Residual Finite State Automata.

Fundamenta Informaticae XX (2002) 1–30
IOS Press

Residual Finite State Automata

François Denis*
LIF, UMR 6166 CNRS, Université de Provence, Marseille

Aurélien Lemay, Alain Terlutte[†]
GRAPPA-LIFL, Université de Lille I

Abstract. We define a new variety of Nondeterministic Finite Automata (NFA): a Residual Finite State Automaton (RFSA) is an NFA all the states of which define residual languages of the language L that it recognizes ; a residual language according to a word u is the set of words v such that uv is in L. We prove that every regular language is recognized by a unique (canonical) RFSA which has a minimal number of states and a maximal number of transitions. Canonical RFSAs are based on the notion of prime residual languages, i.e. that are not the union of other residual languages. We provide an algorithmic construction of the canonical RFSA from a given NFA. We study the size of canonical RFSAs and the complexity of our constructions.

1. Introduction

Regular languages are among the most studied objects in formal language theory. Deterministic finite automata (DFA) and nondeterministic finite automata (NFA) are two basic types of representation of regular languages. DFAs have many good properties: most classical constructions are polynomial and there exists a unique minimal element for a given regular language. Moreover, the Myhill-Nerode theorem shows that the states of a DFA correspond to natural components of the language it recognizes: its *residual languages* or *Brzozowski derivatives* or *left quotients* (here, we shall use the first name). NFAs are a generalization of DFAs which lose most properties of DFAs but gain concision. For example, the minimal DFA which recognizes the language $\Sigma^* a \Sigma^n$, with $\Sigma = \{a, b\}$, has 2^{n+1} states while a minimal equivalent NFA has only $n+2$ states. However, there may exist several non-isomorphic equivalent minimal NFAs and states of NFAs may correspond to no natural component of the language they recognize.

* fdenis@cmi.univ-mrs.fr
[†] {lemay, terlutte}@lifl.fr

Both type of representations properties, concision and the fact that they are based on natural components of the associated language, can be necessary within certain application field such as Grammatical Inference. The main goal of Grammatical Inference is to identify some target language from *examples* of this language, i.e. words together with a piece of information indicating whether it belongs to the language. General machine learning properties show that targets with short representations should be identified from fewer examples than others. On the other hand, inference algorithms try to detect properties of the target language from properties of some of its examples in order to build some representation of it. However, it has been shown that regular languages can be polynomially identified from given data using DFAs representations [7, 8] but that they cannot be identified in the same conditions using NFAs representations. In consequence, languages as simple as $\Sigma^* a \Sigma^n$ cannot be infered efficiently by inference algorithms using DFAs representations and NFAs representations cannot be used. Hence, it is a natural goal to look for intermediary representations having both kind of properties.

In this paper, we consider NFAs all the states of which define residual languages of the language it recognizes. We call Residual Finite State Automata (RFSA) such automata. RFSAs have been introduced in [5].Clearly, all DFAs are RFSAs but the converse is false. We show that we can naturally associate with every regular language L an RFSA which has a minimal number of states and which we call the canonical RFSA of L: each of its states is associated with a *prime* residual language, i.e. a residual language which is not the union of other residual languages. We provide an algorithmic construction of the canonical RFSA equivalent to a given NFA which stems from the classical subset construction used to build the minimal DFA. We give some results on the size of RFSAs: for example, there are canonical RFSAs exponentially larger (resp. smaller) than equivalent minimal NFAs (resp. DFAs). Then, we study RFSAs over a one-letter alphabet and we show that the gap between the sizes of minimal DFAs and canonical RFSAs is quadratic in the worst case: remind that the gap between the sizes of minimal DFAs and NFAs can be superpolynomial. Finally, we give some complexity results which show that natural constructions and decision problems are PSPACE-complete.

In Section 2, we recall classical definitions and notations about regular languages and automata. We define RFSAs in Section 3 and we study their properties in Section 4. In particular, we introduce the notion of canonical RFSA. The construction of the canonical RFSA using the subset method is given in Section 5. In Section 6, we study some particular (and pathological) RFSAs. RFSAs over one-letter alphabet are studied in Section 7. Section 8 is devoted to the study of the complexity of decision and construction problems on RFSAs. We conclude in Section 9.

2. Preliminaries

In this section, we recall some definitions on finite automata. For more information, we invite the reader to consult [9, 14].

2.1. Automata and languages

Let Σ be a finite alphabet, and let Σ^* be the set of words on Σ. We denote by ε the empty word and by $|u|$ the length of a word u. For an integer n, we define $\Sigma^n = \{u \in \Sigma^* \mid |u| = n\}$ and $\Sigma^{\leq n} = \{u \in \Sigma^* \mid |u| \leq n\}$. A *language* is a subset of Σ^*.

A *nondeterministic 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 final states, δ is the *transition function* of the automaton defined from a subset of $Q \times \Sigma$ to 2^Q. We also denote by δ the extended transition function defined from a subset of $2^Q \times \Sigma^*$ to 2^Q by

- $\delta(\{q\}, \varepsilon) = \{q\}$,
- $\delta(\{q\}, x) = \delta(q, x)$, where $x \in \Sigma$,
- $\delta(Q', u) = \bigcup_{q \in Q'} \delta(\{q\}, u)$, where $u \in \Sigma^*$,
- $\delta(\{q\}, ux) = \delta(\delta(q, u), x)$, where $x \in \Sigma$ and $u \in \Sigma^*$.

An NFA is *deterministic* (DFA) if Q_0 contains only one element q_0 and if $\forall q \in Q$, $\forall x \in \Sigma$, $Card(\delta(q, x)) \leq 1$. An NFA is *trimmed* if and only 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 an NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ if $\delta(Q_0, u) \cap F \neq \emptyset$ and the language recognized by A is $L_A = \{u \in \Sigma^* \mid \delta(Q_0, u) \cap F \neq \emptyset\}$. Let $Q' \subseteq Q$. We denote by $L_{A,Q'}$ the language $\{v \in \Sigma^* \mid \delta(Q', v) \cap F \neq \emptyset\}$. So, $L_A = L_{A,Q_0}$. When Q' contains exactly one state q, we simply denote $L_{A,Q'}$ by $L_{A,q}$. 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 [10] proves that the class of regular languages $Reg(\Sigma^*)$ is identical to $Rec(\Sigma^*)$.

The *reversal* u^R of a word u is defined inductively by $\varepsilon^R = \varepsilon$ and $(xv)^R = v^R x$ for $x \in \Sigma$ and $v \in \Sigma^*$. The *reversal* L^R of a language $L \subseteq \Sigma^*$ is defined by $L^R = \{u^R \in \Sigma^* \mid u \in L\}$. The reversal A^R of an NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ is defined by $A^R = \langle \Sigma, Q, F, Q_0, \delta^R \rangle$ with $q \in \delta^R(q', x)$ if and only if $q' \in \delta(q, x)$. We have $(L_A)^R = L_{A^R}$.

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 $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$ and we say that u is a *characterizing word* for $u^{-1}L$. The Myhill-Nerode theorem [11, 12] proves that the set of distinct residual languages of a language L is finite if and only if L is regular. Automata and residual languages are linked by the following properties.

Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. For any state $q \in Q$ and any word $u \in \Sigma^*$, if $q \in \delta(Q_0, u)$, then $L_{A,q} \subseteq u^{-1}L_A$.

Let $A = \langle \Sigma, Q, q_0, F, \delta \rangle$ be a trimmed DFA.

- For every non-empty residual language $u^{-1}L_A$, there exists a state $q \in Q$ such that $L_{A,q} = u^{-1}L_A$.
- For every state $q \in Q$, there exists a residual language $u^{-1}L_A$ such that $u^{-1}L_A = L_{A,q}$.

Furthermore, if A is the minimal DFA, the correspondence between states of A and non-empty residual languages of L_A is bijective.

3. Definition of Residual Finite State Automaton

Definition 3.1. A *Residual Finite State Automaton* (RFSA) is an NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ such that, for each state $q \in Q$, $L_{A,q}$ is a residual language of L_A. More formally, $\forall q \in Q$, $\exists u \in \Sigma^*$ such that $L_{A,q} = u^{-1}L_A$.

Trimmed DFAs are RFSAs. So, every regular language is recognized by a RFSA.

Example 3.1. Let $L = \Sigma^* a \Sigma$ where $\Sigma = \{a, b\}$. This language is recognized by the following automata A_1, A_2 and A_3 (Figures 1, 2, 3):

- A_1 is an NFA which is neither a DFA, nor an RFSA. Languages associated with states are: $L_{A_1,q_0} = \Sigma^* a \Sigma$, $L_{A_1,q_1} = \Sigma$, $L_{A_1,q_2} = \{\varepsilon\}$. As for every u in Σ^*, we have $uL \subseteq L$ and so, $L \subseteq u^{-1}L$, neither L_{A_1,q_1} nor L_{A_1,q_2} are residual languages.

- A_2 is the minimal DFA that recognizes L. A_2 is also an RFSA. We have $L_{A_2,q_0} = \Sigma^* a \Sigma = \varepsilon^{-1}L$, $L_{A_2,q_1} = \Sigma^* a \Sigma \cup \Sigma = a^{-1}L$, $L_{A_2,q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\} = (aa)^{-1}L$, $L_{A_2,q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\} = (ab)^{-1}L$.

- A_3 is an RFSA. Indeed, we have $L_{A_3,q_0} = \varepsilon^{-1}L$, $L_{A_3,q_1} = a^{-1}L$, $L_{A_3,q_2} = (ab)^{-1}L$. One can notice that A_3 is not a DFA.

Definition 3.2. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an RFSA, and let q be a state of A. The word u is a *characterizing word* for q if $L_{A,q} = u^{-1}L_A$. The automaton A is *consistent* if each state q is reachable by a characterizing word for q. We say that A is *strongly consistent* if each state q is reachable by every characterizing word for q.

Examples of not consistent RFSA and not strongly consistent RFSA are shown in Figure 4 and Figure 5.

If, for every state q of an NFA A, there exists a word u_q that only leads to q (that is, such that $\delta(Q_0, u_q) = \{q\}$), then A is an RFSA since $L_{A,q} = u_q^{-1}L_A$. However, next example shows that the converse property is false.

Example 3.2. Let $L = a^*b^* + b^*a^*$ and consider the automaton described in Figure 6. This automaton is a strongly consistent RFSA. But we can observe that there exists no word u such that $\delta(Q_0, u) = \{q_0\}$.

4. Properties of Residual Finite State Automata

4.1. General Properties

Definition 4.1. Let L be a regular language over Σ and let $u \in \Sigma^*$. The residual language $u^{-1}L$ is *prime* if it is not equal to the union of the residual languages it strictly contains: let $R_u = \{v \in \Sigma^* \mid v^{-1}L \subsetneq u^{-1}L\}$,

$$u^{-1}L \text{ is prime if } \bigcup_{v \in R_u} v^{-1}L \subsetneq u^{-1}L.$$

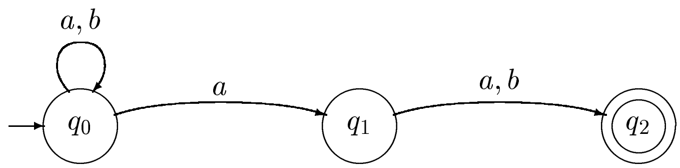

Figure 1. A_1 is an automaton recognizing $\Sigma^* a \Sigma$; it is neither a DFA nor an RFSA.

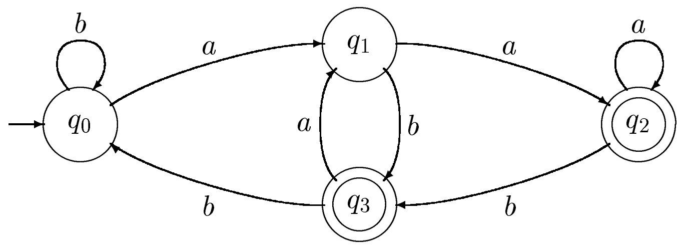

Figure 2. A_2 is the minimal DFA recognizing $\Sigma^* a \Sigma$.

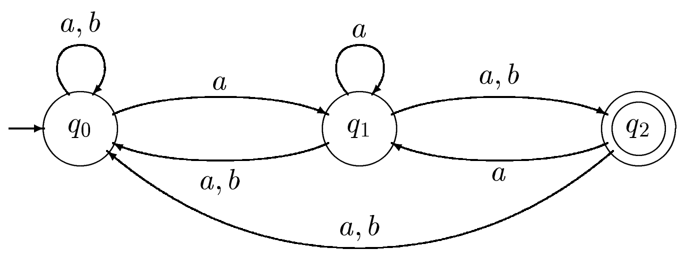

Figure 3. A_3 is the canonical RFSA recognizing $\Sigma^* a \Sigma$.

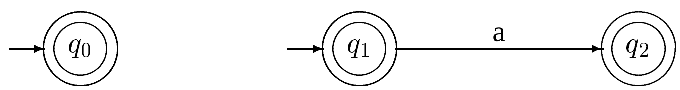

Figure 4. An RFSA which recognizes $L = \{\varepsilon, a\}$. It is not consistent as the only word which reaches q_0 is ε but $L_{q_0} \neq \varepsilon^{-1} L$.

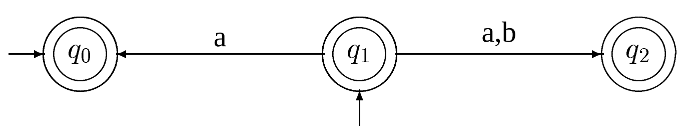

Figure 5. An RFSA which is consistent but not strongly consistent.

Figure 6. An RFSA recognizing the language $a^*b^* + b^*a^*$.

A residual language is *composite* if it is not prime. A state q of an RFSA A is *prime* (resp. *composite*) if the residual language $L_{A,q}$ is prime (resp. composite).

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

Proposition 4.1. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an RFSA. For each prime residual language $u^{-1}L_A$, there exists a state $q \in \delta(Q_0, u)$ such that $L_{A,q} = u^{-1}L_A$.

Proof:
As $u^{-1}L_A$ is prime, $\delta(Q_0, u)$ is not empty. Let $\delta(Q_0, u) = \{q_1, \ldots, q_s\}$ and let $v_1, \ldots, v_s$ be words such that $L_{A,q_i} = v_i{}^{-1}L_A$ for every $1 \le i \le s$. We have

$$u^{-1}L_A = \bigcup_{i=1}^{s} L_{A,q_i} = \bigcup_{i=1}^{s} v_i^{-1}L_A.$$

As $u^{-1}L_A$ is prime, there exists v_k such that $u^{-1}L_A = v_k^{-1}L_A = L_{A,q_k}$. □

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

4.2. Saturation operator

We define a *saturation* operator which may add transitions and initial states to an NFA without modifying the language it recognizes.

Definition 4.2. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. The *saturated* of A is the automaton $A^s = \langle \Sigma, Q, Q_0^s, F, \delta^s \rangle$ where $Q_0^s = \{q \in Q \mid L_{A,q} \subseteq L_A\}$ and $\delta^s(q, x) = \{q' \in Q \mid xL_{A,q'} \subseteq L_{A,q}\}$ for $q \in Q$ and $x \in \Sigma$. We say that an automaton A is saturated if $A = A^s$.

Lemma 4.1. Let A_1 and A_2 be two NFAs over Σ sharing the same set of states Q. If $L_{A_1} = L_{A_2}$ and if for every state $q \in Q$, $L_{A_1,q} = L_{A_2,q}$, then $A_1^s = A_2^s$.

Proof:
Let $A_1 = \langle \Sigma, Q, Q_{0,1}, F_1, \delta_1 \rangle$ and $A_2 = \langle \Sigma, Q, Q_{0,2}, F_2, \delta_2 \rangle$. The state q is initial in A_1^s iff $L_{A_1,q} \subseteq L_{A_1}$, i.e. iff q is initial in A_2^s. In the same way, $q' \in \delta_1^s(q, x)$ in A_1^s iff $xL_{A_1,q'} \subseteq L_{A_1,q}$, i.e. iff $q' \in \delta_2^s(q, x)$ in A_2^s. Eventually, the state q is terminal in A_1^s iff $\varepsilon \in L_{A_1,q}$, i.e. iff q is terminal in A_2^s. □

Proposition 4.2. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA and let $A^s = \langle \Sigma, Q, Q_0^s, F, \delta^s \rangle$ be the saturated of A. For each q in Q, we have $L_{A,q} = L_{A^s,q}$.

Proof:
Clearly, $L_{A,q} \subseteq L_{A^s,q}$ as the saturated of an automaton is obtained by adding transitions and initial states.

The converse inclusion can be proved by induction on the length of the words of $L_{A^s,q}$. □

We can then deduce the following corollaries.

Corollary 4.1. Let A be an NFA and A^s be its saturated. Then A and A^s recognize the same language and $A^s = (A^s)^s$.

Corollary 4.2. If A is an RFSA, then A^s is also an RFSA.

It could have seemed better to define the saturation operator another way: in order to saturate an NFA, add initial states and transitions as long as this operation does not modify the language recognized by the automaton. Unfortunately, this procedure does not lead to a unique NFA in the general case: in the automaton defined in Figure 4, it is possible to add a transition labelled by a from q_1 to q_0 and from q_0 to q_2 without modifying the language but not simultaneously. However, we have the following lemma:

Lemma 4.2. Let A be a consistent RFSA, let q, q' be two states of A and let x be a letter such that adding the transition (q, x, q') to A does not modify the language recognized by A. Then, $xL_{A,q'} \subseteq L_{A,q}$.

Proof:
Let $v \in L_{A,q'}$ and let u such that $L_{A,q} = u^{-1}L_A$ and q is reachable by u. As adding the transition (q, x, q') to A does not modify the language recognized by A, the word $uxv \in L_A$. Then, $xv \in u^{-1}L_A = L_{A,q}$. Therefore, $xL_{A,q'} \subseteq L_{A,q}$. □

Corollary 4.3. In order to compute the saturated of a consistent RFSA A, the following procedure can be applied: add initial states and transitions as long as the recognized language is not modified.

Proof:
A state q of A can become an initial state without changing the recognized language if and only if $L_{A,q} \subseteq L_A$. The result directly comes from this remark and previous lemma. □

There are saturated RFSAs which are not consistent (see Figure 7). Moreover, there are consistent saturated RFSAs which are not strongly consistent (see Figure 8). However, we have the following result.

Proposition 4.3. If A is a saturated RFSA and if q is a prime state of A, then for every word u such that $L_{A,q} = u^{-1}L_A$, q is reachable by u.

Proof:

If $u = \varepsilon$ and $L_{A,q} = u^{-1}L_A$, we have$L_{A,q} \subseteq L_A$ and as A is saturated, q is initial.

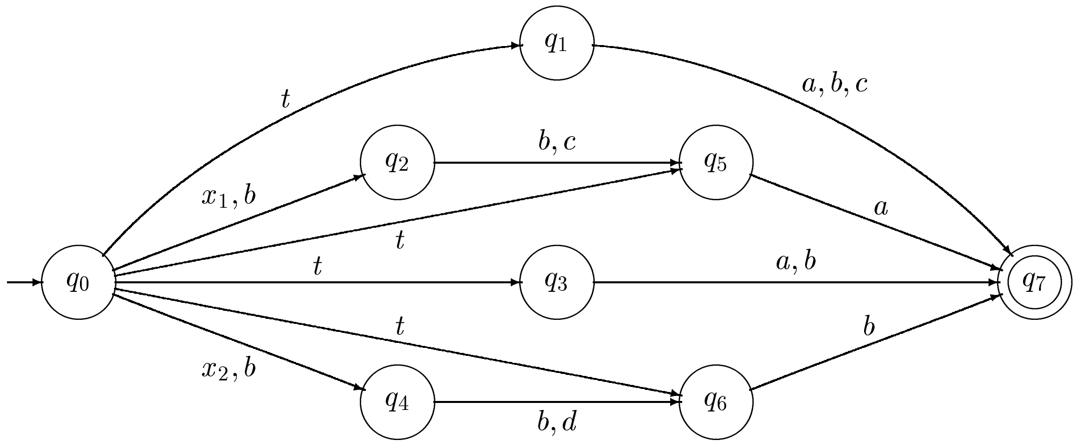

Figure 7. A saturated inconsistent RFSA: the state q_3 defines the residual language of bb but it is not reachable by bb.

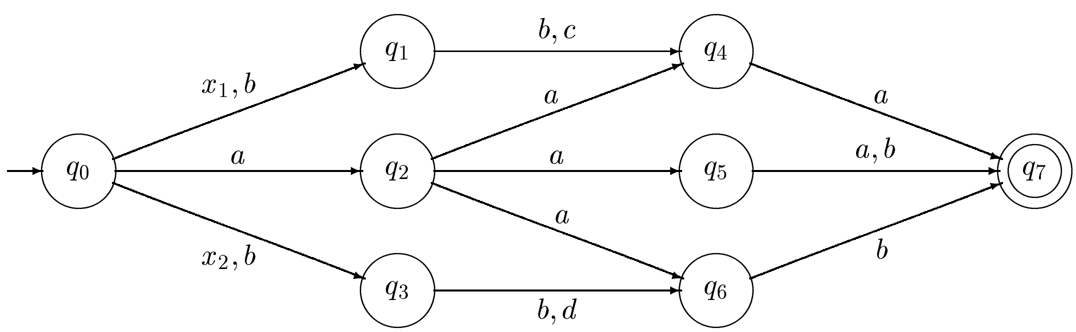

Figure 8. A saturated consistent RFSA which is not strongly consistent; aa and bb define the same residual language $\{a, b\}$ but the state q_5 is not reachable by bb.

Suppose now that $L_{A,q} = (ux)^{-1}L_A$, where $x \in \Sigma$. As $L_{A,q}$ is a prime residual language, by Proposition 4.1, there exists $q'' \in \delta(Q_0, ux)$ such that $L_{A,q''} = (ux)^{-1}L_A$. Let q' be such that $q' \in \delta(Q_0, u)$ and $q'' \in \delta(q', x)$.

Let $v \in L_{A,q}$. As $L_{A,q} = L_{A,q''}$, $xv \in L_{A,q'}$. That is, $xL_{A,q} \subseteq L_{A,q'}$ and as A is saturated, $q \in \delta(q', x)$. Therefore, q is reachable by ux in A. □

4.3. Reduction operator ϕ

We define a *reduction* operator ϕ which may delete states in an NFA without changing the language it recognizes.

Definition 4.3. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA, and let q be a state of Q. We denote by $R(q)$ the set $\{q' \in Q \setminus \{q\} \mid L_{A,q'} \subseteq L_{A,q}\}$. We say that q is *erasable* in A if $L_{A,q} = \bigcup_{q' \in R(q)} L_{A,q'}$.

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

Note that if A is a saturated NFA and if q is an erasable state in A, then $\phi(A, q)$ is obtained by deleting q and its associated transitions from A ; no transitions are added.

Definition 4.4. Let A be an NFA. If there is no erasable state in A, we say that A is *reduced*.

Proposition 4.4. Let A be an NFA, q a state of A and $A' = \phi(A, q)$. For every state $q' \in Q \setminus \{q\}$, we have $L_{A,q'} = L_{A',q'}$. As a consequence, $L_A = L_{A'}$.

Proof:
If q is not an erasable state, the proposition is straightforward. Suppose now that q is an erasable state. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ and $\phi(A, q) = A' = \langle \Sigma, Q', Q'_0, F', \delta' \rangle$.

We first prove by induction on n that, for every state $\bar{q} \neq q$ and every integer n,

$$L_{A,\bar{q}} \cap \Sigma^{\leq n} \subseteq L_{A',\bar{q}} \cap \Sigma^{\leq n}.$$

For $n = 0$, if $\varepsilon \in L_{A,\bar{q}}$, then $\bar{q} \in F$ and as $\bar{q} \neq q$, $\bar{q} \in F'$ and $\varepsilon \in L_{A',\bar{q}}$.

Let $u = x\bar{u} \in L_{A,\bar{q}}$ where $u \in \Sigma^{\leq n}$, $x \in \Sigma$ and let $\bar{q}_1 \in Q$ such that $\bar{q}_1 \in \delta(\bar{q}, x)$ and $\delta(\{\bar{q}_1\}, \bar{u}) \cap F \neq \emptyset$.

- If $\bar{q}_1 \in Q'$, we can apply the inductive hypothesis: $\bar{u} \in L_{A',\bar{q}_1}$ and as $\bar{q}_1 \in \delta'(\bar{q}, x)$, we have $u \in L_{A',\bar{q}}$.

- Otherwise, $\bar{q}_1 = q$ and there exists a state $\bar{\bar{q}}_1 \neq q$ such that $L_{A,\bar{\bar{q}}_1} \subseteq L_{A,q}$ and $\bar{u} \in L_{A,\bar{\bar{q}}_1}$. Using the inductive hypothesis again, we have $\bar{u} \in L_{A',\bar{\bar{q}}_1}$ and as $\bar{\bar{q}}_1 \in \delta'(\bar{q}, x)$, we have $u \in L_{A',\bar{q}}$.

So, we have shown that for every state $\bar{q} \neq q$, $L_{A,\bar{q}} \subseteq L_{A',\bar{q}}$.

We prove now by induction that for every state $\bar{q} \neq q$ and every integer n,

$$L_{A',\bar{q}} \cap \Sigma^{\leq n} \subseteq L_{A,\bar{q}} \cap \Sigma^{\leq n}$$

For $n = 0$, if $\varepsilon \in L_{A',\bar{q}}$, then $\bar{q} \in F' \subseteq F$ and then $\varepsilon \in L_{A,\bar{q}}$.

Let $u = x\bar{u} \in L_{A',\bar{q}}$ where $u \in \Sigma^{\leq n}$, $x \in \Sigma$ and let $\bar{q}_1 \in Q'$ such that $\bar{q}_1 \in \delta'(\bar{q}, x)$ and $\delta'(\{\bar{q}_1\}, \bar{u}) \cap F' \neq \emptyset$. We can apply the inductive hypothesis: $\bar{u} \in L_{A,\bar{q}_1}$. If $\bar{q}_1 \in \delta(\bar{q}, x)$, we directly have $u \in L_{A,\bar{q}}$. Otherwise, we have $L_{A,\bar{q}_1} \subseteq L_{A,q}$ and $q \in \delta(\bar{q}, x)$. We also have $u \in L_{A,\bar{q}}$.

So, we have shown that for every state $\bar{q} \neq q$, $L_{A',\bar{q}} = L_{A,\bar{q}}$.

We show that A and A' recognize the same language by studying the two following cases.

- If $q \notin Q_0$, we have $L_A = \bigcup_{q_0 \in Q_0} L_{A,q_0} = \bigcup_{q_0 \in Q_0} L_{A',q_0} = \bigcup_{q_0 \in Q'_0} L_{A',q_0} = L_{A'}$.

- If $q \in Q_0$, we have

$$\begin{aligned}
L_A &= \bigcup_{q_0 \in Q_0} L_{A,q_0} \\
&= (\bigcup_{q_0 \in Q_0, q_0 \neq q} L_{A,q_0}) \cup L_{A,q} \\
&= (\bigcup_{q_0 \in Q_0, q_0 \neq q} L_{A,q_0}) \cup (\bigcup_{\bar{q} \in R(q)} L_{A,\bar{q}}) \\
&= (\bigcup_{q_0 \in Q_0, q_0 \neq q} L_{A',q_0}) \cup (\bigcup_{\bar{q} \in R(q)} L_{A',\bar{q}}) \\
&= \bigcup_{q_0 \in Q'_0} L_{A',q_0} \\
&= L_{A'}.
\end{aligned}$$

The reduction operator ϕ does not change the language recognized by the automaton. □

Corollary 4.4. The reduction operator ϕ is an internal operator in the class of RFSAs.

We shall now show that saturation and reduction operators commute.

Lemma 4.3. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA and let q be a state of Q. Then the automaton $\phi(A^s, q)$ is saturated.

Proof:
Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle, A^s = \langle \Sigma, Q, Q_0^s, F, \delta^s \rangle, \phi(A^s, q) = \langle \Sigma, Q', Q'_0, F', \delta' \rangle$ and let L be the language recognized by these three automata.

Let $q_0 \in Q' = Q \setminus \{q\}$. If $L_{\phi(A^s,q),q_0} \subseteq L_{\phi(A^s,q)}$ then $L_{A^s,q_0} \subseteq L_{A^s}$ from Proposition 4.4. Since A^s is saturated, we have $q_0 \in Q_0^s$ and as $Q'_0 = Q_0^s \setminus \{q\}$, we have $q_0 \in Q'_0$.

Let $x \in \Sigma$ and $q', q'' \in Q'$ be such that $xL_{\phi(A^s,q),q''} \subseteq L_{\phi(A^s,q),q'}$. Then, $xL_{A^s,q''} \subseteq L_{A^s,q'}$ from Proposition 4.4. Since A^s is saturated, we have $q'' \in \delta^s(q', x)$ and since $\delta^s(q', x) \setminus \{q\} \subseteq \delta'(q', x)$, we have $q'' \in \delta'(q', x)$.

Thenrefore $\phi(A^s, q)$ is saturated.

□

Proposition 4.5. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA recognizing a regular language L and q a state of Q. We have

$$[\phi(A, q)]^s = \phi(A^s, q)$$

Proof:

We can observe that $\phi(A, q)$ and $\phi(A^s, q)$ have the same set of states. Furthermore, languages associated with every state q' are identical in the two automata because of Propositions 4.2 and 4.4. Because of Lemma 4.1, the saturated of these automata are isomorphic, therefore $[\phi(A, q)]^s = [\phi(A^s, q)]^s$. But as $\phi(A^s, q)$ is a saturated automaton by Lemma 4.3, the proposition is proved. □

4.4. Canonical RFSA

Definition 4.5. Let L be a regular language. We define the canonical RFSA A of L the following way: $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ where

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

Example 4.1. An example of canonical RFSA is shown in Figure 3.

This definition assumes that the canonical RFSA is an RFSA ; we shall prove this presumption below.

We have showed that the reduction operator ϕ transforms an RFSA into an RFSA, and that it commutes with the saturation operator. We shall now show that, if A is a saturated RFSA, the reduction operator converges and that the resulting automaton is the canonical RFSA of L_A.

Proposition 4.6. Let L be a regular language. If $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ is a reduced saturated RFSA recognizing L, then A is (isomorphic to) the canonical RFSA of L.

Proof:
As A is an RFSA, every prime residual language $u^{-1}L$ of L can be defined as a language $L_{A,q}$ associated with some state $q \in Q$ (Proposition 4.1). As there are no erasable states in A, for every state q, $L_{A,q}$ is a prime residual language and distinct states define distinct languages. As A is saturated, prime residual languages contained in L correspond to initial states of Q_0. As A is saturated, for a prime residual language $u^{-1}L$ and a letter $x \in \Sigma$, we have $\delta(u^{-1}L, x) = \{v^{-1}L \in Q \mid x(v^{-1}L) \subseteq u^{-1}L\} = \{v^{-1}L \in Q \mid v^{-1}L \subseteq (ux)^{-1}L\}$ which is the transition function of the canonical RFSA. □

Let $A_0, \ldots, A_n$ be a sequence of NFAs such that for every index $1 \le i \le n$, there exists a state q_i of A_{i-1} such that $A_i = \phi(A_{i-1}, q_i)$. Propositions 4.5 and 4.6 show 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.

Theorem 4.1. The canonical RFSA of a regular language L is a strongly consistent RFSA which recognizes L and is minimal regarding to the number of states. Moreover, the canonical RFSA possesses a maximal number of transitions.

Proof:
It is an RFSA that recognizes L since it can be obtained from any RFSA that recognizes L using saturation and reduction operators, and since these two operators do not change either the language recognized by the automaton or the fact that the automaton is an RFSA. It possesses a minimal number of states because of Proposition 4.1 and it is strongly consistent from Proposition 4.3. It has a maximal number of transitions from Corollary 4.3. □

Several transitions of the canonical RFSA may be redundant as Example 4.3 shows. Unfortunately, there may exist several non isomorphic RFSAs having as many states as the canonical RFSA and a minimal number of transitions, i.e. such that no transition is redundant. However, it is possible to describe a procedure that rules out some redundant transitions in a systematic way.

Definition 4.6. Let L be a regular language. For any set of distinct residual languages R of L, define $max(R) = \{L' \in R \mid \forall L'' \in R, L' \subseteq L'' \Rightarrow L' = L''\}$. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be the canonical RFSA which recognizes L. The *simplified canonical RFSA* of L is the automaton $A' = \langle \Sigma, Q, Q_0', F, \delta' \rangle$ where $Q_0' = max(Q_0)$ and $\delta'(q, x) = max(\delta(q, x))$, for $q \in Q$ and $x \in \Sigma$.

It can be shown that the simplified canonical RFSA of L is an RFSA which recognizes L. Clearly, every regular language admits a unique simplified canonical RFSA.

Example 4.2. Let $\Sigma = \{a, b\}$. The simplified canonical RFSA for $\Sigma^* a \Sigma$ has 4 transitions less than the canonical RFSA and is the minimal RFSA with respect to number of states and transitions.

However, it may happen that several non isomorphic RFSAs have as many states as the simplified canonical RFSA but less transitions.

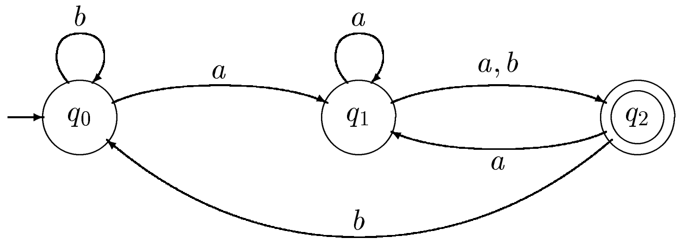

Figure 9. The simplified canonical RFSA for $\Sigma^* a \Sigma$.

Example 4.3. Consider the language $L = \{aa, ab, ba, bc, cb, cc, da, ea, eb, ec\}$. With regard to the number of states, the minimal RFSAs recognizing L have 6 states. The canonical RFSA has 17 transitions (see Figure 10). The simplified canonical RFSA has 14 transitions as $a^{-1}L = d^{-1}L$ and $b^{-1}L = d^{-1}L$. There are three non-isomorphic RFSAs with 13 transitions as $e^{-1}L = a^{-1}L \cup b^{-1}L = a^{-1}L \cup c^{-1}L = b^{-1}L \cup c^{-1}L$ and none with less transitions.

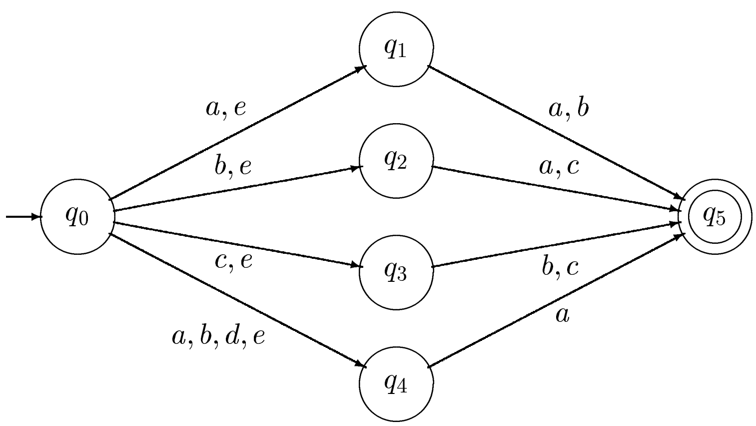

Figure 10. A canonical RFSA with 17 transitions. The simplified canonical RFSA has 14 transitions. There are three equivalent non-isomorphic RFSAs with 6 states and 13 transitions.

5. Construction of the canonical RFSA using the subset method

In the previous section, we described a way to build the canonical RFSA from a given DFA using saturation and reduction operators. Starting from an NFA, this method requires to build an equivalent DFA and to check whether residual languages are composite. These checks can be very expensive, even for simple automata. In this section, we present another method which stems from a classical construction of the minimal DFA of a language and which is easier to implement.

The subset construction is a classical method used to build a DFA equivalent to a given NFA. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. The method consists in building the set of reachable sets of states of A.

We denote by $Q_{R(A)}$ the set $\{p \in 2^Q \mid \exists u \in \Sigma^* \text{ s.t. } \delta(Q_0, u) = p\}$ and we define the subset automaton $D(A) = \langle \Sigma, Q_D, Q_{D0}, F_D, \delta_D \rangle$ with

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

The language associated with a state p in the automaton $D(A)$ is the union of the languages associated with the states that compose p in the automaton A, i.e. $L_{D(A),p} = \bigcup_{q \in p} L_{A,q}$.

The automaton $D(A)$ is a deterministic trimmed automaton that recognizes the same language as A.

We remind that the reversal of a language L (resp. of an NFA A) is denoted by L^R (resp. A^R). The following result provides a method to build the minimal DFA of L.

Theorem 5.1. [1] Let L be a regular language and B an NFA such that B^R is a DFA that recognizes L^R. Then $D(B)$ is the minimal DFA recognizing L.

We can deduce from this theorem that for an NFA A, $D(D(A^R)^R)$ is the minimal DFA recognizing the language L_A.

We adapt the subset construction technique to deal with inclusions of sets of states. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. We say that $p \in Q_{R(A)}$ is *coverable* if there exist $p_1, \ldots, p_l \in Q_{R(A)} \setminus \{p\}$, such that $p = \bigcup_{i=1}^{l} p_i$. We define the automaton $C(A) = \langle \Sigma, Q_C, Q_{C0}, F_C, \delta_C \rangle$ with

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

Lemma 5.1. Let A be an NFA. The automaton $C(A)$ is an RFSA recognizing L_A whose all states are reachable.

Proof:

The automaton $C(A)$ can be obtained from the DFA $D(A)$ in three steps, by using operations like saturation and reduction defined in Section 4. The states of $D(A)$ are associated with residual languages of L. We only have to verify that the transformations do not change these residual languages.

Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. Let $D(A) = \langle \Sigma, Q_D, Q_{D0}, F_D, \delta_D \rangle$.

- Build $A_1 = \langle \Sigma, Q_D, Q_{DC0}, F_D, \delta_D \rangle$ where $Q_{DC0} = \{p \in Q_D \mid p \subseteq Q_0\}$. The new initial states p verify $L_{D(A),p} \subseteq L_{D(A),Q_{D0}} \subseteq L$. This does not change the language.

- Build $A_2 = \langle \Sigma, Q_D, Q_{DC0}, F_D, \delta_{A_2} \rangle$ where $\delta_{A_2}(p, x) = \{p' \in Q_D \mid p' \subseteq \delta(p, x)\}$ for $p \in Q_D$ and $x \in \Sigma$.

  We have $L_{D(A),p'} = L_{A,p'} \subseteq L_{A,\delta(p,x)} = L_{D(A),\delta_D(p,x)}$ and then $xL_{D(A),p'} \subseteq L_{D(A),p}$. Thus, as in Proposition 4.2, the languages associated with states are not changed: $L_{A_2,p} = L_{D(A),p}$ for $p \in Q_D$.

- Build $C(A) = \langle \Sigma, Q_C, Q_{C0}, F_C, \delta_C \rangle$ by removing coverable states from A_2.

  Let p'' be a coverable state and (p, x, p'') be a transition leading to p'' in A_2. Let $p''_1, \ldots, p''_l$ be such that $p'' = \bigcup_{i=1}^{l} p''_i$. By the previous step, $p''_i \in \delta_{A_2}(p, x)$ for every $1 \leq i \leq l$.

  We also have $xL_{A_2,p''} = x(\bigcup_{i=1}^{l} L_{A_2,p''_i})$. So we can remove the transition (p, x, p'') from A_2 without changing the language $L_{A_2,p}$.

  When all transitions leading to p'' are removed, we can remove p'' itself ; if p'' is an initial state, each p''_i belongs to Q_{DC0} and the language L_{A_2} is not changed. When all coverable states are removed, we obtain the automaton $C(A)$.

□

Theorem 5.2. Let L be a regular language and let B be an NFA such that B^R is an RFSA recognizing L^R whose all states are reachable. Then $C(B)$ is the canonical RFSA recognizing L.

In order to prove this theorem, we introduce some lemmas.

Lemma 5.2. Let $B = \langle \Sigma, Q_B, Q_0, F, \delta \rangle$ be an NFA such that B^R is a trimmed RFSA. Let $q \in Q_B$ and $v \in \Sigma^*$ such that $L_{B^R,q} = (v^R)^{-1} L_B^R$, let $p \in Q_{R(B)}$. Then $v \in L_{B,p}$ if and only if $q \in p$.

Proof:
Let u be such that $p = \delta(Q_0, u)$; thus $L_{B,p} = u^{-1} L_B$.

We have

$$\begin{aligned} q \in p &\Leftrightarrow q \in \delta(Q_0, u) \\ &\Leftrightarrow u^R \in L_{B^R,q} = (v^R)^{-1} L_B^R \\ &\Leftrightarrow v^R u^R \in L_B^R \\ &\Leftrightarrow uv \in L_B \\ &\Leftrightarrow v \in u^{-1} L_B = L_{B,p}. \end{aligned}$$

□

Lemma 5.3. Let $B = \langle \Sigma, Q_B, Q_0, F, \delta \rangle$ be an NFA such that B^R is a trimmed RFSA. For every p, $p' \in Q_{R(B)}$, we have $L_{B,p} \subseteq L_{B,p'}$ if and only if $p \subseteq p'$.

Proof:
If $p \subseteq p'$, then $L_{B,p} = \bigcup_{q \in p} L_{B,q} \subseteq \bigcup_{q \in p'} L_{B,q} = L_{B,p'}$.

Conversely, let $q \in p$. As B^R is an RFSA, there exists a word v such that $L_{B^R,q} = (v^R)^{-1} L_B^R$. From Lemma 5.2, we know that v is in $L_{B,p}$. As $L_{B,p} \subseteq L_{B,p'}$, v also belongs to $L_{B,p'}$ and, using Lemma 5.2, we have $q \in p'$. □

Lemma 5.4. Let B be an NFA such that B^R is a trimmed RFSA. For every p, p_1, $p_2 \ldots p_n \in Q_{R(B)}$, $L_{B,p} = \bigcup_{1 \leq k \leq n} L_{B,p_k}$ is equivalent to $p = \bigcup_{1 \leq k \leq n} p_k$.

Proof:
It is obvious that $p = \bigcup_{1 \le k \le n} p_k$ implies $L_{B,p} = \bigcup_{1 \le k \le n} L_{B,p_k}$.

Suppose that $L_{B,p} = \bigcup_{1 \le k \le n} L_{B,p_k}$. Then $L_{B,p_k} \subseteq L_{B,p}$ for every $1 \le k \le n$. Using Lemma 5.4, we have, $p_k \subseteq p$ for all k and so, $\bigcup_{1 \le k \le n} p_k \subseteq p$.

Let $q \in p$. Since B^R is a trimmed RFSA, there exists a word v such that $L_{B^R,q} = (v^R)^{-1} L_B^R$. From Lemma 5.2, we have $v \in L_{B,p}$. As $L_{B,p} = \bigcup_{1 \le k \le n} L_{B,p_k}$, there exists an index k such that $v \in L_{B,p_k}$. From Lemma 5.2 again, we have $q \in p_k$. So $p \subseteq \bigcup_{1 \le k \le n} p_k$. □

Proof:
[Proof of Theorem 5.2]

- Reachable sets of states of B correspond to residual languages of L and Lemma 5.4 shows that composite residual languages correspond to coverable sets of states. So, $p \in Q_C$ if and only if $L_{B,p}$ is a prime residual language of L and Q_C can naturally be identified with the set of states of the canonical RFSA. Due to Lemma 5.3, we also verify that $L_{B,p} = u^{-1}L$ implies $p = \delta(Q_0, u)$; the inverse is obvious.

- $Q_{C0} = \{p \in Q_C \mid p \subseteq Q_0\}$. Lemma 5.3 tells us that $p \subseteq Q_0$ is equivalent to $L_{B,p} \subseteq L_{B,Q_0} = L$. So $Q_{C0} = \{p \in Q_C \mid L_{B,p} \subseteq L\}$ corresponds to the set of initial states of the canonical RFSA.

- $F_C = \{p \in Q_C \mid p \cap F \neq \emptyset\}$. We have $\varepsilon \in L_{B,p}$ if and only if there exists $q_i \in p \cap F$. So $F_C = \{p \in Q_C \mid \varepsilon \in L_{B,p}\}$ is the set of final states of the canonical RFSA.

- For each state $p \in Q_C$, let $u_p \in \Sigma^*$ be such that $\delta(Q_0, u_p) = p$. Clearly, $u_p^{-1}L = L_{B,p}$.

  For any $p, p' \in Q_C$ and $x \in \Sigma$,

  $$\delta(p, x) = \delta(Q_0, u_p x) \in R(B) \text{ and } p' \in \delta_C(p, x) \text{ iff } p' \subseteq \delta(p, x).$$

  As $L_{B,p'} = u_{p'}^{-1}L$ and $L_{B,\delta(p,x)} = (u_p x)^{-1}L$, we can deduce from Lemma 5.4 that

  $$p' \subseteq \delta(p, x) \text{ iff } u_{p'}^{-1}L \subseteq (u_p x)^{-1}L.$$

  So, the transition functions are equivalent.

□

We can deduce from Lemma 5.1 and Theorem 5.2 that for any NFA A, $C(C(A^R)^R)$ is the canonical RFSA of L_A.

The simplified canonical RFSA can be computed from the canonical RFSA by using the operator *max*. However, there exists a similar construction which allows to obtain it directly from any NFA. Let $C'(A) = \langle \Sigma, Q_C, Q_{C'0}, F_C, \delta_{C'} \rangle$ with

$$\begin{aligned}
Q_C &= \{p \in Q_{R(A)} \mid p \text{ is not coverable }\} \\
Q_{C'0} &= \{p \in Q_C \mid p \subseteq Q_0 \text{ and } \nexists p' \in Q_C \text{ s.t. } p \subsetneq p' \subseteq Q_0\} \\
F_C &= \{p \in Q_C \mid p \cap F \neq \emptyset\} \\
\delta_C(p, x) &= \{p' \in Q_C \mid p' \subseteq \delta(p, x) \text{ and } \nexists p'' \in Q_C \text{ s.t. } p' \subsetneq p'' \subseteq \delta(p, x)\}, \text{for } p \in Q_C \text{ and } x \in \Sigma.
\end{aligned}$$

The simplified canonical RFSA is obtained by $C'(C'(A^R)^R)$.

Example 5.1. Figures 11, 12, 13 and 14, shows the construction of the canonical RFSA recognizing the language $\Sigma^* a \Sigma^2$ with $\Sigma = \{a, b\}$ by using the subset construction.

The automaton A recognizes $\Sigma^* a \Sigma^2$ and the reversal automaton A^R is deterministic.

The first steps of the construction of $C(A)$ are represented on Figure 12; they are identical to steps in the classical subset construction. The state 012 (resp. 013) is coverable with 01 and 02 (resp. 01 and 03). Since the states 012 and 013 are coverable, it is not necessary to build the next states.

On the Figure 13, the coverable states 012 and 013 have been removed. Transitions reaching the state 012 (resp. 013) have been redirected to the states 01 and 02 (resp. 01 and 03). We obtain an RFSA which recognizes the language $\Sigma^* a \Sigma^2$ and which is by accident the simplified canonical RFSA too.

Finally, the canonical RFSA $C(A)$ is obtained by saturation. The state 0 being included in each other state, every transition which reaches a state, has also to reach the state 0.

Note that, as in the deterministic case, this construction may produce cumbersome intermediate automata; indeed, it is possible to find examples for which $C(A^R)$ has an exponential number of states with regard to the number of states of A or $C(C(A^R)^R)$. Thus in the worst case, this algorithm is exponential with regard to the size of the canonical RFSA. This situation can be observed with the reversal of the automaton used in Proposition 6.2.

6. Results on size of RFSAs

We classically take the number of states of an automaton as a measure of its size. It can be argued that the number of states of an automaton is suitable for DFAs but not for NFAs since the number of transitions in the latter can be quadratic with regard to the number of states. However, our results show the existence of exponential or superpolynomial gaps between the number of states of particular NFAs, RFSAs and DFAs. These results imply similar gaps between number of transitions.

The size of a canonical RFSA is bounded by the size of the equivalent minimal DFA and by the size of one of its equivalent minimal NFAs. We show that both bounds can be reached despite the fact that there is an exponential gap between them.

Proposition 6.1. 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 the size of a minimal NFAs.

Proof:
Consider the languages $L_n = \Sigma^* a \Sigma^n$, where n is an integer and $\Sigma = \{a, b\}$.

It is well known that minimal NFAs for L_n have $n+2$ states and that L_n has 2^{n+1} distinct residual languages. It is easy to verify that only $n+2$ of them are prime: $\varepsilon^{-1} L_n$ and $(ab^i)^{-1} L_n$ for $0 \leq i \leq n$.

□

Proposition 6.2. The size of the canonical RFSA of a language L can be exponentially larger than the size of a smallest NFA recognizing L.

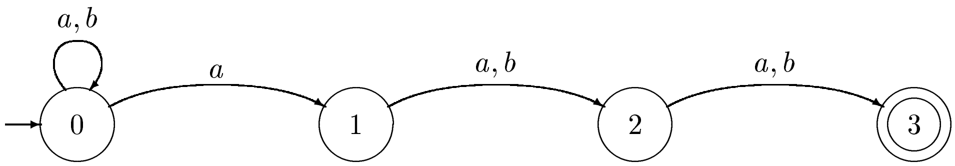

Figure 11. A is an NFA recognizing $\Sigma^* a \Sigma^2$; A^R is deterministic.

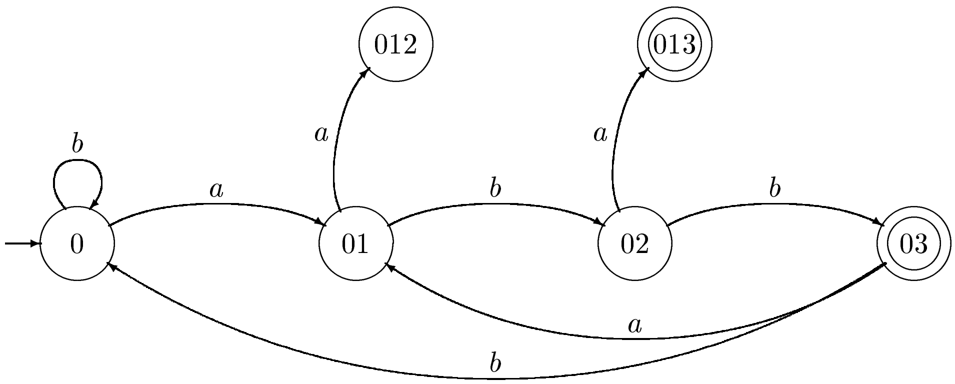

Figure 12. First steps of the construction of $C(A)$.

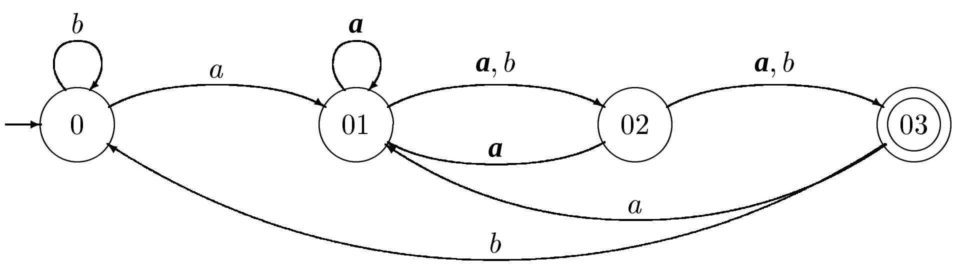

Figure 13. The coverable states have been removed and transitions are redirected; this automaton is an RFSA.

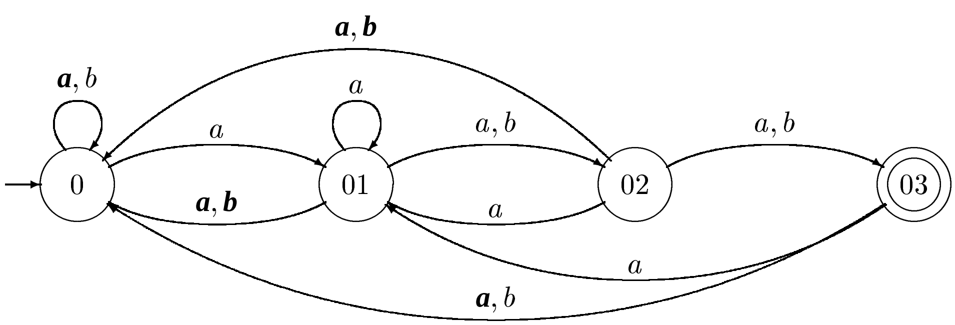

Figure 14. The canonical RFSA $C(A)$ recognizing $\Sigma^* a \Sigma^2$ is obtained by saturation.

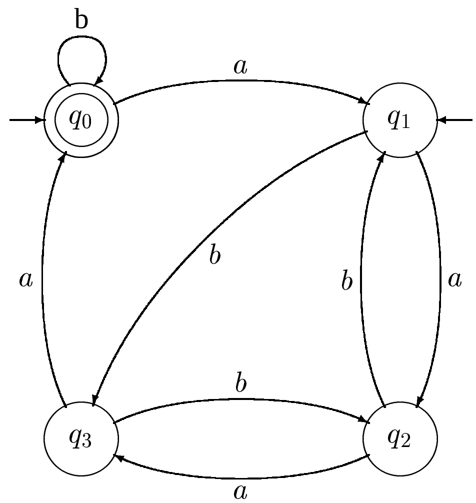

Figure 15. Automaton A_4 whose canonical RFSA is exponentially larger than A_4.

Proof:
We can verify this proposition on automata $A_n = \langle \Sigma, Q, Q_0, F, \delta \rangle$ defined by

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

The automaton A_4 is represented in Figure 15.

The reversal of A_n is trimmed and deterministic, thus we can apply Theorem 5.2. The automaton $C(A_n)$ is the canonical RFSA.

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

Therefore, the canonical RFSA $C(A_n)$ has a size exponentially larger than the size of the initial NFA. $\square$

Every non-empty residual language of a regular language L has a minimal characteristic word whose length is bounded by the number of states of the minimal DFA. Next proposition shows that this is no longer true if we consider RFSA.

Proposition 6.3. There exist regular languages for which the smallest characterizing word for some residual language is longer than any polynomial in the number of states of the canonical RFSA.

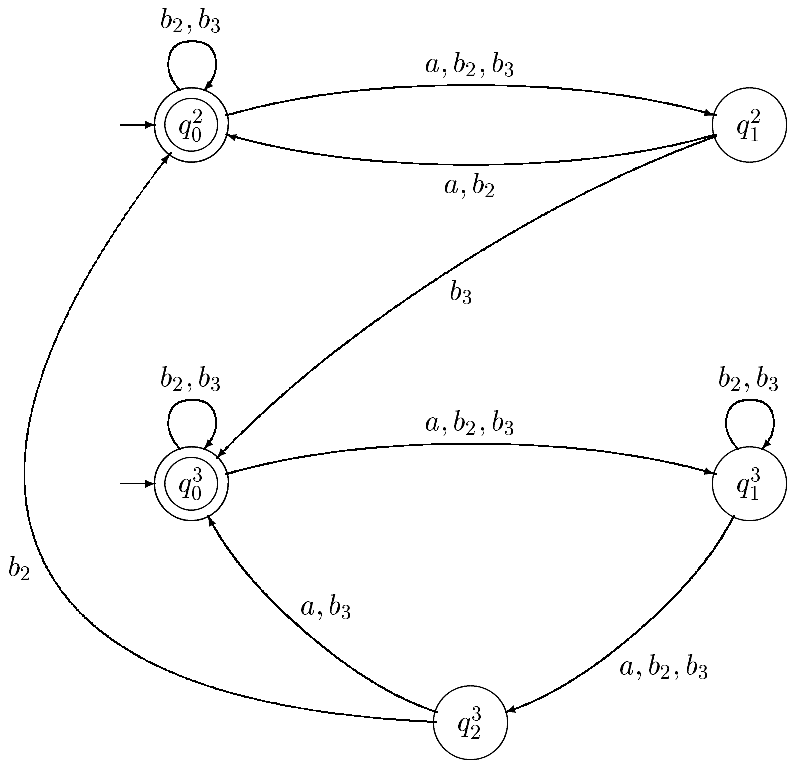

Figure 16. Automaton A_P for $P = \{2, 3\}$.

Proof:

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

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

and

$$\begin{array}{lll} \delta(q_j^p, a) & = \{q^p_{(j+1) mod\ p}\} & \text{for } 0 \leq j < p,\ p \in P \\ \delta(q_j^p, b_{p'}) & = \{q_j^p, q_{j+1}^p\} & \text{for } 0 \leq j < p-1,\ p, p' \in P \\ \delta(q_{p-1}^p, b_{p'}) & = \{q_0^{p'}\} & \text{for } p, p' \in P \end{array}$$

See Figure 16 for $P = \{2, 3\}$.

Let $N = p_1 \times \ldots \times p_n$ and let $u_{ij} = a^{N-1} b_i a^j$ for $1 \leq i \leq n$ and $0 \leq j < p_i$. We can check that $\delta(Q_0, u_{ij}) = \{q_j^{p_i}\}$. Therefore, $L_{A_P, q_j^{p_i}} = u_{ij}^{-1} L$ and A_P is an RFSA.

Let $1 \leq i \leq n$, $0 \leq j < p_i$ and $1 \leq k \leq n$, $0 \leq l < p_k$.

If $p_i - j \neq p_k - l$, then $a^{p_i - j} \in L_{A_P, q_j^{p_i}} \setminus L_{A_P, q_l^{p_k}}$.

If $p_i - j = p_k - l$, then $a^{2p_i - j} \in L_{A_P, q_j^{p_i}} \setminus L_{A_P, q_l^{p_k}}$.

Therefore, all residual languages $L_{A_P, q_j^{p_i}}$ are different, none of them is included in another one and A_P has the same number of states as the canonical RFSA.

We can check that for every $u \in \Sigma^{<N}$, $|\delta(Q_0, u)| > 1$. So, the smallest word u_q such that $L_{A_P,q} = u_q^{-1} L$ has a length $|u_q| \geq N$.

Now, let f be some polynomial. We can choose different prime numbers $p_1, \ldots, p_n$ such that $p_1 \times \ldots \times p_n > f(p_1 + \ldots + p_n) = f(|A_P|)$. □

Next proposition shows that the simplified canonical RFSA can have far less transitions that the canonical RFSA.

Proposition 6.4. The number of transitions of the canonical RFSA of a language L can be quadratic wrt the number of transitions of the equivalent simplified canonical RFSA.

Proof:
We can verify this proposition on languages $L_n = a^* a^n$ where $n \in \mathbb{N}$.

It is easy to verify that the number of transitions of the simplified canonical RFSA of L is $N_Q = n+1$ and that the number of transitions of the canonical RFSA is $(N_Q^2 + 3N_Q)/2 - 1$.

For $n = 3$, the canonical RFSA and simplified canonical RFSA are represented in Figures 18 and 19. □

7. RFSAs over a one-letter alphabet

Here, we consider the case where the underlying alphabet possesses only one letter and we compare the state complexity of DFAs and RFSAs. Such a study has already been done for DFAs and NFAs in [2] where it has been shown that the minimal DFA equivalent to a given NFA with m states could have $\Theta(e^{\sqrt{m \log m}})$ states. Here we show that the number of states of the minimal DFA over a one-letter alphabet is at most quadratic in terms of the number of states of the equivalent canonical RFSA.

In all Section 7, we suppose that $\Sigma = \{a\}$, L is a non-empty regular language over Σ and $A = \langle \Sigma, Q, q_0, F, \delta \rangle$ is the trimmed minimal DFA which recognizes L. Let us set $n_L = |Q| - 1$. Let us define $q_i = \delta(q_0, a^i)$ and for sake of simplicity, let $L_i = L_{A,q_i}$, for $0 \leq i \leq n_L$.

If L is infinite, $\delta(q_0, a^i) \neq \emptyset$ for any integer i and previous notations can be extended: for every integer i, let $q_i = \delta(q_0, a^i)$ and $L_i = L_{A,q_i}$. Let m_L be the smallest index such that $q_{m_L} = \delta(q_{n_L}, a)$ and let $d = n - m_L + 1$ be the length of the loop (see Figure 20). Note that $d > 0$ and that for all $i \geq n_1$ and every non-negative integer k, $q_{i+kd} = q_i$.

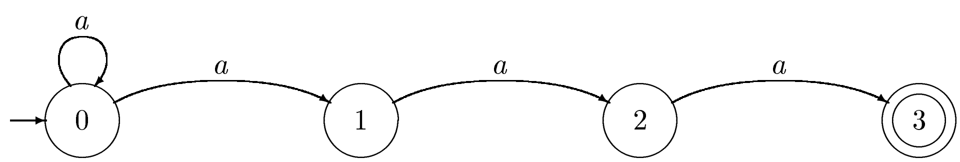

Figure 17. An NFA A which recognizes a^*a^3; A^R is deterministic.

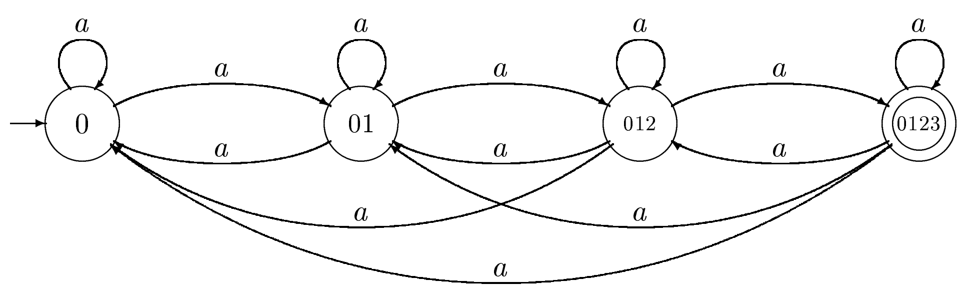

Figure 18. The canonical RFSA $C(A)$ which recognizes a^*a^3.

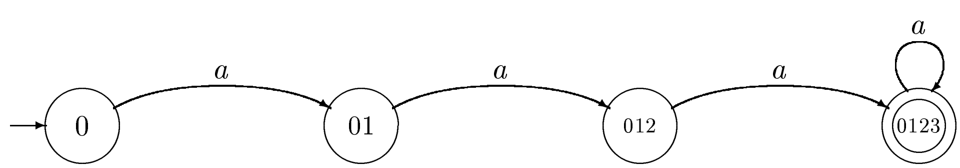

Figure 19. The simplified canonical RFSA $C'(A)$ which recognizes a^*a^3.

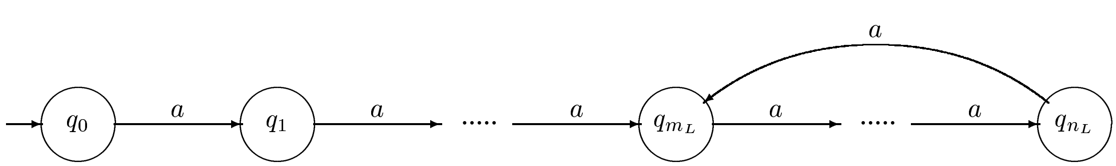

Figure 20. A minimal DFA that recognizes an infinite regular language over $\{a\}$ (final states have been omitted).

7.1. Inclusion relations between residual languages of an infinite regular language over a one-letter alphabet

We suppose in this subsection that L is infinite.

Lemma 7.1. For all integers i, j, k,

$$L_i \subseteq L_j \Rightarrow L_{i+k} \subseteq L_{j+k}.$$

Proof:
Let $u \in L_{i+k}$. We have $a^{i+k}u \in L$, i.e. $a^k u \in L_i$. Then $a^k u \in L_j$ and $a^{j+k}u \in L$. Therefore $u \in L_{j+k}$. □

Lemma 7.2. For all integers i, j,

$$L_i \subseteq L_j \Rightarrow d \text{ divides } (i - j).$$

Moreover, if $L_i \subsetneq L_j$, $min(i, j) < m_L$.

Proof:
Let r be an integer such that $rd - i + j \geq 0$ and $m_L + rd - i \geq 0$. Let $i_1 = i + (m_L + rd - i)$ and $j_1 = j + (m_L + rd - i)$. We have $L_{i_1} = L_{m_L+rd} = L_{m_L} \subseteq L_{j_1}$ and $j_1 \geq m_L$. Applying Lemma 7.1 with $k = j_1 - m_L$ we obtain,

$$L_{m_L} \subseteq L_{j_1} \subseteq L_{2j_1-m_L} \subseteq \ldots \subseteq L_{d(j_1-m_L)+m_L} = L_{m_L}.$$

As A is minimal, this implies that $q_{j_1} = q_{m_L}$, i.e. that d divides $j_1 - m_L$ and also $i - j$.

As $max(i, j) = min(i, j) + \frac{max(i,j)-min(i,j)}{d} \cdot d$, if $L_i \subsetneq L_j$, we must have $min(i, j) < m_L$. □

Proposition 7.1. We have the following cases

1. If A is a loop, i.e. if $m_L = 0$, then there are no inclusion relations between residual languages.
2. If there are no inclusion relations between L_{m_L-1} and L_{n_L}, then there are no inclusions between residual languages.
3. If $L_{m_L-1} \subsetneq L_{n_L}$, then $[L_i \subsetneq L_j \Rightarrow i < j]$.
4. If $L_{n_L} \subsetneq L_{m_L-1}$, then $[L_i \subsetneq L_j \Rightarrow i > j]$.

Proof:
The first assertion is clear from Lemma 7.2.

If there exist some i, j such that $L_i \subsetneq L_j$, we have $min(i, j) \leq m_L - 1$. If $k = m_L - 1 - min(i, j)$, then $L_{min(i,j)+k} = L_{m_L-1}$ and $L_{max(i,j)+k} = L_{n_L}$. Now, using the Lemma 7.1, the three last points are clear. □

7.2. Prime and composite residual languages

Proposition 7.2. Every non-empty residual language of a finite regular language over a one-letter alphabet is prime.

Proof:
Let L be a finite regular language over Σ and let A be its trimmed minimal DFA. We have $q_{n_L} \in F$ and $\delta(q_{n_L}, a) = \emptyset$. For any $0 \leq i \leq n_L$, $a^{n_L-i} \in L_i$. Let $j \neq i$. If $a^{n_L-i} \in L_j$, then $j < i$ and $a^{n_L-j} \notin L_i$, i.e. $L_j \not\subseteq L_i$. Hence, L_i is prime. $\square$

Lemma 7.3. Let L be an infinite regular language over a one-letter alphabet and let A be its trimmed minimal DFA. If some residual language of L is composite, then A is not a loop, i.e. $m_L > 0$, and $L_{m_L-1} \subsetneq L_{n_L}$.

Proof:
Suppose that some residual language of L is composite. From Proposition 7.1, $m_L > 0$ and there must exists an inclusion relation between L_{m_L-1} and L_{n_L}. Suppose that $L_{n_L} \subsetneq L_{m_L-1}$. Let q_j be a composite state. From Proposition 7.1 and Lemma 7.2, we have $j < m_L$. Let $u \in L_{m_L-1}$. We have $a^{m_L-1-j}u \in L_j$. Let i such that $L_i \subsetneq L_j$ and $a^{m_L-1-j}u \in L_i$. We have $u \in L_{m_L-1+i-j}$. From Proposition 7.1, we have $i > j$ and from Lemma 7.2, we have $i - j = rd$ with $r > 0$. So, $u \in L_{m_L-1+i-j} = L_{n_L+(m-1)d} = L_{n_L}$. Therefore, $L_{m_L-1} = L_{n_L}$ which is contradictory. $\square$

Proposition 7.3. Let L be a regular language over a one-letter alphabet and let A be its trimmed minimal DFA. If some state q of A is composite, then all the states which follow q are composite too.

Proof:
From previous lemmas, L is infinite, A is not a loop, $L_{m_L-1} \subsetneq L_{n_L}$ and $L_i \subsetneq L_j$ implies $i < j$. Let i be the first index such that L_i is a composite residual language. Let $R_i = \{j \mid j < i$ and $L_j \subsetneq L_i\}$. We have $L_i = \bigcup_{j \in R_i} L_j$. It is easy to verify that for every $i \leq k \leq n_L$, we have $L_k = \bigcup_{j \in R_i} L_{j+k-i}$ and $L_{j+k-i} \neq L_k$ for every $j \in R_i$. $\square$

7.3. Ratio between the size of the minimal DFA and the canonical RFSA for a one-letter alphabet regular language

If all states of the minimal DFA A are prime, the canonical RFSA has the same size as the minimal DFA. From Proposition7.2 and Lemma 7.3, it remains to consider the case where $L_{m_L-1} \subsetneq L_{n_L}$. Let Q_P be the set of prime states of Q and $n_P = |Q_P|$. Due to Proposition 7.3, $i \geq n_P$ implies that q_i is composite. Let $A_S = \langle \Sigma, Q_P, \{q_0\}, F \cap Q_P, \delta_S \rangle$ be the trimmed NFA built from $\phi(A, q_{n_P})$ where ϕ is the reduction operator defined in Section 4.3: A_S is an RFSA recognizing L that has the same number of states as the canonical RFSA. Let n_0 be the smallest index such that $q_{n_0} \in \delta_S(q_{n_P-1}, a)$.

Lemma 7.4. For every $i \geq 0$, we have $|\delta_S(q_0, a^i)| \leq |\delta_S(q_0, a^{i+1})|$. Moreover, if $|\delta_S(q_0, a^i)| = |\delta_S(q_0, a^{i+n_P-n_0})|$ for some $i \geq n_0$, then $\delta_S(q_0, a^i) = \delta_S(q_0, a^{i+n_P-n_0})$.

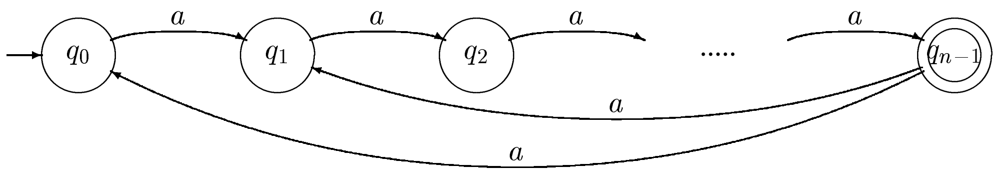

Figure 21. The minimal DFA corresponding to this canonical RFSA has $(n-1)^2+2$ states.

Proof:
If $i < n_P$, $\delta_S(q_0, a^i) = \{q_i\}$. Now let $i \geq n_P$. Due to the way δ_S is built, $q_j \in \delta_S(q_0, a^i)$ implies $n_0 \leq j < n_P$. We have $\delta_S(q_0, a^{i+1}) = \bigcup_{q_j \in \delta_S(q_0,a^i)} \delta_S(q_j, a)$. If $j < n_P - 1$, then $\delta_S(q_j, a) = \{q_{j+1}\}$ and if $j = n_P - 1$, then $q_{n_0} \in \delta_S(q_j, a)$. And as $q_{n_0} \notin \bigcup_{\{q_j \in \delta_S(q_0,a^i) | j < n_P - 1\}} \delta_S(q_j, a)$, we have $|\delta_S(q_0, a^{i+1})| \geq |\delta_S(q_0, a^i)|$, which proves the first point.

For every $j \geq n_0$, we have $q_j \in \delta_S(q_j, a^{n_P - n_0})$. So, for every $i \geq n_0$, $\delta_S(q_0, a^i) \subseteq \delta_S(q_0, a^{i+n_P-n_0})$. This proves the second point. □

Proposition 7.4. Let L be a regular language over a one-letter alphabet. Let n_R (resp. n_P) be the number of non empty residual languages (resp. prime residual languages) of L. Then,

$$n_R \leq (n_P - 1)^2 + 2.$$

Proof:
If $n_R = n_P$, the proposition is clear. Otherwise, let A be the minimal DFA of L and let $A_S = \langle \Sigma, Q_S, \{q_0\}, F_S, \delta_S \rangle$ be the trimmed NFA built from $\phi(A, q_{n_P})$. The number of reachable states in A_S is an upper bound for n_R. Let us define the function f by $f(i) = |\delta_S(q_0, a^i)|$ for any integer i. We verify that:

- For every n such that $1 < n < n_P$, we can show using Lemma 7.4 that either there are $n_P - n_0$ states q_i such that $f(i) = n$ but in this case there is no index i such that $f(i) > n$, or there are at most $n_P - n_0 - 1$ states q_i such that $f(i) = n$.
- There is at most one state q_i such that $f(i) = n_P - n_0$.

From this, we can calculate that $n_R \leq n_P + (n_P - 1)(n_P - 2) + 1 = (n_P - 1)^2 + 2$ states.

□

The upper bound is reached by the automaton described in Figure 21.

8. Complexity results

We have defined the notions of RFSAs, saturated automata, canonical RFSAs; in this section, we evaluate the complexity of constructions and decision problems linked to them.

We shall mainly use the following classical complexity results concerning finite automata (quoted from [6]).

Proposition 8.1. Deciding whether two NFAs recognize the same language is a $PSPACE-complete$ problem.

As an immediate corollary: given two NFAs A and A', deciding whether $L_A \subseteq L_{A'}$ is a $PSPACE-complete$ problem. The problem is quadratic if A and A' are DFAs.

Deciding whether the intersection of two DFAs is empty can be done in quadratic time. On the other hand:

Proposition 8.2. Deciding whether the intersection of n DFAs is empty or not is a $PSPACE-complete$ problem.

The first notion that we defined is *saturation*. Clearly, deciding whether a DFA is saturated is a polynomial problem. For NFAs, we have the following result.

Proposition 8.3. Deciding whether an NFA is saturated is a $PSPACE-complete$ problem.

Proof:
Given an oracle which decides whether the language recognized by a given NFA is included in another one, we can build the saturated of a given NFA within polynomial time.

Given an oracle which builds the saturated of a given NFA, we can say whether a given NFA is saturated within polynomial time.

It remains to prove that the inclusion problem between two languages represented by NFAs polynomially reduces to the problem of deciding whether an NFA is saturated.

Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ and $A' = \langle \Sigma, Q', Q'_0, F', \delta' \rangle$ be two NFAs. We can suppose that A and A' are trimmed, that $Q \cap Q' = \emptyset$ and that they have a unique initial state which cannot be reached from other states. Let $Q_0 = \{q_0\}$, $Q = \{q_0, q_1, \ldots, q_l\}$, $Q'_0 = \{q'_0\}$ and $Q' = \{q'_0, q'_1, \ldots, q'_{l'}\}$. We complete the alphabet Σ by adding $l + l' + 2$ new letters $x_1, \ldots, x_l, x'_1, \ldots, x'_{l'}, z, t$: let Σ' be the new alphabet. We consider two new states q_e and q_f and let $B = \langle \Sigma', Q \cup Q' \cup \{q_e, q_f\}, \{q'_0\}, F \cup F' \cup \{q_f\}, \delta'' \rangle$ where δ'' contains the transitions of $\delta \cup \delta'$ and the transitions defined below:

$$q_f \in \delta(q_i, x_i) \text{ for } i = 1 \ldots l \text{ and } q_f \in \delta(q'_i, x'_i) \text{ for } i = 1 \ldots l',$$

$$q_e \in \delta(q_0, x) \cap \delta(q'_0, x) \text{ for } x \in \Sigma,$$

$$q_e \in \delta(q_e, x) \text{ for } x \in \Sigma,$$

$$q_f \in \delta(q_e, x_i) \text{ for } i = 1 \ldots l \text{ and } q_f \in \delta(q_e, x'_i) \text{ for } i = 1 \ldots l',$$

$$q_f \in \delta(q_f, z) \text{ and } q_f \in \delta(q_e, t) \text{ (see Figure 22).}$$

Now, it is easy to verify that B is saturated if and only if $L_A \not\subseteq L_{A'}$. □

Next proposition shows that deciding whether a given NFA is an RFSA is also difficult.

Proposition 8.4. Deciding whether an NFA is an RFSA is a $PSPACE-complete$ problem.

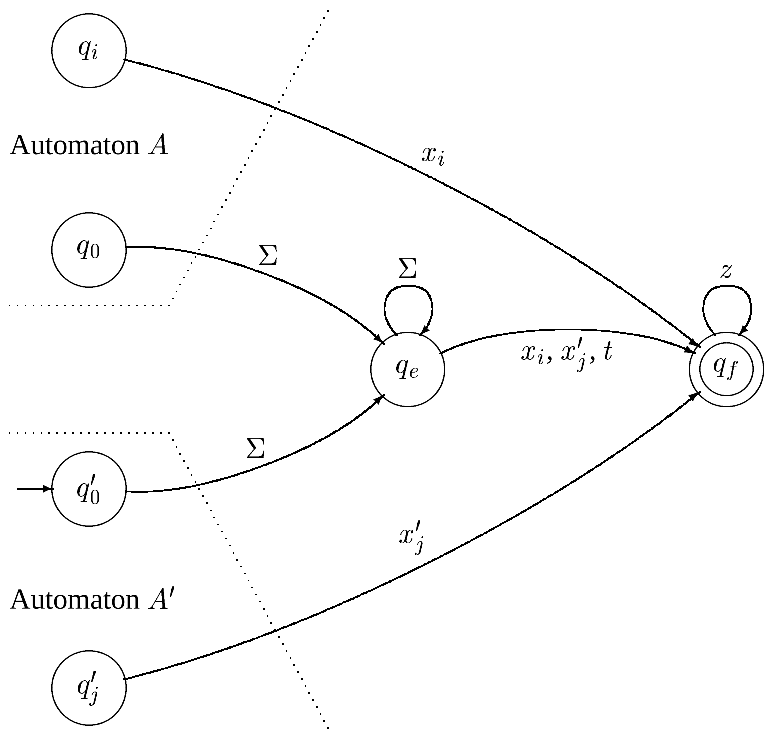

Figure 22. The automaton is saturated iff $L_{q_0} \not\subseteq L_{q'_0}$.

Proof:

First, we prove that the problem of deciding whether the union of n regular languages described by DFAs is equal to Σ^* can be polynomially reduced to the problem of deciding whether an NFA is a RFSA.

We can consider n DFAs $A^1 = \langle \Sigma, Q^1, q^1, Q_F^1, \delta^1 \rangle, \ldots, A^n = \langle \Sigma, Q^n, q^n, Q_F^n, \delta^n \rangle$ where $i \neq j$ implies $Q^i \cap Q^j = \emptyset$.

Let $x_1, \ldots, x_n, y_1, \ldots, y_n, a, b$ be $2n+2$ new letters. We can build the NFA $A = \langle \Sigma, Q, Q_I, Q_F, \delta \rangle$ where

- $Q = \bigcup_{i=1\ldots n} Q^i \cup \{q_1, \ldots, q_n, q_e, q_f, q_g\}$ where $\{q_1, \ldots, q_n, q_e, q_f, q_g\}$ are new states,
- $Q_I = \{q_1, \ldots, q_n, q_e, q_f\}$
- $Q_F = \bigcup_{i=1\ldots n} Q_F^i \cup \{q_g\}$
- $\delta = (\bigcup_{i=1\ldots n} \delta^i) \cup (\bigcup_{i=1\ldots n} \{(q_i, x_i, q^i), (q_i, y_i, q_i), (q_e, a, q^i)\}) \cup \{(q_e, b, q_e), (q_f, b, q_f), (q_f, a, q_g)\} \cup \{(q_g, x, q_g) \mid x \in \Sigma\}$ (see Figure 23).

Every states, except maybe q_e, defines a residual language of the described language:

- states q_i correspond to residual languages of words y_i,
- states q of sets Q^i correspond to residual languages of $x_i u$ where $\delta^i(q^i, u) = q$,
- q_f corresponds to the residual language of b and
- q_g corresponds to the residual language of a.

State q_e defines a residual language of the language if and only if the union of recognized languages by A_i automaton is equal to Σ^*. That is, A is an RFSA if and only if the union of languages described by A_i automaton is equal to Σ^*

It remains to show that we can decide whether a given NFA A is an RFSA within polynomial space. Consider the subset construction defined in Section 5. The reachable set of states of A can be enumerated within polynomial space: therefore, for each state q of A and for each reachable set of states, decide whether they define equal regular languages. The NFA A is an RFSA if and only if all its states are equivalent to some reachable set of states. □

Using similar techniques, it can be shown that deciding whether the saturated of a given DFA is the canonical RFSA is a PSPACE-complete problem. Hint: given n DFAs $A_1, \ldots, A_n$ over Σ, it is easy to build a DFA over an extended alphabet Σ' such that $\cup_{i=1}^n L_{A_i} \neq \Sigma^*$ iff the saturated of A is the canonical RFSA.

9. Conclusion

The class of RFSAs can be viewed as an intermediary class between the DFAs and the NFAs. Based on an important property of the DFAs, namely the fact that each state of an automaton A must define a residual language of the language recognized by A, the RFSAs also share with the DFAs the property

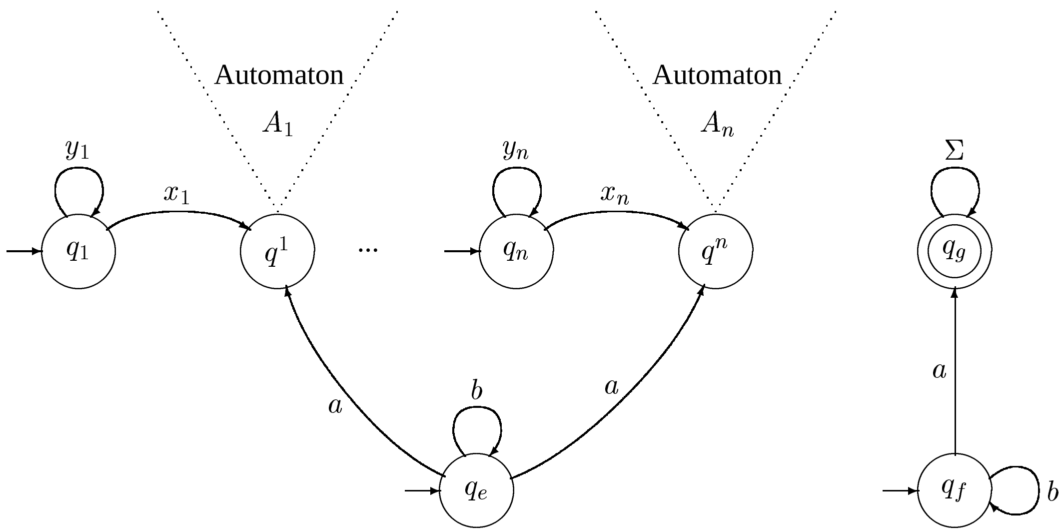

Figure 23. A is an RFSA iff the union of languages described by the A_i is equal to Σ^*.

of having a minimal canonical form. On the other hand, the canonical RFSAs can be in some cases as concise as minimal NFAs.

It has been indicated in the introduction that the ideas developped 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 DFAs. For example, it has been shown that Regular Languages represented by DFAs can be infered from *given data* ([7, 8]). In this framework, classical inference algorithms such as RPNI [13] 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 $\Sigma^* a \Sigma^n$ cannot be infered efficiently using these algorithms. Hence, it is a natural idea to think of using other kind of representations for regular languages, such as NFAs. Unfortunately, it has been shown that Regular Languages represented by NFAs cannot be efficiently infered from given data ([8]). The main difficulty that arises when one try to build an NFA from examples comes from the fact that states do not correspond to natural components of the associated language. So, we defined RFSAs in order to obtain an automata representation of regular languages for which states correspond to residual languages. RFSAs have been used to design grammatical inference algorithms in [3, 4].

References

[1] Brzozowski, J. A.: Canonical regular expressions and minimal state graphs for definite events, in: *Mathematical Theory of Automata*, vol. 12 of *MRI Symposia Series*, 1962, 52–561.

[2] Chrobak, M.: Finite automata and unary languages., *Theorical Computer Science*, 1986, 47(2):149–158.

[3] Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using non deterministic finite automata, *ICGI'2000, 5th International Colloquium on Grammatical Inference*, 1891, Springer Verlag, 2000.

[4] Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using RFSA, *ALT 2001*, number 2225 in Lecture Notes in Artificial Intelligence, Springer Verlag, 2001.

[5] Denis, F., Lemay, A., Terlutte, A.: Residual Finite State Automata, *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science*, number 2010 in Lecture Notes in Computer Science, Springer Verlag, 2001.

[6] Garey, M. R., Johnson, D. S.: *Computers and Intractability, a Guide to the Theory of NP-Completness*, W.H. Freeman and Co, San Francisco, 1979.

[7] Gold, E.: Complexity of Automaton Identification from Given Data, *Inform. Control*, **37**, 1978, 302–320.

[8] Higuera, C. D. L.: Characteristic Sets for Polynomial Grammatical Inference, *Machine Learning*, **27**, 1997, 125–137.

[9] Hopcroft, J., Ullman, J.: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[10] Kleene, S. C.: Representation of Events in Nerve Nets and Finite Automata, in: *Automata Studies, Annals of Math. Studies 34* (C. Shannon, J. McCarthy, Eds.), New Jersey, 1956.

[11] Myhill, J.: *Finite Automata and the Representation of Events*, Technical Report 57-624, WADC, 1957.

[12] Nerode, A.: Linear Automaton Transformation, *Proc. American Mathematical Society*, 9, 1958.

[13] Oncina, J., Garcia, P.: Inferring regular languages in polynomial update time, *Pattern Recognition and Image Analysis*, 1992.

[14] Yu, S.: *Handbook of Formal Languages, Regular Languages*, vol. 1, chapter 2, Springer Verlag, 1997, 41–110.