in Q'$ *and every* $x \in \Sigma$

$$\delta'(q',x) = \begin{cases} \delta(q',x) \text{ if } q \notin \delta(q',x) \\ (\delta(q',x) \setminus \{q\}) \cup R(q) \\ \text{otherwise.} \end{cases}$$

If q is not erasable, we define $\phi(A,q) = A$.

Let $q' \in Q$ be a state different from q. We note $L_{q'}$ the language generated from q' in the automaton A and $L'_{q'}$ the language generated from q' in $A' = \phi(A,q)$.

Proposition 3. *Let A be a NFA and let q be a state of A. The automata A and $A' = \phi(A,q)$ recognize the same language and for every state $q' \neq q$, $L_{q'} = L'_{q'}$.*

Sketch of proof:

If q is not an erasable state, the proposition is straightforward. If q is an erasable state, we first prove that $L_{q'} = L'_{q'}$ using the fact that every path that allows to read a word u in A through q corresponds to a path in A' that uses an added transition and vice-versa.

Finally, we prove that $L_A = \bigcup_{q_0 \in Q_0} L_{q_0} = (\bigcup_{q_0 \in Q'_0} L'_{q_0}) = L_{A'}$.

□

Proposition 4. *The operator ϕ is an internal operator for the class of RFSA.*

Proof: Neither the language recognized by a RFSA A nor the languages associated with its states are modified by the reduction operator ϕ (c.f. previous proposition). So, languages associated with states keep being residual languages of L_A. □

We prove now that saturation and reduction operators can be swapped.

Lemma 2. *Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA and let q be a state of Q. Then the automaton $\phi(S(A), q)$ is saturated.*

Proof: We note $L'_{q'}$ (resp. $L_{q'}$) the language associated with a state q' in $\phi(S(A),q)$ (resp. in $S(A)$), δ' (resp. δ) the transition function of $\phi(S(A),q)$ (resp. in $S(A)$) and L the language recognized by the automata A, $S(A)$ and $\phi(S(A),q)$.

- If $L'_{q'} \subseteq L$ then $L_{q'} \subseteq L$ and so q' is initial in $S(A)$ and in $\phi(S(A),q)$.
- If $xL'_{q'} \subseteq L'_{q''}$ then $xL_{q'} \subseteq L_{q''}$ and so $q' \in \delta(q'',x)$ and $q' \in \delta'(q'',x)$.

□

Proposition 5. *Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA and let q be a state of Q. We have*

$$S(\phi(A,q)) = \phi(S(A),q)$$

Proof: $\phi(A, q)$ and $\phi(S(A), q)$ have the same set of states. Furthermore, languages associated with every state q' in $\phi(A, q)$ and $\phi(S(A), q)$ are identical because of previous lemmas. Because of lemma 1, $S(\phi(A, q)) = S(\phi(S(A), q))$. As $\phi(S(A), q)$ is a saturated automaton (cf lemma 2), the proposition is proved. □

Definition 6. *Let A be a NFA. If there is no erasable state in A, we say that A is* reduced.

4.4 Canonical RFSA

Definition 7. *Let L be a regular language. We define $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ the canonical RFSA of L in the following way:*

- *Σ is the alphabet of L,*
- *Q is the set of prime residuals of L, so $Q = \{u^{-1}L \mid u^{-1}L$ is prime $\}$,*
- *its initial states are prime residuals included in L, so $Q_0 = \{u^{-1}L \in Q \mid u^{-1}L \subseteq L\}$,*
- *its final states are prime residuals containing the empty word, so $F = \{u^{-1}L \in Q \mid \varepsilon \in u^{-1}L\}$,*
- *its transition function is $\delta(u^{-1}L, x) = \{v^{-1}L \in Q \mid v^{-1}L \subseteq (ux)^{-1}L\}$.*

This definition assumes that the canonical RFSA is a RFSA, we will prove this presumption below.

We have proved that the reduction operator ϕ transforms a RFSA into a RFSA, and that it could be swapped with the saturation operator. We prove now that, if A is a saturated RFSA, the reduction operator converges and that the resulting automaton is the canonical RFSA of the language recognized by A.

Proposition 6. *Let L be a regular language and let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a reduced and saturated RFSA recognizing L. A is the canonical RFSA of L.*

Proof: As A is a RFSA, every prime residual $u^{-1}L$ of L can be defined as a language L_q associated with some states $q \in Q$. As there are no erasable states in A, for every state q, L_q is a prime residual and distinct states define distinct languages. As A is saturated, prime residuals contained in L correspond to initial states of Q_0. For the same reason, we can verify that the transition function is the same as in the canonical RFSA. □

Theorem 1. *The canonical RFSA of a regular language L is a RFSA which recognizes L and which is minimal regarding the number of states.*

Proof: Let $A_0, \ldots, A_n$ be a sequence of NFA such that for every index $i \geq 1$, there exists a state q_i of A_{i-1} such that $A_i = \phi(A_{i-1}, q_i)$. Proposition 5 and 6 prove that if A_0 is a saturated RFSA and if A_n is reduced, then A_n is the canonical RFSA of the language recognized by A_0.

So the canonical RFSA can be obtained from any RFSA that recognizes L using saturation and reduction operators. Proposition 1 proves that it has a minimal number of states. □

Remark that it is possible to find a RFSA that has as many states as the canonical RFSA of L, but fewer transitions. We have the following proposition:

Theorem 2. *The canonical RFSA of a regular language L is the unique RFSA that has a maximal number of transitions among the set of RFSA which have a minimal number of states.*

Proof: Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be the canonical RFSA of a language L and let $A' = \langle \Sigma, Q', Q'_0, F', \delta' \rangle$ be a RFSA which has a minimal number of states. So, A' is reduced. From proposition 6, the saturated automaton of A' is A. Therefore, A' has at most as many transitions as A. □

5 Construction of the Canonical RFSA Using the Subset Method

In the previous section, we provided a way to build the canonical RFSA from a given NFA using saturation and reduction operators. This method requires to check whether a language is included into another one and to check whether a language is composed or not. Those checks can be very expensive, even for simple automata. We present in this section another method which stems from a classical construction of the minimal DFA of a language and which is easier to implement.

Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a NFA. The subset construction is a classical method used to build a DFA equivalent to a given NFA. It consists in building the set of reachable sets of states of A. We note $Q_{R(A)} = \{p \in 2^Q \mid \exists u \in \Sigma^* \text{ s. t. } \delta(Q_0, u) = p\}$ and we define the subset automaton $D(A) = \langle \Sigma, Q_D, Q_{D0}, F_D, \delta_D \rangle$ with

$$\begin{aligned} Q_D &= Q_{R(A)}, \\ Q_{D0} &= \{Q_0\}, \\ F_D &= \{p \in Q_D \mid p \cap F \neq \emptyset\}, \\ \delta_D(p, x) &= \delta(p, x). \end{aligned}$$

The automaton $D(A)$ is a determi-nistic automaton that recognizes the same language as A.

We remind that $\overline{L}$ (resp. $\overline{B}$) denotes the mirror of a language L (resp. of an automaton B). The following result provides a method to build the minimal DFA of L.

Theorem 3. *[Brz62] Let L be a regular language and let B be an automaton such that $\overline{B}$ is a DFA that recognizes $\overline{L}$. Then $D(B)$ is the minimal DFA recognizing L.*

We can deduce from this theorem that $D(\overline{D(\overline{A})})$ is the minimal DFA recognizing the language L_A.

We adapt the subset construction technique to deal with inclusions of sets of states. We say that a state $p \in Q_{R(A)}$ is *coverable* if there exist states $p_i \in Q_{R(A)}$, $p_i \neq p$, such that $p = \cup_i p_i$. We define the automaton $C(A) = \langle \Sigma, Q_C, Q_{C0}, F_C, \delta_C \rangle$ by

$$\begin{aligned} Q_C &= \{p \in Q_{R(A)} \mid \\ &\qquad p \text{ is not coverable } \}, \\ Q_{C0} &= \{p \in Q_C \mid p \subseteq Q_0\}, \\ F_C &= \{p \in Q_C \mid p \cap F \neq \emptyset\}, \\ \delta_C(p, x) &= \{p' \in Q_C \mid p' \subseteq \delta(p, x)\}. \end{aligned}$$

Lemma 3. *Let A be a NFA, $C(A)$ is a RFSA recognizing L_A such that all states are reachable.*

Sketch of proof: $C(A)$ can be obtained from $D(A)$ by using techniques which are similar to the ones used by the reduction operator. □

Theorem 4. *Let L be a regular language and let B be an automaton such that $\overline{B}$ is a RFSA recognizing $\overline{L}$ such that all states are reachable. Then $C(B)$ is the canonical RFSA recognizing L.*

Sketch of proof:

Let $q_i \in Q_B$, let $\overline{L}_{q_i}$ be the language associated with q_i in $\overline{B}$ and let $v_i \in \Sigma^*$ be such that $\overline{L}_{q_i} = \overline{v_i}^{-1}\overline{L}$. Let p, $p' \in Q_{R(B)}$. We prove that:

- $v_i \in L_p$ iff $q_i \in p$.
- $L_p \subseteq L_{p'}$ iff $p \subseteq p'$.
- For every state p, p_1, $p_2 \ldots p_n \in Q_{R(B)}$, $L_p = \cup_{1 \leq k \leq n} L_{p_k}$ iff $p = \cup_{1 \leq k \leq n} p_k$.

From the last three statements, we can prove that $C(B)$ can be obtained from $D(B)$ by reduction and saturation. As $D(B)$ is deterministic, and using proposition 6, $C(B)$ is the canonical RFSA of L. □

We can deduce from this proposition and from lemma 3 that $C(\overline{C(\overline{A})})$ is the canonical RFSA of L_A.

However, this construction also has some weaknesses. Indeed, it is possible to find examples for which $C(\overline{A})$ has an exponential number of states with regard to the number of states of A or $C(\overline{C(\overline{A})})$. We can observe this situation with the mirror of the automaton used in the proposition 8.

We can also observe that, if we are interested only in covering without saturation (if a state is covered, we delete it and we reload its transitions to covering states), we get a RFSA which has the same number of states (non-coverable states) and fewer transitions.

6 Results on Size of RFSA

We classically take the number of states of an automaton as a measure of its size. The canonical RFSA of a regular language has the size of the equivalent minimal DFA as an upper bound and the size of one of its equivalent minimal NFA as a lower bound. We show that both bounds can be reached even if there exists an exponential gap between these two bounds.

Proposition 7. *There exist languages for which the minimal DFA has a size exponentially larger than the size of the canonical RFSA, and for which the canonical RFSA has the same size as minimal NFA.*

Proof: $\Sigma^* 0 \Sigma^n$ languages, where n is an integer and $\Sigma = \{0, 1\}$, can illustrate this proposition.

Residuals of $L = \Sigma^* 0 \Sigma^n$ are languages $L \cup (\bigcup_{p \in P} \Sigma^p)$ where $P \subseteq \{0, \ldots, n\}$. One can observe that there exist 2^{n+1} distinct residuals. The minimal DFA recognizing this language has 2^{n+1} states. There exist only $n + 2$ prime residuals: L, $L \cup \Sigma^0$, …, $L \cup \Sigma^n$, so, the canonical RFSA of L has $n + 2$ states. □

Proposition 8. *There exist languages for which the size of the canonical RFSA is exponential with regard to the size of a minimal NFA.*

Proof: Let $A_n = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be automata such that, for $n \geq 1$

- $\Sigma = \{a, b\}$,
- $Q = \{q_i \mid 0 \leq i \leq n-1\}$,
- δ is defined by
  $\delta(q_i, a) = q_{i+1}$ (for $0 \leq i < n-1$),
  $\delta(q_{n-1}, a) = q_0$,
  $\delta(q_0, b) = q_0$,
  $\delta(q_i, b) = q_{i-1}$ (for $1 < i < n$) and
  $\delta(q_1, b) = q_{n-1}$,
- $Q_0 = \{q_i \mid 0 \leq i < n/2\}$,
- $F = \{q_0\}$.

Figure 5 represents A_4.

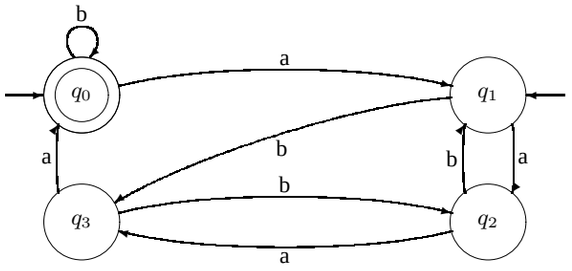

Fig. 5. An automaton A_n, $n = 4$, for which the Equivalent RFSA Is Exponentially Larger.

The mirror automata $\overline{A_n}$ are trimmed and deterministic, thus we can apply theorem 4. The automata $C(A_n)$ are canonical RFSA.

The initial state of the subset construction has $n/2$ elements. Moreover the reachable states are all the states with $n/2$ elements. So, none of them is coverable.

The canonical RFSA $C(A_n)$ are exponentially larger than the initial NFA. □

Proposition 9. *There exist languages for which the smallest characterizing word for some state has a length exponentially bigger than the number of states of the canonical RFSA.*

Sketch of proof: Let $P = \{p_1, \ldots, p_n\}$ be a set of n distinct prime numbers. We define the NFA $A_P = \langle \Sigma, Q, Q_0, F, \delta \rangle$ by:

- $\Sigma = \{a\} \cup \{b_p \mid p \in P\}$
- $Q = \{q_i^p \mid p \in P, 0 \leq i < p\}$
- $Q_0 = \{q_0^p \mid p \in P\}$

- $F = Q_0$
- δ is defined by:
  $\delta(q_i^p, a) = \{q_{(i+1) mod\ p}^p\}$
  for $0 \leq i < p, p \in P$,

  $\delta(q_i^p, b_{p'}) = \{q_i^p, q_{i+1}^p\}$
  for $0 \leq i < p-1, p, p' \in P$,

  $\delta(q_{p-1}^p, b_{p'}) = \{q_0^{p'}\}$
  for $p, p' \in P$.

The following results can be proved:

- A_P is a RFSA.
- The smallest characterizing word u_q of a state $q \in Q$ is such that $|u_q| \geq \Pi_i p_i$ which is exponential with regard to the size of A_P and therefore exponential with regard to the size of the canonical RFSA.

□

Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be a RFSA and let $q \in Q$ such that L_q is prime. There must exist a smallest word $u \in L_q$ such that $L_{q'} \subsetneq L_q \Rightarrow u \notin L_{q'}$. Next proposition proves that this word can be very long.

Proposition 10. *There exist languages for which the smallest word that proves that a state of the canonical RFSA is not composed has an exponential size with regard to the number of states of the minimal DFA.*

Proof: Let p_1,...,p_n be distinct prime numbers. For each i, $1 \leq i \leq n$, we note $L_i = \{\varepsilon\} \cup \{a^k \mid p_i \text{ is not a divisor of } k\}$. Let b_0, b_1,... b_n be distinct letters different from a. We consider the language $L = b_0 a^* \cup (\bigcup_{1 \leq i \leq n} b_i L_i)$.

We can easily build a minimal DFA for this language ; it contains $\sum p_i + n + 2$ states. The language $b_0^{-1} L = a^*$ is not an union of residuals $b_i^{-1} L$, $i \geq 1$. But the shortest word that belongs to $b_0^{-1} L \setminus \cup_{1 \leq i \leq n} b_i^{-1} L_i$ is $a^{p_1 \cdots p_n}$ and its length is exponential with regard to the size of the minimal DFA. □

7 Complexity Results about RFSA

We have defined notions of RFSA, saturated automata, canonical RFSA ; in this section, we evaluate the complexity of our constructions and of decision problems linked to them: deciding if an automaton is saturated, building the canonical RFSA of a given language, and so on ...

Classical definitions about complexity can be found in [GJ79] and complexity results about automata can be found in [HU79]. We present here simple complexity results about RFSA, proofs of which can be found in [DLT00b].

The first notion that we defined is the notion of *saturation*. As one could guess, deciding if an automaton is saturated is easier for a DFA than for a NFA.

Proposition 11. *Deciding whether a DFA is saturated is a polynomial problem. On the other hand, deciding whether a NFA is saturated is a $PSPACE$-complete problem. Building the saturated of a NFA is also a $PSPACE$-complete problem.*

The next proposition tells us that it is not practically possible, in the worst case, to check whether a NFA is a RFSA.

Proposition 12. *Deciding if a NFA is a RFSA is a $PSPACE$-complete problem.*

Building the canonical RFSA equivalent to a given NFA is an exponential problem in general, as proved by proposition 8. The next proposition tells us that, even if the starting automaton is deterministic, this problem is $PSPACE$-complete. The problem of deciding whether the saturated of a DFA is a canonical RFSA is also $PSPACE$-complete.

Proposition 13. *Deciding if the saturated of a DFA is a canonical RFSA is a $PSPACE$-complete problem. Building the canonical RFSA equivalent to a DFA is also a $PSPACE$-complete problem.*

8 Comments and Conclusion

Ideas developed in this paper come from a work done in the domain of Grammatical Inference. A main problem in this field is to infer efficiently (a representation of) a regular language from a finite set of examples of this language. Some positive results can be proved when regular languages are represented by Deterministic Finite Automata (DFA). For example, it has been proved that Regular Languages represented by DFA can be infered from *given data* ([Gol78,Hig97]). In this framework, classical inference algorithms such as RPNI ([OG92]) need a polynomial number of examples relatively to the size of the minimal DFA that recognizes the language to be infered. So, regular languages as simple as $\Sigma^* 0 \Sigma^n$ cannot be infered efficiently using these algorithms since their minimal DFA have an exponential number of states. Hence, it is a natural idea to try to use other kind of representations for regular languages, such as Non deterministic Finite Automata (NFA). Unfortunately, it has been proved that Regular Languages represented by NFA cannot be efficiently infered from given data ([Hig97]). We described in [DLT00a] an inference algorithm (*DeLeTe*) that computes the canonical RFSA of a target regular language from given data. Using this algorithm, languages such as $\Sigma^* 0 \Sigma^n$ become efficiently learnable. So, introducing the class of RFSA in the field of grammatical inference seems to be a promising idea. However, we have to deal with the fact that most decision and construction problems linked to the class of RFSA are untractable in the worst case. What are the practical consequences of these worst-case complexity results ? Experiments we are currently leading in the field of grammatical inference let us think that they could be not too dramatic.

While achieving this work, we have felt that RFSA was a class of automata worth being studied for itself, from a language theory point of view and this is what we have done in this paper. The class of RFSA has a very simple definition. It provides a description level of regular languages which is intermediate between a representation by deterministic automata and a representation that uses the whole class of non deterministic automata. RFSA shares two main properties with the class of DFA: the existence of a canonical minimal form and the fact that states correspond to natural component of the recognized language. Moreover canonical RFSA can be exponentially smaller than

the equivalent minimal DFA. All these properties show that the RFSA is an interesting class whose study must be carried on.

Acknowledgments

We would like to thank Michel Latteux for his helpful comments and advice.

References

ADN92. A. Arnold, A. Dicky and M. Nivat. A note about minimal non-deterministic automata. *Bulletin of the EATCS*, 47:166–169, June 1992.

Brz62. J. A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In *Mathematical Theory of Automata*, volume 12 of *MRI Symposia Series*, pages 52–561. 1962.

Car70. C. Carrez. On the minimalization of non-deterministic automaton. Technical report, Laboratoire de Calcul de la Faculté des Sciences de l'Université de Lille, 1970.

DLT00a. F. Denis, A. Lemay and A. Terlutte. Apprentissage de langages réguliers à l'aide d'automates non déterministes. In *CAP'2000*, 2000.

DLT00b. F. Denis, A. Lemay and A. Terlutte. Residual finite state automata. Technical Report LIFL 2000-08, L.I.F.L., 2000.

GJ79. Michael R. Garey and David S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completness*. W.H. Freeman and Co, San Francisco, 1979.

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.

HU79. J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

Kle56. S. C. Kleene. Representation of events in nerve nets and finite automata. In C. Shannon and J. McCarthy, editors, *Automata Studies, Annals of Math. Studies 34*. New Jersey, 1956.

Myh57. J. Myhill. Finite automata and the representation of events. Technical Report 57-624, WADC, 1957.

Ner58. A. Nerode. Linear automaton transformation. In *Proc. American Mathematical Society*, volume 9, pages 541–544, 1958.

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

Yu97. Sheng Yu. *Handbook of Formal Languages, Regular Languages*, volume 1, chapter 2, pages 41–110. 1997.