Residual Languages and Probabilistic Automata

François Denis and Yann Esposito

LIF-CMI, UMR 6166
39, rue F. Joliot Curie
13453 Marseille Cedex 13 FRANCE
fdenis,esposito@cmi.univ-mrs.fr

Abstract. A stochastic generalisation of residual languages and operations on Probabilistic Finite Automata (PFA) are studied. When these operations are iteratively applied to a subclass of PFA called PRFA, they lead to a unique canonical form (up to an isomorphism) which can be efficiently computed from any equivalent PRFA representation.

1 Introduction

Probabilistic Automata are formal objects, equivalent to Hidden Markov Models under many aspects [6], which can be used to model stochastic processes in many application domains such as Pattern Recognition [1, 2], Information Extraction [3], Bioinformatics [4, 5]. A probabilistic automaton (PFA) has a structural component, which is a non deterministic automaton (NFA), and several continuous parameters which specify the probability for a state to be initial, to be terminal, and the probability to reach a state from another one while reading or emitting a given letter. A probabilistic automaton generates a regular stochastic language.

Determining an appropriate PFA structural component from a finite number of observations is an important open problem. In order to tackle this problem, it is necessary to identify subclasses of PFA which can be identified from given data. Deterministic PFA (PDFA), i.e. PFA whose structure is a deterministic NFA (DFA), have this property and have been used in several inference works [7–10]. Unfortunately, contrary to the case of non stochastic regular languages, the class of stochastic languages which can be represented by PDFA is a very restricted subclass of the class of regular stochastic languages and it is necessary to find out new richer subclasses of PFA.

Several works have pointed out the importance of residual languages for Grammatical Inference [11, 12]. A residual language of a language L is any language of the form $\{w|uw \in L\}$, for some word u. Most classical inference algorithms try to identify the residual languages of the target language L from a finite sample of L. A stochastic generalisation of residual languages has been introduced in [13] and has lead to the definition of Probabilistic Residual Finite State Automata (PRFA). A PRFA is a PFA whose states define residual languages of the language which is generated.

Here, we methodically pursue this study by introducing a reduction operator and a saturation operator which act on PFA (Section 3). We show that if a stochastic language P can be generated by a PRFA, then iteratively applying the reduction and saturation operators to any PRFA which generates P provide a single object (up to an isomorphism): the canonical PRFA of L (Section 4). These canonical PRFAs are based on particular residual languages of P which cannot be decomposed by using other residual languages of P: we call them prime residual languages. Finally, we show in Section 5 that all the operations that we define are polynomial (whereas similar operations for non stochastic languages are PSPACE-complete [14]).

2 Preliminaries

2.1 Automata and languages

Let Σ be a finite *alphabet*, and Σ^* be the set of words on Σ. We denote by ε the empty word and by $|u|$ the length of a word u. A *language* is a subset of Σ^*. A *nondeterministic finite automaton* (NFA) is a tuple $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ 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* defined from $Q \times \Sigma$ to 2^Q. We also denote by δ the extended transition function defined from $2^Q \times \Sigma^*$ to 2^Q. An NFA is *deterministic* (DFA) if Q_0 contains only one element q_0 and if $\forall q \in Q$, $\forall x \in \Sigma$, $Card(\delta(q,x)) \leq 1$. 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 $\{v \in \Sigma^* \mid \delta(Q', v) \cap F \neq \emptyset\}$. When Q' contains exactly one state q, we simply denote $L_{A,Q'}$ by $L_{A,q}$. It can be proved that the class of recognizable languages is identical to the class of regular languages (Kleene Theorem) and 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). Let L be a language and u be a word. The *residual language* of L wrt u is $u^{-1}L = \{v \mid uv \in L\}$. A *Residual Finite State Automaton* (RFSA) is an NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ such that, for each $q \in Q$, $L_{A,q}$ is a residual language of L_A [14].

2.2 Probabilistic automata and stochastic languages

A *probabilistic finite state automaton (PFA)* is a tuple $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ where Q is a finite set of states, $\varphi : Q \times \Sigma \times Q \to [0,1]$ is the transition function, $\iota : Q \to [0,1]$ is the probability for each state to be initial and $\tau : Q \to [0,1]$ is the probability for each state to be terminal. A PFA need satisfy $\sum_{q \in Q} \iota(q) = 1$ and for each state q, $\tau(q) + \sum_{a \in \Sigma} \sum_{q' \in Q} \varphi(q, a, q') = 1$. Let φ also denote the extension of the transition function, defined on $Q \times \Sigma^* \times Q$ by $\varphi(q, wa, q') = \sum_{q'' \in Q} \varphi(q, w, q'')\varphi(q'', a, q')$ and $\varphi(q, \varepsilon, q') = 1$ if $q = q'$ and 0 otherwise. We extend φ again on $Q \times 2^{\Sigma^*} \times 2^Q$ by $\varphi(q, U, R) = \sum_{w \in U} \sum_{r \in R} \varphi(q, w, r)$. The set of initial states is defined by $Q_I = \{q \in Q \mid \iota(q) > 0\}$, the set of reachable states is defined by $Q_{reach} = \{q \in Q \mid \exists r \in Q_I, \varphi(r, \Sigma^*, q) \neq 0\}$ and the set of terminal states is defined by $Q_T = \{q \in Q \mid \tau(q) > 0\}$. The *support* of a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ is the NFA $\langle \Sigma, Q, Q_I, Q_T, \delta \rangle$ such that $\delta(q, x) = \{q' | \varphi(q, x, q') \neq 0\}$. A PFA is *admissible* if for any $q \in Q_{reach}$, $\varphi(q, \Sigma^*, Q_T) \neq 0$.We shall only consider admissible PFA. A *probabilistic deterministic finite state automaton (PDFA)* is a PFA whose support is deterministic.

A *stochastic language* on Σ is a function P defined from Σ^* to $[0,1]$ such that $\sum_{u \in \Sigma^*} P(u) = 1$. For any $W \subseteq \Sigma^*$, let $P(W) = \sum_{w \in W} P(w)$. Let $S(\Sigma)$ be the set of stochastic languages on Σ. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be an admissible PFA. Let P_A be the function defined on Σ^* by $P_A(u) = \sum_{q,q' \in Q \times Q} \iota(q)\varphi(q, u, q')\tau(q')$. It can be proved that P_A is a stochastic language on Σ which is called the stochastic language generated by A.

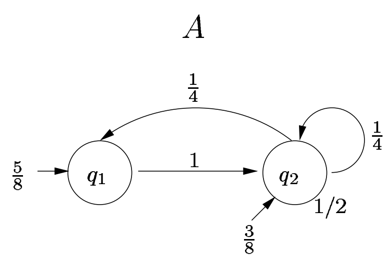

Fig. 1. An example of PFA A on $\Sigma = \{a\}$: $\iota(q_1) = 5/8, \iota(q_2) = 3/8, \phi(q_1, a, q_1) = 0, \phi(q_1, a, q_2) = 1, \phi(q_2, a, q_1) = 1/4, \phi(q_2, a, q_2) = 1/4, \tau(q_1) = 0$ and $\tau(q_2) = 1/2$. For sake of clarity, the letter a has not been drawn, nor null parameters such as $\phi(q_1, a, q_1)$ or $\tau(q_1)$. We have $P_A(\epsilon) = 3/16$ and $P_A(a) = 5/8 \cdot 1 \cdot 1/2 + 3/8 \cdot 1/4 \cdot 1/2 = 23/64$.

For every $q \in Q$, we denote by A_q the PFA $< \Sigma, Q, \varphi, \iota_q, \tau >$ where $\iota_q(q) = 1$. $P_{A,q} = P_{A_q}$ is the stochastic language generated from q. Note that for any word u and any state q, $\varphi(q, u, Q) = P_{A,q}(u\Sigma^*)$. Let $\mathcal{L}_A = \{P_{A,q} | q \in Q\}$. Let $A =< \Sigma, Q, \varphi_A, \iota_A, \tau_A >$ and $B =< \Sigma, Q, \varphi_B, \iota_B, \tau_B >$ be two PFAs. A and B are *equivalent* if they define the same stochastic language, i.e. if $P_A = P_B$. A and B are *state-equivalent* if $P_A = P_B$ and if for every $q \in Q$, $P_{A,q} = P_{B,q}$. A and B are *isomorphic* ($A \sim B$) if they are state-equivalent and if they have the same support.

We extend the notion of residual languages to the stochastic case as follows. Let P be a *stochastic* language, the *residual language* $u^{-1}P$ of P with respect to u associates with every word w the probability $u^{-1}P(w) = P(uw)/P(u\Sigma^*)$ if $P(u\Sigma^*) \neq 0$. If $P(u\Sigma^*) = 0$, $u^{-1}P$ is not defined.

Let $\mathcal{L} \subseteq S(\Sigma)$ be a finite set of stochastic languages. We define the convex hull of $\mathcal{L}$ by $conv(\mathcal{L}) = \{L \in S(\Sigma) | \exists L_1, \ldots, L_n \in \mathcal{L}, \exists \lambda_1, \ldots \lambda_n \geq 0 | L = \sum_{i=1}^{n} \lambda_i L_i\}$. For any $P \in conv(\mathcal{L})$, there exists a maximal subset of $\mathcal{L}$ that we denote by $cov(P, \mathcal{L})$ such that $P = \sum_{P_q \in cov(P,\mathcal{L})} \lambda_{P_q} P_q$ and $\lambda_{P_q} > 0$. We say that $\mathcal{L}$ is a *residual net* if for any $P \in \mathcal{L}$ and any letter $x \in \Sigma$, we have $x^{-1}P \in conv(\mathcal{L})$.

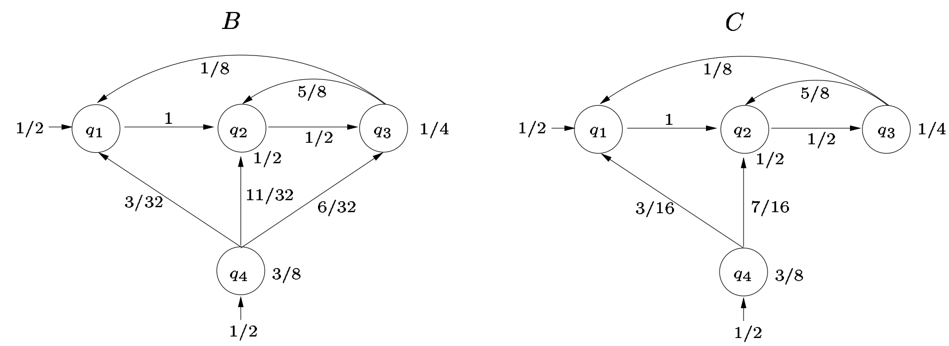

Fig. 2. B and C are two PFAs on $\Sigma = \{a\}$ which are state-equivalent but not isomorphic. They are equivalent to the PFA represented on Fig. 1.

Example 1. Consider the PFA B on Fig. 2. We have $P_B = (P_{B,q_1} + P_{B,q_4})/2$, so $P_B \in conv(\{P_{B,q_1}, P_{B,q_4}\})$. As $P_{B,q_1} \neq P_{B,q_4}$, $cov(P_{B,q_1}, \{P_{B,q_1}, P_{B,q_4}\}) = \{P_{B,q_1}\}$. The set $\{P_{B,q_1}, P_{B,q_4}\}$ is not a residual net. Indeed, $a^{-1}P_{B,q_1} \notin conv(\{P_{B,q_1}, P_{B,q_4}\})$ as $a^{-1}P_{B,q_1}(\epsilon) = 1/2$, $P_{B,q_1}(\epsilon) = 0$ and $P_{B,q_4}(\epsilon) = 3/8$. On the other hand, $\{P_{B,q_1}, P_{B,q_2}, P_{B,q_3}, P_{B,q_4}\}$ is a residual net.

A PRFA is a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ such that every state defines a residual language, i.e. such that $\forall q \in Q, \exists u \in \Sigma^*, L_{A,q} = u^{-1}P_A$ [13].

We denote by $\mathcal{L}_{PFA}(\Sigma)$ (resp. $\mathcal{L}_{PDFA}(\Sigma)$, resp. $\mathcal{L}_{PRFA}(\Sigma)$) the set of stochastic languages generated by some PFA (resp. PDFA, resp. PRFA). It can be shown that $\mathcal{L}_{PDFA}(\Sigma) \subsetneq \mathcal{L}_{PRFA}(\Sigma) \subsetneq \mathcal{L}_{PFA}(\Sigma)$ [13]. Each of these classes can be characterized in terms of residual languages [13].

Let P be a stochastic language:

- $P \in \mathcal{L}_{PDFA}(\Sigma)$ iff P has a finite number of residual languages.
- $P \in \mathcal{L}_{PRFA}(\Sigma)$ iff there exists a residual net $\mathcal{L}$ composed of residual languages of P such that $P \in conv(\mathcal{L})$.
- $P \in \mathcal{L}_{PFA}(\Sigma)$ iff there exists a residual net $\mathcal{L}$ such that $P \in conv(\mathcal{L})$.

3 Reduction and saturation of probabilistic finite automata

It is sometimes possible to suppress a state from a PFA while keeping the associated stochastic language. The reduction operator defined below takes as input a PFA A and a state q of A and outputs

- $\{A\}$ if $P_{A,q} \notin conv(\mathcal{L}_A \setminus \{P_{A,q}\})$,
- a set of PFAs equivalent to A which stem from the deletion of q otherwise.

Definition 1. *Let* $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ *be an admissible PFA, let* $q \in Q$*, let* $Q' = Q \setminus \{q\}$ *and let* $\Lambda_q^A = \{(\lambda_r)_{r \in Q'} \mid \lambda_r \in \mathbb{R}^{\geq 0}$ *and* $P_{A,q} = \sum_{r \in Q'} \lambda_r P_{A,r}\}$.

– *If $\Lambda_q^A = \emptyset$, i.e. $P_{A,q} \notin conv(\mathcal{L}_A \setminus \{P_{A,q}\})$, then $red(A,q) = \{A\}$,*
– *Otherwise, $red(A,q)$ is composed of the PFAs $A' = \langle \Sigma, Q', \varphi', \iota', \tau' \rangle$ such that there exists $(\lambda_r)_{r\in Q'} \in \Lambda_q^A$ such that*
  - $\tau' = \tau_{|Q'}$,
  - $\iota'(r) = \iota(r) + \lambda_r \iota(q)$, *for all* $r \in Q'$,
  - $\varphi'(r,x,s) = \varphi(r,x,s) + \lambda_s \varphi(r,x,q)$ *for all* $r,s \in Q'$ *and* $x \in \Sigma$.

It can easily be checked that every element in $red(A,q)$ is an admissible PFA. Note that for any $A' \in red(A,q)$ and any states r and s of A', $\varphi(r,x,s) \neq 0 \Rightarrow \varphi'(r,x,s) \neq 0$ and $\iota(r) \neq 0 \Rightarrow \iota'(r) \neq 0$. However, two different PFAs in $red(A,q)$ may have different support.

Example 2. Consider the PFA B defined on Fig. 2. We can show that $P_{B,q_3} = (P_{B,q_1} + P_{B,q_2})/2$ and that $P_{B,q_4} = (P_{B,q_1} + 5P_{B,q_2} + 2P_{B,q_3})/8 = (P_{B,q_1} + 3P_{B,q_2})/4$.

Proposition 1. *Let A be a PFA, let q be one of its states and let $A' \in red(A,q)$. Then, A' is equivalent to A and for any state r of A', $P_{A,r} = P_{A',r}$.*

Proof. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA, let $q \in Q$, let $A' = \langle \Sigma, Q', \varphi', \iota', \tau' \rangle \in red(A,q)$. Suppose that $A' \neq A$ and let $(\lambda_r) \in \Lambda_q^A$ be such that $P_{A,q} = \sum_{r\in Q'} \lambda_r P_{A,r}$. For any state r of Q', we have $P_{A',r}(\varepsilon) = \tau(r) = P_{A,r}(\varepsilon)$. Now, assume that for any word w of length $\leq k$ and any state r of Q' we have $P_{A',r}(w) = P_{A,r}(w)$. Let x be a letter, we have:

$$\begin{aligned} P_{A',r}(xw) &= \sum_{s\in Q'} \varphi'(r,x,s) P_{A',s}(w) = \sum_{s\in Q'} \left(\varphi(r,x,s) + \lambda_s \varphi(r,x,q)\right) P_{A,s}(w) \\ &= \sum_{s\in Q'} \varphi(r,x,s) P_{A,s}(w) + \varphi(r,x,q) \sum_{s\in Q'} \lambda_s P_{A,s}(w) \\ &= \sum_{s\in Q'} \varphi(r,x,s) P_{A,s}(w) + \varphi(r,x,q) P_{A,q}(w) \\ &= \sum_{s\in Q} \varphi(r,x,s) P_{A,s}(w) = P_{A,r}(xw). \end{aligned}$$

Then $P_{A',r} = P_{A,r}$ for any r of Q'. We remark that

$$\begin{aligned} P_{A'} &= \sum_{s\in Q'} \iota'(s) P_{A,s} = \sum_{s\in Q'} \left(\iota(s) + \lambda_s \iota(q)\right) P_{A,s} \\ &= \sum_{s\in Q'} \iota(s) P_{A,s} + \iota(q) \sum_{s\in Q'} \lambda_s P_{A,s} = \sum_{s\in Q} \iota(s) P_{A,s} = P_A. \end{aligned}$$

We shall say that a PFA is reduced if none of its states can be reduced while preserving the associated language.

Definition 2. *A PFA A is* reduced *if for every state q, $red(A,q) = \{A\}$.*

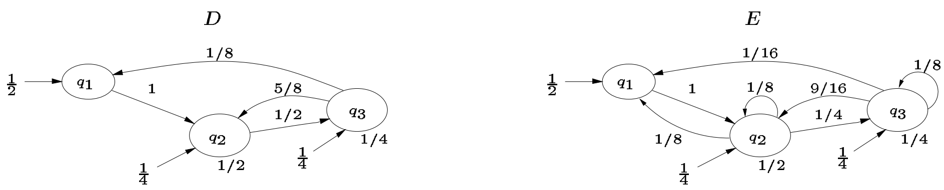

Fig. 3. $D \in \mathrm{red}\,(C, q_4)$ using $P_{C,q_4} = \frac{1}{2}P_{C,q_2} + \frac{1}{2}P_{C,q_3}$ and $E \in \mathrm{sat}\,(D)$.

Proposition 2. *Every PFA is equivalent to a reduced PFA.*

Proof. Any PFA is reduced or equivalent to a PFA which has less states. $\square$

Two elements of $red(A, q)$ may not be isomorphic, even if they are reduced (see Fig. 4). We shall obtain a unique element (up to an isomorphism) by adding as much transitions as possible while preserving the associated stochastic language. This will be achieved by using the saturation operator.

Definition 3. *Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA. We define $sat(A)$ as the set of PFA $A' = \langle \Sigma, Q, \varphi', \iota', \tau \rangle$ such that for any states $q, r \in Q$, any letter $x \in \Sigma$, there exist non negative real numbers $\lambda^x_{q,r}$ such that*

- $x^{-1}P_{A,q} = \sum_{r \in Q} \lambda^x_{q,r} P_{A,r}$ *and* $[P_{A,r} \in cov(x^{-1}P_{A,q}, \mathcal{L}_A) \Rightarrow \lambda^x_{q,r} > 0]$,
- $P_A = \sum_{r \in Q} \iota'(r) P_{A,r}$ *and* $[P_{A,r} \in cov(P_A, \mathcal{L}_A) \Rightarrow \iota'(r) > 0]$,
- $\varphi'(r, x, s) = \lambda^x_{r,s}\varphi(r, x, Q)$.

It can easily be checked that any element in $sat(A)$ is an admissible PFA.

Proposition 3. *Let A be a PFA and let $A' \in sat(A)$. Then, A and A' are state-equivalent and for any state q of A, $P_{A,q} = P_{A',q}$.*

Proof. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA and let $A' = \langle \Sigma, Q, \varphi', \iota', \tau \rangle \in sat(A)$. We have for any state q, $P_{A',q}(\varepsilon) = \tau(q) = P_{A,q}(\varepsilon)$. Now assume that for any word w of length $\leq k$, and for any state q, $P_{A',q}(w) = P_{A,q}(w)$. Let x be a letter, we have:

$$
\begin{aligned}
P_{A',q}(xw) &= \sum_{r \in Q} \varphi'(q, x, r) P_{A',r}(w) \\
&= \sum_{r \in Q} \lambda^x_{q,r} \varphi(q, x, Q) P_{A,r}(w) \text{ where the } \lambda^x_{q,r} \text{ satisfy the conditions of Def. 3,} \\
&= \varphi(q, x, Q) \cdot \left[\sum_{r \in Q} \lambda^x_{q,r} P_{A,r} \right](w) = P_{A,q}(x\Sigma^*) \cdot \left[x^{-1} P_{A,q} \right](w) = P_{A,q}(xw).
\end{aligned}
$$

Then for any state q, $P_{A,q} = P_{A',q}$. We remark that $P_{A'} = \sum_{q \in Q} \iota'(q) P_{A',q} = \sum_{q \in Q} \iota'(q) P_{A,q} = P_A$, which concludes the proof. $\square$

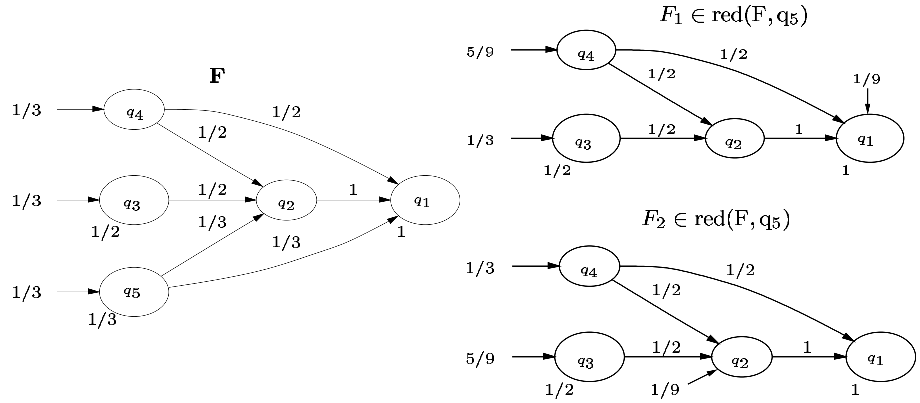

Fig. 4. Two non isomorphic reduced PFAs of F.

We say that a PFA is saturated if it has a maximal number of transitions.

Definition 4. *A PFA A is* saturated *if A is in $sat(A)$.*

Next proposition states a number of properties of the *sat* operator. Proofs are omitted.

Proposition 4. *If A and B are state-equivalent, then $sat(A) = sat(B)$. A PFA A is saturated iff for any states r, s and any letter x we have $P_{A,s} \in cov(x^{-1}P_{A,r}, \mathcal{L}_A) \Rightarrow \varphi(r, x, s) \neq 0$ and $P_{A,r} \in cov(P_A, \mathcal{L}_A) \Rightarrow \iota(r) \neq 0$. Any element B of $sat(A)$ is saturated and moreover $sat(A) = sat(B)$. Any two elements of $sat(A)$ are isomorphic. If B is isomorphic to A and if A is saturated then B is saturated.*

Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA and let $\mathcal{A}$ be the set of PFAs $A' = \langle \Sigma, Q, \varphi', \iota', \tau \rangle$ such that A' is state equivalent to A. Define the relation $\prec$ on $\mathcal{A}$ by:

$$B \prec C \text{ iff } \iota_B(q) \neq 0 \Rightarrow \iota_C(q) \neq 0 \text{ and } \varphi_B(q, x, q') \neq 0 \Rightarrow \varphi_C(q, x, q') \neq 0$$

for any states q, q' and any letter x, where $B = \langle \Sigma, Q, \varphi_B, \iota_B, \tau \rangle$ and $C = \langle \Sigma, Q, \varphi_C, \iota_C, \tau \rangle$.

Proposition 5. *$(\mathcal{A}/\sim, \prec)$ is a semi-upper lattice whose maximal element is $sat(A)$.*

Proof. Let $B = \langle \Sigma, Q, \varphi_B, \iota_B, \tau \rangle, C = \langle \Sigma, Q, \varphi_C, \iota_C, \tau \rangle \in \mathcal{A}$. Define the PFA $B \vee C = \langle \Sigma, Q, \varphi', \iota', \tau \rangle$ where for any states r, s and any letter x, we have $\iota'(r) = (\iota_B(r) + \iota_C(r))/2$ and $\varphi'(r, x, s) = (\varphi_B(r, x, s) + \varphi_C(r, x, s))/2$. Check that $B \prec B \vee C$, $C \prec B \vee C$ and that for any D such that $B \prec D$ and $C \prec D$, we have $B \vee C \prec D$.

Now, it is clear from the definition of *cov* and from Prop. 4 that the elements of $sat(A)$ define a class which is the maximal element of $(\mathcal{A}/\sim, \prec)$. □

Let $\mathcal{A}$ be a set of PFAs defined on the same alphabet and the same set of states Q and let $q \in Q$. Define $sat(\mathcal{A}) = \cup_{A\in\mathcal{A}} sat(A)$ and $red(\mathcal{A}, q) = \cup_{A\in\mathcal{A}} red(A, q)$.

Proposition 6. *Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA and let q be a state of A. Let $B \in sat(red(A, q))$ and $C \in red(sat(A), q)$. Then B and C are isomorphic.*

Proof. Let $A' \in sat(A)$ such that $C \in red(A', q)$. Let r, s be states of C and let x be a letter such that $P_{C,s} \in cov(x^{-1}P_{C,r}, \mathcal{L}_C)$. From Prop. 1, $P_{A',s} \in cov(x^{-1}P_{A',r}, \mathcal{L}_{A'})$. From Prop. 4, A' is saturated and then $\varphi_{A'}(r, x, s) \neq 0$. Therefore, $\varphi_C(r, x, s) \neq 0$. In a similar way, it can be shown that if $P_{C,s} \in cov(P_C, \mathcal{L}_C)$ then $\iota_C(s) \neq 0$. From Prop. 4, C is saturated. Now, $sat(B) = sat(C)$ as B and C are state-equivalent ; $C \in sat(B)$ as C is saturated. Therefore C is isomorphic to B from Prop. 5. □

Proposition 7. *Let A be a saturated PFA and let q_1 and q_2 be two states from A. Let $B \in red(red(A, q_1), q_2)$ and $C \in red(red(A, q_2), q_1)$. Then, B and C are isomorphic.*

Proof. From Prop. 1, B and C are state-equivalent. Then, from Prop. 4, $sat(B) = sat(C)$. From Prop. 6, B and C are saturated. So, $B \in sat(C)$ and B and C are isomorphic. □

Given a PFA, saturing and reducing it while it is possible provides an equivalent PFA which is reduced, saturated and unique up to an isomorphism. However, there exist non isomorphic reduced saturated equivalent PFAs (see Fig. 5).

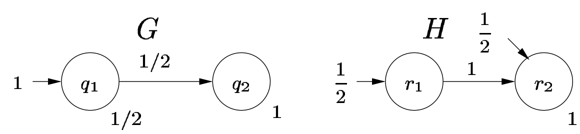

Fig. 5. Two non isomorphic reduced saturated equivalent PFAs.

4 Canonical PRFA

The application of reduction or saturation to a PRFA always yields a PRFA.

Proposition 8. *Let A be a PRFA and let q be a state of A. Then, all elements of $red(A, q) \cup sat(A)$ are PRFAs.*

Proof. As reduction and saturation do not change the languages generated from the states, every state will continue to generate a residual language.

Definition 5. *Let P be a stochastic language, a residual language $u^{-1}P$ is said to be* composed *if there exist residual languages $u_1^{-1}P, \ldots, u_k^{-1}P$ such that $u^{-1}P \neq u_i^{-1}P$ for any $i = 1, \ldots, k$ and such that $u^{-1}P \in conv(\{u_1^{-1}P, \ldots, u_k^{-1}P\})$. A residual language is* prime *if and only if it is not composed.*

Clearly, a stochastic language generated by a PRFA with n states has at most n prime residual languages. The converse is false. Let $A = \langle\{a\}, \{q_1, q_2\}, \varphi, \iota, \tau\rangle$ be a PFA such that $\iota(q_1) = \iota(q_2) = 1/2$, $\tau(q_1) = 1-\alpha$, $\tau(q_1) = 1-\beta$, $\varphi(q_1, a, q_1) = \alpha$, $\varphi(q_2, a, q_2) = \beta$. The stochastic language P_A has only one prime residual language and cannot be generated by a PRFA. We have $P(a^n) = \frac{\alpha^n(1-\alpha)+\beta^n(1-\beta)}{2}$. If $\alpha < \beta$, $\varepsilon^{-1}P_A$ is the unique prime residual language and for any integer $n > 0$, $(a^n)^{-1}P_A$ is composed of $\varepsilon^{-1}P_A$ and $(a^{n+1})^{-1}P_A$. However, it can easily be shown that a stochastic language whose set of prime residual languages is a finite residual net is in $\mathcal{L}_{PRFA}$. Furthermore, if $P \in \mathcal{L}_{PRFA}$ and if $\mathcal{P}$ is the set of its prime residual languages, every residual language of P is in $conv(\mathcal{P})$.

Proposition 9. *Let $A = \langle\Sigma, Q, \varphi, \iota, \tau\rangle$ be a PRFA. Then, for any prime residual language $u^{-1}P_A$ and any $q \in \delta(Q_I, u)$, we have $P_{A,q} = u^{-1}P_A$, where δ is the transition function of the support of A. If A is reduced, then there exists only one state $q \in Q$ such that $P_{A,q} = u^{-1}P_A$. Moreover, any $P_{A,q}$ is a prime residual language of P_A.*

Proof. Let $R = \delta(Q_I, u)$, there exist non negative real numbers $(\alpha_r)_{r\in R}$ such that $u^{-1}P_A = \sum_{r\in R} \alpha_r P_{A,r}$. As $u^{-1}P_A$ is prime and as A is a PRFA, there must exist $r \in R$ such that $u^{-1}P_A = P_{A,r}$. Let $S = \{r \in R | P_{A,r} = u^{-1}P_A\}$, $\overline{S} = R\backslash S$ and let $\alpha = \sum_{s\in S} \alpha_s$. If $\alpha < 1$, we would have $u^{-1}P_A = \sum_{s\in\overline{S}} \frac{\alpha_s}{1-\alpha} P_{A,s}$ which is impossible since each $P_{A,s}$ is a residual language of P_A and $u^{-1}P_A$ is prime. Therefore, $\alpha = 1$ and $S = \delta(Q_I, u)$. If A is reduced, there cannot be two distinct states of A which define the same stochastic language. Finally, as any residual language is composed of prime residual languages, any $P_{A,q}$ is a prime residual language of P_A if A is reduced. □

As a corollary, it can be shown that the support of reduced PRFAs are exactly RFSAs $\langle\Sigma, Q, Q_0, F, \delta\rangle$ such that for every state $q \in Q$, there exists $u \in \Sigma^*$ such that $\delta(Q_0, u) = \{q\}$. So, not all RFSAs can be the support of a PRFA.

Proposition 10. *Let $P \in \mathcal{L}_{PRFA}$, let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be the set of all prime residual languages of P. Let $\alpha^x_{i,j}$ and β_i be non negative real numbers defined for all $1 \le i, j \le k$ and $x \in \Sigma$ such that*

- $x^{-1}P_i = \sum_{j=1}^k \alpha^x_{i,j} P_j$ *with* $P_j \in cov\left(x^{-1}P_i, \mathcal{P}\right) \Rightarrow \alpha^x_{i,j} > 0$
- $P = \sum_{i=1}^k \beta_i P_i$ *with* $P_i \in cov\left(P, \mathcal{P}\right) \Rightarrow \beta_i > 0$.

Let $A = \langle\Sigma, \mathcal{P}, \varphi, \iota, \tau\rangle$ be the PFA such that $\varphi(P_i, x, P_j) = \alpha^x_{i,j} P_i(x\Sigma^*)$, $\iota(P_i) = \beta_i$ and $\tau(P_i) = P_i(\varepsilon)$ for any $1 \le i, j \le k$ and any letter x. Then, A is a reduced saturated PRFA which generates P.

Proof. First, we prove by induction that for any state P_i of A, $P_{A,P_i} = P_i$. We have $P_{A,P_i}(\varepsilon) = P_i(\varepsilon)$. Assume now that for any state P_i and any word w of length $\leq l$, $P_{A,P_i}(w) = P_i(w)$. Let x be a letter, then we have:

$$P_{A,P_i}(xw) = \sum_{j=1}^{k} \varphi(P_i, x, P_j) P_{A,P_j}(w) = \sum_{j=1}^{k} \alpha_{i,j}^{x} P_i(x\Sigma^*) P_j(w)$$

$$= P_i(x\Sigma^*) \sum_{j=1}^{k} \alpha_{i,j}^{x} P_j(w) = P_i(x\Sigma^*)[x^{-1}P_i](w) = P_i(xw).$$

Then for any state P_i, $P_{A,P_i} = P_i$. We have

$$P_A = \sum_{i=1}^{k} \beta_i P_{A,P_i} = \sum_{i=1}^{k} \beta_i P_i = P$$

so A generates P. Therefore, A is a PRFA. It is clear that A is reduced and saturated as every p is a prime residual language. □

Let $can(P)$ be the set of *canonical* PRFAs obtained by the last construction. It is clear that any two elements of $can(P)$ are isomorphic.

Theorem 1. *Let $P \in \mathcal{L}_{PRFA}$. All reduced PRFAs that generate P are state-equivalent. All saturated reduced PRFAs that generate P are canonical PRFAs.*

Proof. From Prop. 9, all reduced PRFAs that generate P are state-equivalent. From Prop. 5 and 10, all saturated reduced PRFAs that generate P are in $can(P)$. □

Previous results have a geometrical interpretation: the (possibly infinite) set of residual languages of a stochastic language generated by a PRFA is contained in a polytope whose vertices are its prime residual languages.

5 Decision and complexity problems.

Deciding whether two NFAs are equivalent is a PSPACE-complete problem but deciding whether two PFAs are equivalent can be done within polynomial time [15]. Given a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$, there exist states $q_1, \ldots, q_k$ s.t. any $P_{A,q}$ can be uniquely written as a linear combination of $P_{A,q_1}, \ldots, P_{A,q_k}$, i.e. $P_{A,q} = \sum_{i=1}^{k} \alpha_q^i P_{A,q_i}$ where the α_q^i need not be non negative. Also, by adapting results from [16] and [15], it can easily be shown that there exists a polynomial algorithm which computes such states q_i and coefficients α_q^i from a given PFA. So, given a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$, $q \in Q$, $x \in \Sigma$ and $R \subseteq Q$, it can be decided within polynomial time whether P_A or $x^{-1}P_{A,q}$ belongs to $conv(\{P_{A,r} \mid r \in R\})$. Moreover, $S_I = \{r \in Q \mid P_{A,r} \in cov(P_A, \mathcal{L}_A)\}$ and $S_{q,x} = \{r \in Q \mid P_{A,r} \in cov(x^{-1}P_{A,q}, \mathcal{L}_A)\}$ can also be computed within polynomial time. Finally, using linear programming techniques, strictly positive coefficients such that

$$x^{-1}P_{A,q} = \sum_{r \in S_{q,x}} \alpha_{q,r}^{x} P_{A,r} \text{ and } P_A = \sum_{r \in S_I} \beta_r P_{A,r}$$

can be found within polynomial time. So,

Theorem 2. *Given a PFA A,*

- *it is decidable in polynomial time whether A is reduced,*
- *a reduction of A can be computed within polynomial time,*
- *it is decidable in polynomial time whether A is saturated,*
- *a saturation of A can be computed within polynomial time.*

Note that these results contrast dramatically with the situation on NFAs. It has been shown in [14] that deciding whether an NFA is saturated or deciding whether it can be reduced are PSPACE-complete problems.

Proposition 11. *It is decidable whether a given reduced PFA is a PRFA.*

Proof. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a reduced PFA. A is a PRFA iff its support is an RFSA $\langle \Sigma, Q, Q_0, F, \delta \rangle$ such that for every state $q \in Q$, there exists some $u \in \Sigma^*$ such that $\delta(Q_0, u) = \{q\}$. This last property can be decided, for example by using the subset construction to determinize A.

Proposition 12. *It is decidable whether a given PFA is equivalent to some PRFA having at most n states.*

Proof. Each state of a PRFA A having n states is uniquely reachable by a word whose length is $\leq 2^n$. So, $P_A \in \mathcal{L}_{PRFA}$ iff for any word $u \in \Sigma^{2^n}$ and any letter x, $(ux)^{-1}P_A \in conv(\{v^{-1}P_A | v \in \Sigma^{\leq 2^n}\})$ and this last property is decidable. □

We do not know whether the following problems are decidable:

- given a PFA A, $P_A \in \mathcal{L}_{PDFA}$?
- given a PFA A, $P_A \in \mathcal{L}_{PRFA}$?
- given a PRFA A, $P_A \in \mathcal{L}_{PDFA}$?

The first problem is decidable when A is non ambiguous, i.e. if each word recognized by the support of A has only one derivation [17, 18]. The second problem has an interesting geometrical formulation as for any $P \in \mathcal{L}_{PFA}$, the set $\{u^{-1}P | u \in \Sigma^*\}$ can be naturally embedded into a vector space of finite dimension: $P_A \in \mathcal{L}_{PRFA}$ iff the polyhedron $conv(\{u^{-1}P_A | u \in \Sigma^{\leq n}\})$ is stationary from some index n.

6 Conclusion

Residual languages are natural components of stochastic languages. This notion proves to be as useful as it is in classical language theory. In particular, it allows to define interesting subclasses of PFA and of regular stochastic languages. The fact that languages generated by PRFA have a unique canonical PRFA representation which can be computed within polynomial time from any equivalent PRFA is promising and should allow to design specific inference algorithms: this is a work in progress. Deciding whether a regular stochastic language can be generated by a PDFA is a classical difficult open problem. Deciding whether such a language can be generated by a PRFA seems to be at least as difficult as the previous problem.

References

1. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–285
2. Jelinek, F.: Statistical Methods for Speech Recognition. The MIT Press, Cambridge, Massachusetts (1997)
3. Freitag, D., McCallum, A.: Information extraction with HMM structures learned by stochastic optimization. In: AAAI/IAAI. (2000) 584–589
4. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. MIT Press (1998)
5. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. CUP (1998)
6. Casacuberta, F.: Some relations among stochastic finite state networks used in automatic speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-12** (1990) 691–695
7. Carrasco, R., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: International Conference on Grammatical Inference, Heidelberg, Springer-Verlag (1994) 139–152
8. Carrasco, R.C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in p olynomial time. RAIRO (Theoretical Informatics and Applications) **33** (1999) 1–20
9. Thollard, F., Dupont, P., de la Higuera, C.: Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In: Proc. 17th International Conf. on Machine Learning, (KAUFM) 975–982
10. Higuera, C.D.L., Thollard, F.: Identification in the limit with probability one of stochastic deterministic finite automata. In: Grammatical Inference: Algorithms and Applications, 5th International Colloquium, ICGI 2000, Lisbon, Portugal, September 11 - 13, 2000 ; Proceedings. Volume 1891 of Lecture Notes in Artificial Intelligence., Springer (2000) 141–156
11. Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using RFSA. In Springer-Verlag, ed.: Proceedings of the 12th International Conference on Algorithmic Learning Theory (ALT-01). Number 2225 in Lecture Notes in Computer Science (2001) 348–359
12. Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using non deterministic finite automata. In: ICGI'2000, 5th International Colloquium on Grammatical Inference. Volume 1891 of LNAI., Springer Verlag (2000) 39–50
13. Esposito, Y., Lemay, A., Denis, F., Dupont, P.: Learning probabilistic residual finite state automata. In: ICGI'2002, 6th International Colloquium on Grammatical Inference. LNAI, Springer Verlag (2002)
14. Denis, F., Lemay, A., Terlutte, A.: Residual Finite State Automata. Fundamenta Informaticae **51** (2002) 339–368
15. Balasubramanian, V.: Equivalence and reduction of hidden markov models. Technical Report AITR-1370, MIT (1993)
16. Paz, A.: Introduction to probabilistic automata. Academic Press, London (1971)
17. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics **23** (1997) 269–311
18. Allauzen, C., Mohri, M.: Efficient algorithms for testing the twins property. Journal of Automata, Languages and Combinatorics (to appear, 2002)