



HAL
open science

Using Pseudo-Stochastic Rational Languages in Probabilistic Grammatical Inference

Amaury Habrard, Francois Denis, Yann Esposito

► **To cite this version:**

Amaury Habrard, Francois Denis, Yann Esposito. Using Pseudo-Stochastic Rational Languages in Probabilistic Grammatical Inference. 2006. hal-00085176v1

HAL Id: hal-00085176

<https://hal.science/hal-00085176v1>

Preprint submitted on 11 Jul 2006 (v1), last revised 7 Nov 2008 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Pseudo-Stochastic Rational Languages in Probabilistic Grammatical Inference

Amaury Habrard, François Denis, and Yann Esposito

Laboratoire d'Informatique Fondamentale de Marseille (L.I.F.) UMR CNRS 6166
 {habrard,fdenis,esposito}@cmi.univ-mrs.fr

Abstract. In probabilistic grammatical inference, a usual goal is to infer a good approximation of an unknown distribution P called a *stochastic language*. The estimate of P stands in some class of probabilistic models such as probabilistic automata (PA). In this paper, we focus on probabilistic models based on multiplicity automata (MA). The stochastic languages generated by MA are called *rational stochastic languages*; they strictly include stochastic languages generated by PA; they also admit a very concise canonical representation. Despite the fact that this class is not recursively enumerable, it is efficiently identifiable in the limit by using the algorithm DEES, introduced by the authors in a previous paper. However, the identification is not proper and before the convergence of the algorithm, DEES can produce MA that do not define stochastic languages. Nevertheless, it is possible to use these MA to define stochastic languages. We show that they belong to a broader class of rational series, that we call *pseudo-stochastic rational languages*. The aim of this paper is twofold. First we provide a theoretical study of pseudo-stochastic rational languages, the languages output by DEES, showing for example that this class is decidable within polynomial time. Second, we have carried out a lot of experiments in order to compare DEES to classical inference algorithms such as ALERGIA and MDI. They show that DEES outperforms them in most cases.

Keywords. pseudo-stochastic rational languages, multiplicity automata, probabilistic grammatical inference.

1 Introduction

In probabilistic grammatical inference, we often consider stochastic languages which define distributions over Σ^* , the set of all the possible words over an alphabet Σ . In general, we consider an unknown distribution P and the goal is to find a good approximation given a finite sample of words independently drawn from P .

The class of probabilistic automata (PA) is often used for modeling such distributions. This class has the same expressiveness as Hidden Markov Models and is identifiable in the limit [4]. However, there exists no efficient algorithm for identifying PA. This can be explained by the fact that there exists no canonical representation of these automata which makes it difficult to correctly identify the structure of the target. One solution is to focus on subclasses of PA such as probabilistic deterministic automata [3,9] but with an important lack of expressiveness. Another solution consists in considering the class of multiplicity automata (MA). These models admit a canonical representation which offers good opportunities from a machine learning point of view. MA define

functions that compute rational series with values in \mathbb{R} [5]. MA are a strict generalization of PA and the stochastic languages generated by PA are special cases of rational stochastic languages. Let us denote by $S_K^{rat}(\Sigma)$ the class of rational stochastic languages computed by MA with parameters in K where $K \in \{\mathbb{Q}, \mathbb{Q}^+, \mathbb{R}, \mathbb{R}^+\}$. With $K = \mathbb{Q}^+$ or $K = \mathbb{R}^+$, $S_K^{rat}(\Sigma)$ is exactly the class of stochastic languages generated by PA with parameters in K . But, when $K = \mathbb{Q}$ or $K = \mathbb{R}$, we obtain strictly greater classes. This provides several advantages: Elements of $S_K^{rat}(\Sigma)$ have a minimal normal representation, thus elements of $S_{\mathbb{Q}^+}^{rat}(\Sigma)$ may have significantly smaller representation in $S_K^{rat}(\Sigma)$; parameters of these minimal representations are directly related to probabilities of some natural events of the form $u\Sigma^*$, which can be efficiently estimated from stochastic samples; lastly when K is a field, rational series over K form a vector space and efficient linear algebra techniques can be used to deal with rational stochastic languages.

However, the class $S_{\mathbb{Q}}^{rat}(\Sigma)$ presents a serious drawback: There exists no recursively enumerable subset class of MA which exactly generates it [4]. As a consequence, no proper identification algorithm can exist: indeed, applying a proper identification algorithm to an enumeration of samples of Σ^* would provide an enumeration of the class of rational stochastic languages over \mathbb{Q} . In spite of this result, there exists an efficient algorithm, DEES, which is able to identify $S_K^{rat}(\Sigma)$ in the limit. But before reaching the target, DEES can produce MA that do not define stochastic languages. However, it has been shown in [6] that with probability one, for any rational stochastic language p , if DEES is given as input a sufficiently large sample S drawn according to p , DEES outputs a rational series such that $\sum_{u \in \Sigma^*} r(u)$ converges absolutely to 1. Moreover, $\sum_{u \in \Sigma^*} |p(u) - r(u)|$ converges to 0 as the size of S increases. We show that these MA belong to a broader class of rational series, that we call *pseudo-stochastic rational languages*. A pseudo-stochastic rational language r has the property that $r(u\Sigma^*) = \lim_{n \rightarrow \infty} r(u\Sigma^{\leq n})$ is defined for any word u and that $r(\Sigma^*) = 1$. A stochastic language p_r can be associated with r in such a way that $\sum_{u \in \Sigma^*} |p_r(u) - r(u)| = 2 \sum_{r(u) < 0} |r(u)|$ when the sum $\sum_{u \in \Sigma^*} r(u)$ is absolutely convergent. As a first consequence, $p_r = r$ when r is a stochastic language. As a second consequence, for any rational stochastic language p , if DEES is given as input increasing samples drawn according to p , DEES outputs pseudo-stochastic rational languages r such that $\sum_{u \in \Sigma^*} |p(u) - p_r(u)|$ converges to 0 as the size of S increases.

The aim of this paper is twofold: To provide a theoretical study of the class of pseudo-stochastic rational languages and a series of experiments in order to compare the performance of DEES to two classical inference algorithms: ALERGIA [3] and MDI [9]. We show that the class of pseudo-stochastic rational languages is decidable within polynomial time. We provide an algorithm that can be used to compute $p_r(u)$ from any MA that computes r . We also show how it is possible to simulate p_r using such an automaton. We show that there exist pseudo-stochastic rational languages r such that p_r is not rational. Finally, we show that it is undecidable whether two pseudo-stochastic rational languages define the same stochastic language. We have carried out a lot of experiments which show that DEES outperforms ALERGIA and MDI in most cases. These results were expected since ALERGIA and MDI have not the same theoretical expressiveness and since DEES aims at producing a minimal representation of the target

in the set of MA, which can be significantly smaller than the smaller equivalent PDA (if it exists).

The paper is organized as follows. In section 2, we introduce some background about multiplicity automata, rational series and stochastic languages and present the algorithm DEES. Section 3 deals with our study of pseudo-rational stochastic languages. Our experiments are detailed in Section 4.

2 Definitions and notations

2.1 Rational series, multiplicity automata and stochastic languages

Let Σ^* be the set of words on the finite alphabet Σ . A language is a subset of Σ^* . The empty word is denoted by ε and the length of a word u is denoted by $|u|$. For any integer k , let $\Sigma^k = \{u \in \Sigma^* : |u| = k\}$ and $\Sigma^{\leq k} = \{u \in \Sigma^* : |u| \leq k\}$. We denote by $<$ the length-lexicographic order on Σ^* and by $MinL$ the minimal element of a non empty language L according to this order. A subset S of Σ^* is *prefix-closed* if for any $u, v \in \Sigma^*$, $uv \in S \Rightarrow u \in S$. For any $S \subseteq \Sigma^*$, let $pref(S) = \{u \in \Sigma^* : \exists v \in \Sigma^*, uv \in S\}$ and $fact(S) = \{v \in \Sigma^* : \exists u, w \in \Sigma^*, uvw \in S\}$.

A *formal power series* is a mapping r of Σ^* into \mathbb{R} . The set of all formal power series is denoted by $\mathbb{R}\langle\langle\Sigma\rangle\rangle$. It is a vector space. For any series r and any word u , let us denote by $\dot{u}r$ the series defined by $\dot{u}r(w) = r(uw)$ for every word w . Let us denote by $supp(r)$ the *support* of r , i.e. the set $\{w \in \Sigma^* : r(w) \neq 0\}$. A *stochastic language* is a formal series p which takes its values in \mathbb{R}^+ and such that $\sum_{w \in \Sigma^*} p(w) = 1$. The set of all stochastic languages over Σ is denoted by $\mathcal{S}(\Sigma)$. For any language $L \subseteq \Sigma^*$ and any $p \in \mathcal{S}(\Sigma)$, let us denote $\sum_{w \in L} p(w)$ by $p(L)$. For any $p \in \mathcal{S}(\Sigma)$ and $u \in \Sigma$ such that $p(u\Sigma^*) \neq 0$, the *residual language* of p wrt u is the stochastic language defined by $u^{-1}p$ by $u^{-1}p(w) = \frac{p(uw)}{p(u\Sigma^*)}$. We denote by $res(p)$ the set $\{u \in \Sigma^* : p(u\Sigma^*) \neq 0\}$ and by $Res(p)$ the set $\{u^{-1}p : u \in res(p)\}$.

Let S be a sample over Σ^* , i.e. a multiset composed of words over Σ^* . We denote by p_S the empirical distribution over Σ^* associated with S . Let S be an infinite sample composed of words independently drawn according to a stochastic language p . We denote by S_n the sequence composed of the n first words of S .

We introduce now the notion of multiplicity automata (MA). Let $K \in \{\mathbb{R}, \mathbb{Q}, \mathbb{R}^+, \mathbb{Q}^+\}$. A K -multiplicity automaton (MA) is a 5-tuple $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ where Q is a finite set of states, $\varphi : Q \times \Sigma \times Q \rightarrow K$ is the transition function, $\iota : Q \rightarrow K$ is the initialization function, $\tau : Q \rightarrow K$ is the termination function. We extend the transition function φ to $Q \times \Sigma^* \times Q$ by $\varphi(q, wx, r) = \sum_{s \in Q} \varphi(q, w, s) \varphi(s, x, r)$ and $\varphi(q, \varepsilon, r) = 1$ if $q = r$ and 0 otherwise, for any $q, r \in Q$, $x \in \Sigma$ and $w \in \Sigma^*$. For any finite subset $L \subset \Sigma^*$ and any $R \subseteq Q$, define $\varphi(q, L, R) = \sum_{w \in L, r \in R} \varphi(q, w, r)$. We denote by $Q_I = \{q \in Q | \iota(q) \neq 0\}$ the set of *initial states* and by $Q_T = \{q \in Q | \tau(q) \neq 0\}$ the set of *terminal states*. A state $q \in Q$ is *accessible* (resp. *co-accessible*) if there exists $q_0 \in Q_I$ (resp. $q_t \in Q_T$) and $u \in \Sigma^*$ such that $\varphi(q_0, u, q) \neq 0$ (resp. $\varphi(q, u, q_t) \neq 0$). An MA is *trimmed* if all its states are accessible and co-accessible. From now, we only consider trimmed MA. The *support* of an MA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ is the *Non-deterministic Finite Automaton (NFA)* $\langle \Sigma, Q, Q_I, Q_T, \delta \rangle$ where $\delta(q, x) = \{q' \in Q | \varphi(q, x, q') \neq 0\}$.

The *spectral radius* of a square matrix M is the maximum magnitude of its eigenvalues. Let $A = \langle \Sigma, Q = \{q_1, \dots, q_n\}, \iota, \varphi, \tau \rangle$ be an MA. Let us denote by $\rho(A)$ be the spectral radius of the square matrix $[\varphi(q_i, \Sigma, q_j)]_{1 \leq i, j \leq n}$ ($\rho(A)$ does not depend on the order of the states). If $\rho(A) < 1$ then each sequence $r_{A,q}(\Sigma^{\leq n})$ converges to a number s_q and hence, $r(\Sigma^{\leq n})$ converges too [6]. Let us denote by $r(\Sigma^*)$ the limit of $r(\Sigma^{\leq n})$ when it exists. The numbers s_q are the unique solutions of the following linear system of equations (and therefore are computable within polynomial time):

$$s_q = r_{A,q} + \sum_{q' \in Q} \varphi(q, \Sigma, q') s_{q'} \text{ for } q \in Q.$$

It is decidable within polynomial time whether $\rho(A) < 1$ [2,7].

A *Probabilistic Automaton (PA)* is a trimmed MA $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ s.t. ι, φ and τ take their values in $[0, 1]$, s.t. $\sum_{q \in Q} \iota(q) = 1$ and for any state q , $\tau(q) + \varphi(q, \Sigma, Q) = 1$. A *Probabilistic Deterministic Automaton (PDA)* is a PA whose support is deterministic. It can be shown that Probabilistic Automata generate stochastic languages. Let us denote by $\mathcal{S}_K^{PA}(\Sigma)$ (resp. $\mathcal{S}_K^{PDA}(\Sigma)$) the class of all stochastic languages which can be computed by a PA (resp. a PDA).

For any MA A , let r_A be the series defined by $r_A(w) = \sum_{q,r \in Q} \iota(q) \varphi(q, w, r) \tau(r)$. For any $q \in Q$, we also define the series $r_{A,q}$ by $r_{A,q}(w) = \sum_{r \in Q} \varphi(q, w, r) \tau(r)$. An MA A is *reduced* if the set $\{r_{A,q} | q \in Q\}$ is linearly independent in the vector space $\mathbb{R}\langle\langle \Sigma \rangle\rangle$. An MA A is *prefix-closed* if (i) its set of states Q is a prefix-closed subset of Σ^* , (ii) $Q_I = \{\varepsilon\}$ and (iii) $\forall u \in Q, \delta(\varepsilon, u) = \{u\}$ where δ is the transition function in the support of A .

Rational series have several characterizations ([1,8]). Here, we shall say that a formal power series over Σ is K -rational iff there exists a K -multiplicity automaton A such that $r = r_A$, where $K \in \{\mathbb{R}, \mathbb{R}^+, \mathbb{Q}, \mathbb{Q}^+\}$. Let us denote by $K^{rat}\langle\langle \Sigma \rangle\rangle$ the set of K -rational series over Σ and by $\mathcal{S}_K^{rat}(\Sigma) = K^{rat}\langle\langle \Sigma \rangle\rangle \cap \mathcal{S}(\Sigma)$, the set of *rational stochastic languages* over K . It can be shown that a series r is \mathbb{R} -rational iff the set $\{ur | u \in \Sigma^*\}$ spans a finite dimensional vector subspace of $\mathbb{R}\langle\langle \Sigma \rangle\rangle$. As a corollary, a stochastic language p is \mathbb{R} -rational iff the set $Res(p)$ spans a finite dimensional vector subspace $[Res(p)]$ of $\mathbb{R}\langle\langle \Sigma \rangle\rangle$. Rational stochastic languages have been studied in [5] from a language theoretical point of view. It is worth noting that $\mathcal{S}_{\mathbb{R}}^{PDA}(\Sigma) \subsetneq \mathcal{S}_{\mathbb{R}}^{PA}(\Sigma) = \mathcal{S}_{\mathbb{R}^+}^{rat}(\Sigma) \subsetneq \mathcal{S}_{\mathbb{R}}^{rat}(\Sigma)$. From now on, a rational stochastic language will always denote an \mathbb{R} -rational stochastic language.

Rational stochastic languages have a serious drawback. There exists no recursively enumerable subset of multiplicity automata capable to generate them [4,5]. As a consequence, it is undecidable whether a given MA computes a stochastic language.

Every rational language is the support of a rational series but the converse is false: there exists rational series whose supports are not rational. For example, it can be shown that the complementary set of $\{a^n b^n | n \in \mathbb{N}\}$ in $\{a, b\}^*$ is the support of a rational series. However, a variant of Pumping Lemma holds for languages which are support of rational series. Let L be such a language. There exists an integer N such that for any word $w = uv \in L$ satisfying $|v| \geq N$, there exists v_1, v_2, v_3 such that $v = v_1 v_2 v_3$ and $L \cap uv_1 v_2^* v_3$ is infinite [1].

Rational stochastic languages admit a canonical representation by reduced prefix-closed MA. Let p be a rational stochastic language and let Q_p be the smallest basis of $[Res(p)]$ (for the order induced by $<$ on the finite subsets of Σ^*). Let $A =$

$\langle \Sigma, Q_p, \varphi, \iota, \tau \rangle$ be the MA defined by: (i) $\iota(\varepsilon) = 1$, $\iota(u) = 0$ otherwise; $\tau(u) = u^{-1}p(\varepsilon)$, (ii) $\varphi(u, x, ux) = u^{-1}p(x\Sigma^*)$ if $u, ux \in Q_p$ and $x \in \Sigma$, (iii) $\varphi(u, x, v) = \alpha_v u^{-1}p(x\Sigma^*)$ if $x \in \Sigma$, $ux \in (Q_p\Sigma \setminus Q_p) \cap \text{res}(p)$ and $(ux)^{-1}p = \sum_{v \in Q_p} \alpha_v v^{-1}p$. It can be shown that A is a reduced prefix-closed MA which computes p and such that $\rho(A) < 1$. A is called the *canonical representation* of p . Note that the parameters of A correspond to natural components of the residual of p and can be estimated by using samples of p .

2.2 Inference of rational stochastic languages

The algorithm DEES [6] is able to identify rational stochastic languages: with probability one, for every rational stochastic language p and every infinite sample S of p , there exists an integer N such that for every $n \geq N$, $\text{DEES}(S_n)$ outputs the canonical representation A of p . Before its presentation, we introduce informally the basic idea of the algorithm. First, the goal is to find the structure of the automaton, *i.e.* the set of states Q_p smallest basis of $[\text{Res}(P)]$. The inference proceeds as follows: the algorithm begins by building a unique state which corresponds to the residual $\varepsilon^{-1}p_S$. Each state of the automaton corresponds to some residual $u^{-1}p_s$ where u is the prefix of some examples in S . After having built a state corresponding to $u^{-1}p_s$, for any letter x , the algorithm studies the possibility of adding a new state corresponding to $(ux)^{-1}p_s$ or of creating transitions labeled by x that lead to the states already built in the automaton. A new state will be added to the automaton if the residual language corresponding to $(ux)^{-1}p_s$ cannot be approximated as a linear combination of the residual languages corresponding to the states already built.

The pseudo-code of the algorithm is presented in Algorithm 1. In order to find a linear combination, DEES uses the following set of inequalities where S is a non empty finite sample of Σ^* , Q a prefix-closed subset of $\text{pref}(S)$, $v \in \text{pref}(S) \setminus Q$, and $\epsilon > 0$:

$$I(Q, v, S, \epsilon) = \{ |v^{-1}P_S(w\Sigma^*) - \sum_{u \in Q} X_u u^{-1}P_S(w\Sigma^*)| \leq \epsilon \mid w \in \text{fact}(S) \} \cup \{ \sum_{u \in Q} X_u = 1 \}.$$

DEES runs in polynomial time in the size of S and identifies in the limit the structure of the canonical representation A of the target p . Once the correct structure of A is found, the algorithm computes estimates α_S of each parameter α of A such that $|\alpha - \alpha_S| = O(|S|^{-1/3})$. The output automaton A computes a rational series r_A such that $\sum_{w \in \Sigma^*} r_A(w)$ converges absolutely to 1. Moreover, it can be shown that r_A converges to the target p under the D_1 distance (also called the L_1 norm), stronger than distance D_2 or D_∞ : $\sum_{w \in \Sigma^*} |r_A(w) - p(w)|$ tends to 0 when the size of S tends to ∞ . If the parameters of A are rational numbers, a variant of DEES can identify exactly the target [6].

We give now a simple example that illustrates DEES. Let us consider a sample $S = \{\varepsilon, a, aa, aaa\}$ such that $|\varepsilon| = 10$, $|a| = |aa| = 20$, $|aaa| = 10$. We have the following values for the empirical distribution: $P_S(\varepsilon) = P_S(aaa) = P_S(aaa\Sigma^*) = \frac{1}{6}$, $P_S(a) = P_S(aa) = \frac{1}{3}$, $P_S(a\Sigma^*) = \frac{5}{6}$, $P_S(aa\Sigma^*) = \frac{1}{2}$ and $P_S(aaaa\Sigma^*) = 0$, $\varepsilon = \frac{1}{(60)^{\frac{1}{3}}} \equiv 0.255$. With the sample S , DEES will infer a multiplicity automaton in three steps:

1. We begin by constructing a state for ε (Figure 1(a)).

Input: a sample S

Output: a prefix-closed reduced MA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$

$Q \leftarrow \{\varepsilon\}; \quad \iota(\varepsilon) \leftarrow 1; \quad \tau(\varepsilon) \leftarrow P_S(\varepsilon);$

$F \leftarrow \Sigma \cap \text{pref}(S)$ /* F is the frontier set*/;

while $F \neq \emptyset$ **do**

$v \leftarrow \text{Min}F$ s.t. $v = u.x$ where $u \in \Sigma^*$ and $x \in \Sigma$;

$F \leftarrow F \setminus \{v\}$;

if $I(Q, v, S, |\Sigma|^{-1/3})$ has no solution **then**

$Q \leftarrow Q \cup \{v\}; \quad \iota(v) \leftarrow 0; \quad \tau(v) \leftarrow P_S(v)/P_S(v\Sigma^*);$

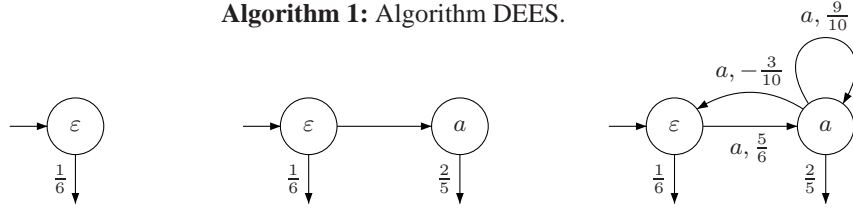
$\varphi(u, x, v) \leftarrow P_S(v\Sigma^*)/P_S(u\Sigma^*); \quad F \leftarrow F \cup \{vx \in \text{res}(P_S) \mid x \in \Sigma\}$;

else

 let $(\alpha_w)_{w \in Q}$ be a solution of $I(Q, v, S, |\Sigma|^{-1/3})$;

foreach $w \in Q$ **do** $\varphi(u, x, w) \leftarrow \alpha_w P_S(v\Sigma^*)/P_S(u\Sigma^*);$

Algorithm 1: Algorithm DEES.



(a) Initialisation with ε .

(b) Creation of a new state.

(c) Final automaton.

Fig. 1. Illustration of the different steps of algorithm DEES.

- We examine $P_S(v\Sigma^*)$ with $v = \varepsilon a$ to decide if we need to add a new state for the string a . We obtain the following system which has in fact no solution and we create a new state as shown in Figure 1(b).

$$\left\{ \begin{array}{l} \left| \frac{P_S(va\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(a\Sigma^*)}{P_S(\Sigma^*)} * X_\varepsilon \right| \leq b, \quad \left| \frac{P_S(vaa\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(aa\Sigma^*)}{P_S(\Sigma^*)} * X_\varepsilon \right| \leq b, \\ \left| \frac{P_S(vaaa\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(aaa\Sigma^*)}{P_S(\Sigma^*)} * X_\varepsilon \right| \leq b, \quad X_\varepsilon = 1 \end{array} \right\}$$

- We examine $P_S(v\Sigma^*)$ with $v = aa$ to decide if we need to create a new state for the string aa . We obtain the system below. It is easy to see that this system admits at least one solution $X_\varepsilon = -\frac{1}{2}$ and $X_a = \frac{3}{2}$. Then, we add two transitions to the automaton and we obtain the automaton of Figure 1(c) and the algorithm halts.

$$\left\{ \begin{array}{l} \left| \frac{P_S(va\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(a\Sigma^*)}{P_S(\Sigma^*)} X_\varepsilon - \frac{P_S(aa\Sigma^*)}{P_S(a\Sigma^*)} X_a \right| \leq b, \quad \left| \frac{P_S(vaa\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(aa\Sigma^*)}{P_S(\Sigma^*)} X_\varepsilon - \frac{P_S(aaa\Sigma^*)}{P_S(a\Sigma^*)} X_a \right| \leq b, \\ \left| \frac{P_S(vaaa\Sigma^*)}{P_S(v\Sigma^*)} - \frac{P_S(aaa\Sigma^*)}{P_S(\Sigma^*)} X_\varepsilon - \frac{P_S(aaaa\Sigma^*)}{P_S(a\Sigma^*)} X_a \right| \leq b, \quad X_\varepsilon + X_a = 1 \end{array} \right\}$$

Since no recursively enumerable subset of MA is capable to generate the set of rational stochastic languages, no identification algorithm can be proper. This remark applies to DEES. There is no guarantee at any step that the automaton A output by DEES computes a stochastic language. However, the rational series r computed by the MA output by DEES can be used to compute a stochastic language p_r that also

converges to the target [6]. Moreover, they have several nice properties which make them close to stochastic languages: We call them pseudo-stochastic rational languages and we study their properties in the next Section.

3 Pseudo-stochastic rational languages

The canonical representation A of a rational stochastic language satisfies $\rho(A) < 1$ and $\sum_{w \in \Sigma^*} r_A(w) = 1$. We use this characteristic to define the notion of pseudo-stochastic rational language.

Definition 1. We say that a rational series r is a pseudo-stochastic language if there exists an MA A which computes r and such that $\rho(A) < 1$ and if $r(\Sigma^*) = 1$.

Note that the condition $\rho(A) < 1$ implies that $r(\Sigma^*)$ is defined without ambiguity. A rational stochastic language is a pseudo-stochastic rational language but the converse is false.

Example. Let $A = \langle \Sigma, \{q_0\}, \varphi, \iota, \tau \rangle$ defined by $\Sigma = \{a, b\}$, $\iota(q_0) = \tau(q_0) = 1$, $\varphi(q_0, a, q_0) = 1$ and $\varphi(q_0, b, q_0) = -1$. We have $r_A(u) = (-1)^{|u|_b}$. Check that $\rho(A) = 0$ and $r_A(u\Sigma^*) = (-1)^{|u|_b}$ for every word u . Hence, r_A is a pseudo stochastic language.

As indicated in the previous section, any canonical representation A of a rational stochastic language satisfies $\rho(A) < 1$. In fact, the next Lemma shows that any reduced representation A of a pseudo-stochastic language satisfies $\rho(A) < 1$.

Lemma 1. Let A be a reduced representation of a pseudo-stochastic language. Then, $\rho(A) < 1$.

Proof. The proof is detailed in Annex 6.1.

Proposition 1. It is decidable within polynomial time whether a given MA computes a pseudo-stochastic language.

Proof. Given an MA B , compute a reduced representation A of B , check whether $\rho(A) < 1$ and then, compute $r_A(\Sigma^*)$. \square

It has been shown in [6] that a stochastic language p_r can be associated with a pseudo-stochastic rational language r : the idea is to prune in Σ^* all subsets $u\Sigma^*$ such that $r(u\Sigma^*) \leq 0$ and to normalize in order to obtain a stochastic language. Let N be the smallest prefix-closed subset of Σ^* satisfying

$$\varepsilon \in N \text{ and } \forall u \in N, x \in \Sigma, ux \in N \text{ iff } r(ux\Sigma^*) > 0.$$

For every $u \in \Sigma^* \setminus N$, define $p_r(u) = 0$. For every $u \in N$, let $\lambda_u = \text{Max}(r(u), 0) + \sum_{x \in \Sigma} \text{Max}(r(ux\Sigma^*), 0)$. Then, define $p_r(u) = \text{Max}(r(u), 0) / \lambda_u$. It can be shown (see [6]) that $r(u) \leq 0 \Rightarrow p_r(u) = 0$ and $r(u) \geq 0 \Rightarrow r(u) \geq p_r(u)$.

The difference between r and p_r is simple to express when the sum $\sum_{u \in \Sigma^*} r(u)$ converges absolutely. Let $N_r = \sum_{r(u) \leq 0} |r(u)|$. We have $\sum_{w \in \Sigma^*} |r(w) - p_r(w)| = N_r + \sum_{r(u) > 0} (r(u) - p_r(u)) = 2N_r + \sum_{u \in \Sigma^*} (r(u) - p_r(u)) = 2N_r$. Note that


```

Input: MA  $A = \langle \Sigma, Q = \{q_1, \dots, q_n\}, \varphi, \iota, \tau \rangle$  s.t.  $\rho(A) < 1$  and  $r_A(\Sigma^*) = 1$ 
      a word  $u$ 
Output:  $p_{r_A}(u), p_{r_A}(u\Sigma^*)$ 
for  $i = 1, \dots, n$  /* this step is polynomial in  $n$  and is done once*/ do
   $s_i \leftarrow r_{A, q_i}(\Sigma^*); \quad e_i \leftarrow \iota(q_i);$ 
 $w \leftarrow \varepsilon; \quad \lambda \leftarrow 1$  /*  $\lambda$  is equal to  $p_{r_A}(w\Sigma^*)$  */;
repeat
   $\mu \leftarrow \sum_{i=1}^n e_i \tau(q_i); \quad S \leftarrow \{(w, \text{Max}(\mu, 0))\};$ 
  for  $x \in \Sigma$  do
     $\mu \leftarrow \sum_{i,j=1}^n e_i \varphi(q_i, x, q_j) s_j; \quad S \leftarrow S \cup \{(wx, \text{Max}(\mu, 0))\};$ 
   $\sigma \leftarrow \sum_{(v, \mu) \in S} \mu; \quad S \leftarrow \{(x, \mu/\sigma) \mid (x, \mu) \in S\}$  /*normalization*/;
  if  $w = u$  then  $p_{r_A}(u) \leftarrow \lambda \mu$  /*where  $(u, \mu) \in S$  and  $\lambda = p_{r_A}(u\Sigma^*)$  */;
  else
    Let  $x \in \Sigma$  s.t.  $wx$  is a prefix of  $u$  and let  $\mu$  s.t.  $(wx, \mu) \in S$ ;
     $w \leftarrow wx; \quad \lambda \leftarrow \lambda \mu; \quad \mathbf{for} \ i = 1, \dots, n \ \mathbf{do} \ e_i \leftarrow \sum_{j=1}^n e_j \varphi(q_j, x, q_i);$ 
  end
until  $w = u$ ;

```

Algorithm 2: Algorithm computing p_r .

when r is a stochastic language, $\sum_{u \in \Sigma^*} r(u)$ converges absolutely and $N_r = 0$. As a consequence, in that case, $p_r = r$. We give in Algorithm 2 an algorithm that computes $p_r(u)$ and $p_r(u\Sigma^*)$ for any word u from any MA that computes r . This algorithm is linear in the length of the input. It can be slightly modified to generate a word drawn according to p_r (see Annex 6.3).

$a, \rho\alpha; b, \rho$ $a, \rho; b, \rho\beta$ The stochastic languages p_r associated with pseudo-stochastic rational languages r can be not rational.

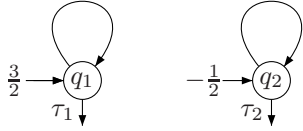


Fig. 2. An example of pseudo-stochastic rational languages r and p_r . *Proof.* Suppose that the parameters of the automaton described on Figure 2 satisfy $\rho(\alpha + 1) + \tau_1 = 1$ and $\rho(\beta + 1) + \tau_2 = 1$ with $\alpha > \beta > 1$. Then the series r_{q_1} and r_{q_2} are rational stochastic languages and therefore, $r_A = 3r_{q_1}/2 - r_{q_2}/2$ is a rational series which satisfies $\sum_{u \in \Sigma^*} |r_A(u)| \leq 2$ and $\sum_{u \in \Sigma^*} r_A(u) = 1$.

Let us show that p_{r_A} is not rational. For any $u \in \Sigma^*$, $r_A(u) = \frac{\rho^{|u|}}{2} (3\alpha^{|u|} \tau_1 - \beta^{|u|} \tau_2)$. For any integer n , there exists an integer m_n such that for any integer i , $r_A(a^n b^i) > 0$ iff $i \leq m_n$. Moreover, it is clear that m_n tends to infinity with n . Suppose now that p_{r_A} is rational and let L be its support. From the Pumping Lemma, there exists an integer N such that for any word $w = uv \in L$ satisfying $|v| \geq N$, there exists v_1, v_2, v_3 such that $v = v_1 v_2 v_3$ and $L \cap uv_1 v_2^* v_3$ is infinite. Let n be such that $m_n \geq N$ and let $u = a^n$ and $v = b^{m_n}$. Since $w = uv \in L$, $L \cap a^n b^*$ should be infinite, which is false. Therefore, L is not the support of a rational language. \square

Different rational series may yield the same pseudo-rational stochastic language. Is it decidable whether two pseudo-stochastic rational series define the same stochastic language? Unfortunately, the answer is no. The proof relies on the following result: it is undecidable whether a multiplicity automaton A over Σ satisfies $r_A(u) \leq 0$ for every $u \in \Sigma^*$ [8]. It is easy to show that this result still holds for the set of MA A which satisfy $|r_A(u)| \leq \lambda^{|u|}$, for any $\lambda > 0$.

Proposition 3. *It is undecidable whether two rational series define the same stochastic language.*

Proof. The proof is detailed in Annex 6.2.

4 Experiments

In this section, we present a set of experiments allowing us to study the performance of the algorithm DEES for learning good stochastic language models. Hence, we will study the behavior of DEES with samples of distributions generated from PDA, PA and non rational stochastic language. We decide to compare DEES to the most well known probabilistic grammatical inference approaches: The algorithms *Alergia* [3] and *MDI* [9] that are able to identify PDAs. These algorithms can be tuned by a parameter, in the experiments we choose the best parameter which gives the best result on all the samples, but we didn't change the parameter according to the size of the sample in order to take into account the impact of the sample sizes.

In our experiments, we use two performance criteria. We measure the size of the inferred models by the number of states. Moreover, to evaluate the quality of the automata, we use the $D1$ norm¹ between two models A and A' defined by :

$$D1(A, A') = \sum_{u \in \Sigma^*} |P_A(u) - P_{A'}(u)|.$$

$D1$ norm is the strongest distance after Kullback Leibler. In practice, we use an approximation by considering a subset of Σ^* generated by A (A will be the target for us).

We carried out a first series of experiment where the target automaton can be represented by a PDA. We consider a stochastic language defined by the automaton on Figure 3. This stochastic language can be represented by a multiplicity automaton of three states and by an equivalent minimal PDA of twelve states [6] (*Alergia* and *MDI* can then identify this automaton). To compare the performances of the three algorithms, we used the following experimental set up. From the target automaton, we generate samples from size 100 to 10000. Then, for each sample we learn an automaton with the three algorithms and compute the norm $D1$ between them and the target. We repeat this experimental setup 10 times and give the average results. Figure 4 reports the results obtained. If we consider the size of the learned models, DEES finds quickly the target automaton, while *MDI* only begins to tend to the target PDA after 10000 examples. The automata produced by *Alergia* are far from this target. This behavior can be explained by the fact that these two algorithms need significantly longer examples to find

¹ Note that we can't use the Kullback-Leibler measure because it is not robust with null probability strings which implies to smooth the learned models, and also because automata produced by DEES do not always define stochastic language, *i.e.* some strings may have a negative value.

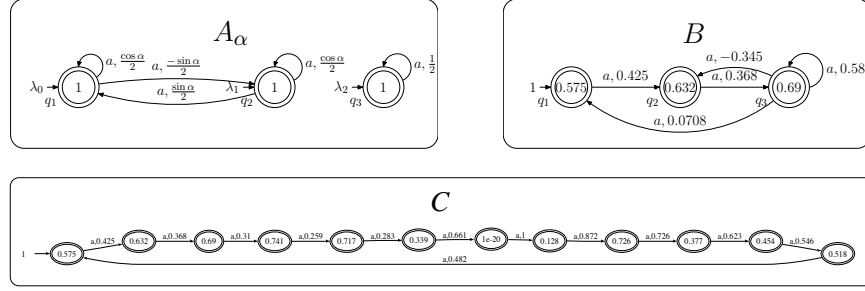
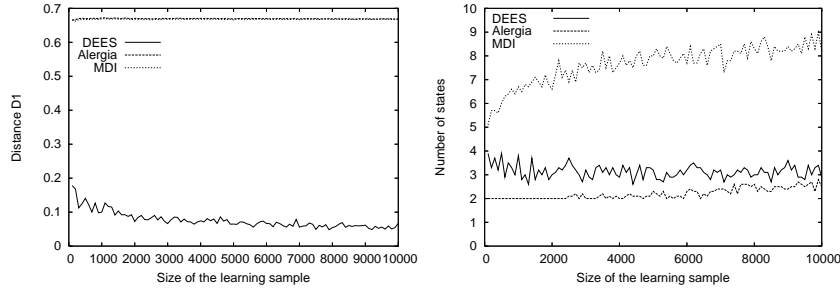


Fig. 3. A_α define stochastic language which can be represented by a PA with at least $2n$ states when $\alpha = \frac{\pi}{n}$. With $\lambda_0 = \lambda_2 = 1$ and $\lambda_1 = 0$, the MA $A_{\pi/6}$ defines a stochastic language P whose prefixed reduced representation is the MA B (with approximate values on transitions). In fact, P can be computed by a PDA and the smallest PA computing it is C .

(a) Results with distance $D1$

(b) Size of the model.

Fig. 4. Results obtained with the prefix reduced multiplicity automaton of three states of Figure 3 admitting a representation with a PDA of twelve states.

the correct target and thus larger samples, this is also amplified because there are more parameters to estimate. In practise we noticed that the correct structure can be found after more than 100000 examples. If we look at the distance $D1$, DEES outperforms MDI and Alergia (which have the same behavior) and begins to converge after 500 examples.

We carried out other series of experiments for evaluating DEES when the target belongs to the class of PA. First, we consider the simple automaton of Figure 5 which defines a stochastic language that can be represented by a PA with parameters in \mathbb{R}^+ . We follow the same experimental setup as in the first experiment, the results are reported on Figure 6. According to our 2 performance criteria, DEES outperforms again Alergia and MDI. In fact, the target can not be modeled correctly by Alergia and MDI because it can not be represented by a PDA. This explains why these algorithms can't find a good model. For them, the best answer is to produce a unigram model. Alergia even diverge at a given step (this behavior is due to its fusion criterion that becomes more restrictive with the increasing of the learning set) and MDI returns always the unigram. DEES finds the correct structure quickly and begins to converge after 1000 examples. This behavior confirms the fact DEES can produce better models with small samples because it constructs small representations. On the other hand, Alergia and MDI seem

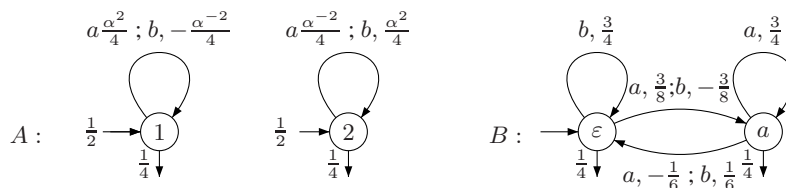


Fig. 5. Automaton A is a PA with non rational parameters in \mathbb{R}^+ ($\alpha = (\sqrt{5} + 1)/2$). A can be represented by an MA B with rational parameters in \mathbb{Q} [5].

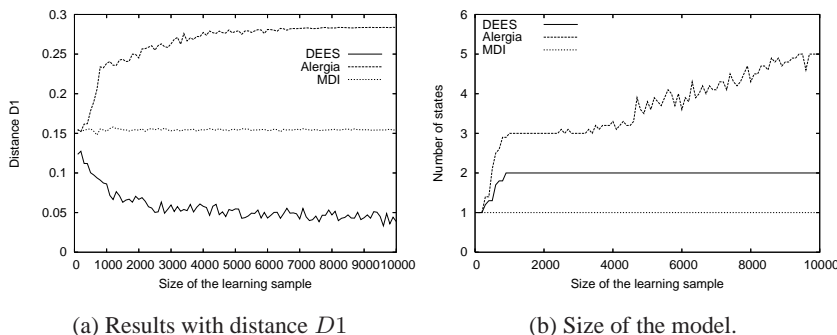
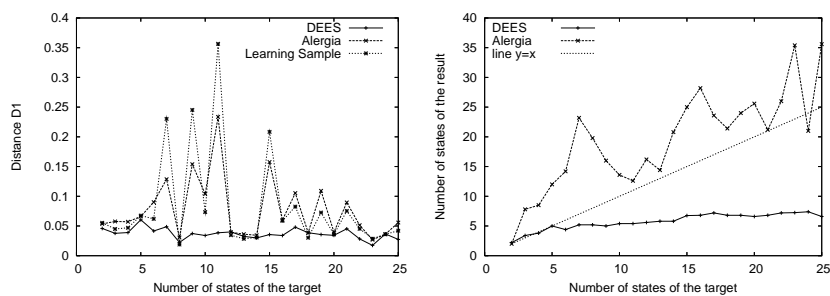


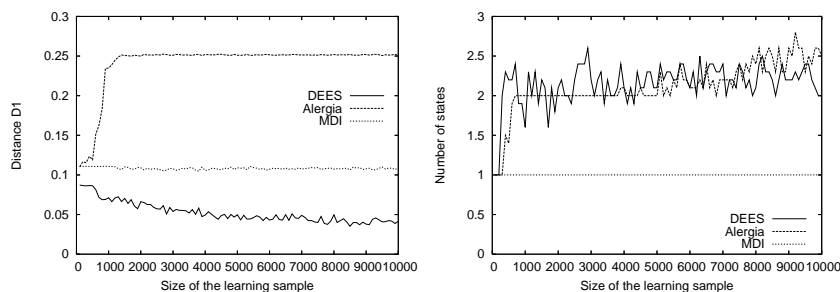
Fig. 6. Results obtained with the target automaton of Figure 5 admitting a representation in the class PA with non rational parameters.

to need a huge number of examples to find a good approximation of the target, even when the target is relatively small.

We made another experiment in the class of PA. We study the behavior of DEES when the learning samples are generated from different targets randomly generated. For this experiment, we take an alphabet of three letters and we generate randomly some PA with a number of states from 2 to 25. The PA are generated in order to have a prefix representation which guarantees that all the states are reachable. The rest of the transitions and the values of the parameters are chosen randomly. Then, for each target, we generate 5 samples of size 300 times the number of states of the target. We made this choice because we think that for small targets the samples may be sufficient to find a good approximation, while for bigger targets there is a clear lack of examples. This last point allows us to see the behaviors of the algorithms with small amounts of data. We learn an automaton from each sample and compare it to the corresponding target. Note that we didn't use MDI in this experiment because this algorithm is extremely hard to tune, which implies an important cost in time for finding a good parameter. The parameter of Alergia is fixed to a reasonable value kept for all the experiment. Results for Alergia and DEES are reported on Figure 7. We also add the empirical distance of the samples to the target automaton. If you consider the $D1$ norm, the performances of Alergia depend highly on the empirical distribution. Alergia infers models close, or better, than those produced by DEES only when the empirical distribution is already very good, thus when it is not necessary to learn. Moreover, Alergia has a greater vari-

(a) Results with distance $D1$

(b) Size of the model.

Fig. 7. Results obtained from a set of PA generated randomly.(a) Results with distance $D1$

(b) Size of the model.

Fig. 8. Results obtained with samples generated from a non rational stochastic language.

ance which implies a weak robustness. On the other hand, DEES is always able to learn significantly small models almost always better, even with small samples.

Finally, we carried out a last experiment where the objective is to study the behavior of the three algorithms with samples generated from a non rational stochastic language. We consider, as a target, the stochastic language generated using the p_r algorithm from the automaton of Figure 2 (note that this automaton admits a prefix reduced representation of 2 states). We took $\rho = 3/10$, $\alpha = 3/2$ and $\beta = 5/4$. We follow the same experimental setup than the first experiment. Since we use rational representations, we measure the distance $D1$ from the automaton of Figure 2 using a sample generated by p_r (*i.e.* we measure the $D1$ only for strings with a strictly positive value). The results are presented on Figure 8. MDI and Alergia are clearly not able to build a good estimation of the target distribution and we see that their best answer is to produce a unigram. On the other hand, DEES is able to identify a structure close to the MA that was used for defining the distribution and produces good automata after 2000 examples. This means that DEES seems able to produce pseudo-stochastic rational languages which are closed to a non rational stochastic distribution.

5 Conclusion

In this paper, we studied the class of pseudo-stochastic rational languages (PSRL) that are stochastic languages defined by multiplicity automata which do not define stochastic languages but share some properties with them. We showed that it is possible to decide whether an MA defines a PSRL, but we can't decide whether two MA define the same PSRL. Moreover, it is possible to define a stochastic language from these MA but this language is not rational in general. Despite of these drawbacks, we showed experimentally that DEES produces MA computing pseudo-stochastic rational languages that provide good estimates of a target stochastic language. We recall here that DEES is able to output automata with a minimal number of parameters which is clearly an advantage from a machine learning standpoint, especially for dealing with small datasets. Moreover, our experiments showed that DEES outperforms standard probabilistic grammatical inference approaches. Thus, we think that the class of pseudo-stochastic rational languages is promising for many applications in grammatical inference. Beyond the fact to continue the study of this class, we also plan to consider methods that could infer a class of MA strictly greater than the class of PSRL. We also began to work on an adaptation of the approaches presented in this paper to trees.

References

1. J. Berstel and C. Reutenauer. *Les séries rationnelles et leurs langages*. Masson, 1984.
2. V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, September 2000.
3. R.C. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *Proceedings of ICGI'94*, LNAI, pages 139–150. Springer, 1994.
4. F. Denis and Y. Esposito. Learning classes of probabilistic automata. In *Proceedings COLT'04*, volume 3120 of *LNCS*, pages 124–139. Springer, 2004.
5. F. Denis and Y. Esposito. Rational stochastic language. Technical report, LIF - Université de Provence, 2006.
6. F. Denis, Y. Esposito, and A. Habrard. Learning rational stochastic languages. In *Proceedings of COLT'06*, 2006.
7. F. R. Gantmacher. *Théorie des matrices, tomes 1 et 2*. Dunod, 1966.
8. A. Salomaa and M. Soittola. *Automata: Theoretic Aspects of Formal Power Series*. Springer-Verlag, 1978.
9. F. Thollard, P. Dupont, and C. de la Higuera. Probabilistic dfa inference using kullback–leibler divergence and minimality. In *Proceedings of ICML'00*, pages 975–982, June 2000.

6 Annex

6.1 Proof of Lemma 1

Lemma 1. *Let A be a reduced representation of a pseudo-stochastic language. Then, $\rho(A) < 1$.*

Proof (sketch). Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a reduced representation of r and let $B = \langle \Sigma, Q_B, \varphi_B, \iota_B, \tau_B \rangle$ be an MA that computes r and such that $\rho(B) < 1$. Since A is reduced, the vector subspace E of $\mathbb{R}\langle\langle \Sigma \rangle\rangle$ spanned by $\{r_{A,q} | q \in Q_A\}$ is equal to $[\{\dot{u}r | u \in \Sigma^*\}]$ and is contained in the vector subspace F spanned by $\{r_{B,q} | q \in Q_B\}$.

The set $\{r_{A,q} | q \in Q_A\}$ is a basis of E . Let us complete it into a basis of F and let P_E be the corresponding projection defined from F over E . Note that for any $x \in \Sigma$ and any $r \in F$, we have $P_E(\dot{x}r) = \dot{x}P_E(r)$.

For any state $q \in Q_B$, let us express $P_E(r_{B,q})$ in this basis.

$$P_E(r_{B,q}) = \sum_{q' \in Q_A} \lambda_{q,q'} r_{A,q'}.$$

Note that for any MA C and any state q of C ,

$$\sum_{x \in \Sigma} \dot{x}r_{C,q} = \sum_{q' \in Q_C} \varphi_C(q, \Sigma, q') r_{C,q'}.$$

Therefore, for any state q of B , we have

$$P_E\left(\sum_{x \in \Sigma} \dot{x}r_{B,q}\right) = P_E\left(\sum_{q' \in Q_B} \varphi_B(q, \Sigma, q') r_{B,q'}\right) = \sum_{q' \in Q_B} \varphi_B(q, \Sigma, q') \sum_{q'' \in Q_A} \lambda_{q',q''} r_{A,q''}$$

but also

$$\begin{aligned} P_E\left(\sum_{x \in \Sigma} \dot{x}r_{B,q}\right) &= \sum_{x \in \Sigma} \dot{x}P_E(r_{B,q}) = \sum_{x \in \Sigma} \dot{x} \sum_{q' \in Q_A} \lambda_{q,q'} r_{A,q'} \\ &= \sum_{q' \in Q_A} \lambda_{q,q'} \sum_{q'' \in Q_A} \varphi_A(q', \Sigma, q'') r_{A,q''} \end{aligned}$$

and therefore

$$\sum_{q' \in Q_B} \sum_{q'' \in Q_A} \varphi_B(q, \Sigma, q') \lambda_{q',q''} = \sum_{q' \in Q_A} \sum_{q'' \in Q_A} \lambda_{q,q'} \varphi_A(q', \Sigma, q'').$$

Now, let M_A (resp. M_B , resp. Λ) be the matrix indexed by $Q_A \times Q_A$ (resp. $Q_B \times Q_B$, resp. $Q_B \times Q_A$) and defined by $M_A[q, q'] = \varphi_A(q, \Sigma, q')$ (resp. $M_B[q, q'] = \varphi_B(q, \Sigma, q')$, resp. $\Lambda[q, q'] = \lambda_{q,q'}$). Note that the rank of Λ is equal to the dimension of E . We have

$$M_B \Lambda = \Lambda M_A.$$

Let μ be an eigenvalue of M_A and let X an associated eigenvector. We have

$$M_B \Lambda X = \Lambda M_A X = \mu \Lambda X$$

and since the rank of Λ is maximal, μ is also an eigenvalue of M_B . Therefore, $\rho(B) < 1$ implies that $\rho(A) < 1$. \square

6.2 Proof of Proposition 3

Proposition 3. *It is undecidable whether two rational series define the same stochastic language.*

Proof. Let $A = \langle \Sigma, Q, \iota, \varphi, \tau \rangle$ be an MA which satisfies $|r_A(u)| \leq \lambda^{|u|}$ for some $\lambda < 1/(2|\Sigma|)$. Let $\bar{\Sigma} = \{\bar{x} | x \in \Sigma\}$ be a disjoint copy of Σ and let c be a new letter: $c \notin \Sigma \cup \bar{\Sigma}$. Let $u \rightarrow \bar{u}$ be the morphism inductively defined from Σ^* into $\bar{\Sigma}^*$ by $\bar{\epsilon} = \epsilon$ and $\bar{u}\bar{x} = \bar{u} \cdot \bar{x}$.

Let $B = \langle \Sigma_B, Q, \iota, \varphi_B, \tau \rangle$ defined by $\Sigma_B = \Sigma \cup \bar{\Sigma} \cup \{c\}$, $\varphi_B(q, c, q') = 1$ if $q = q'$ and 0 otherwise, $\varphi_B(q, x, q') = \varphi_B(q, \bar{x}, q') = \varphi(q, x, q')$ if $x \in \Sigma$.

Let f be the rational series defined by $f(w) = r_A(uv)$ if $w = \bar{u}cv$ for some $u, v \in \Sigma^*$ and 0 otherwise.

Let ρ be such that $2\lambda < \rho < 1/|\Sigma|$, let r be the rational series defined on Σ_B by $r(w) = \rho^{|w|}$ if $w \in \bar{\Sigma}^*$ and 0 otherwise. Let $g = f + r$. Check that

$$\sum_{w \in \bar{\Sigma}^*} \rho^{|w|} = \sum_{n \geq 0} (|\Sigma|\rho)^n = \frac{1}{1 - |\Sigma|\rho} \text{ and}$$

$$\sum_{u, v \in \Sigma^*} |f(\bar{u}cv)| = \sum_{u, v \in \Sigma^*} |r_A(uv)| \leq \sum_{u, v \in \Sigma^*} \lambda^{|uv|} = \left(\sum_{n \geq 0} (|\Sigma|\lambda)^n \right)^2 = \left(\frac{1}{1 - |\Sigma|\lambda} \right)^2.$$

Therefore, the sum $\sum_{w \in \Sigma_B^*} g(w)$ is absolutely convergent. Check also that

$$\sum_{w \in \Sigma_B^*} g(w) \geq \sum_{w \in \Sigma^*} \rho^{|w|} - \sum_{u, v \in \Sigma^*} |f(uv)| \geq \frac{1}{1 - |\Sigma|\rho} - \left(\frac{1}{1 - |\Sigma|\lambda} \right)^2 = \frac{|\Sigma|(|\Sigma|\lambda^2 - 2\lambda + \rho)}{(1 - |\Sigma|\rho)(1 - |\Sigma|\lambda)^2} > 0.$$

Let $\mu = (\sum_{w \in \Sigma_B^*} g(w))^{-1}$ and $h = \mu g$.

For any $u \in \Sigma^*$, $h(\bar{u}) = \mu\rho^{|u|}$, $h(\bar{u}c\Sigma_B^*) = h(\bar{u}c\Sigma^*) = \mu r_A(u\Sigma^*)$ and $h(\bar{u}\Sigma_B^*) = h(\bar{u}\bar{\Sigma}^*) + h(\bar{u}c\Sigma^*) = \mu \left(\frac{\rho^{|u|}}{1 - |\Sigma|\rho^{|u|}} + r_A(u\Sigma^*) \right)$.

Check also that for any $u \in \Sigma^*$,

$$\frac{\rho^{|u|}}{1 - |\Sigma|\rho^{|u|}} + r_A(u\Sigma^*) \geq \frac{\rho^{|u|}}{1 - |\Sigma|\rho^{|u|}} - \sum_{v \in \Sigma^*} |r_A(uv)| \geq \frac{\rho^{|u|}}{1 - |\Sigma|\rho^{|u|}} - \frac{\lambda^{|u|}}{1 - |\Sigma|\lambda^{|u|}} > 0.$$

Therefore, $h(\bar{u}) > 0$ and $h(\bar{u}\Sigma_B^*) > 0$ for every $u \in \Sigma^*$ and any letter $x \in \Sigma$. On the other hand, $h(\bar{u}c\Sigma_B^*) > 0$ iff $r_A(u\Sigma^*) \leq 0$. That is, $p_h = p_r$ iff $r_A(u\Sigma^*) \leq 0$ for every $u \in \Sigma^*$. An algorithm capable to decide whether $p_h = p_r$ could be used to decide whether $r_A(u\Sigma^*) \leq 0$ for every $u \in \Sigma^*$. \square

6.3 Drawing a word according to p_r

Modification of Algorithm 2 in order to draw a word according to the distribution p_r .

```

Input: an MA  $A = \langle \Sigma, Q = \{q_1, \dots, q_n\}, \varphi, \iota, \tau \rangle$  s.t.  $\rho(A) < 1$  and  $r_A(\Sigma^*) = 1$ 
Output: a word  $u$  drawn according to  $p_{r_A}$ 
for  $i = 1, \dots, n$  /* this step is polynomial in  $n$  and is done once */ do
   $s_i \leftarrow r_{A, q_i}(\Sigma^*); \quad e_i \leftarrow \iota(q_i);$ 
 $u \leftarrow \varepsilon;$ 
 $finished \leftarrow false;$ 
 $w \leftarrow \varepsilon; \quad \lambda \leftarrow 1$  /*  $\lambda$  is equal to  $p_{r_A}(w\Sigma^*)$  */;
while not finished do
   $S \leftarrow \emptyset;$ 
   $\lambda \leftarrow 1;$ 
   $v \leftarrow \sum_{i=1}^n e_i \tau(q_i);$ 
  if  $v > 0$  then  $S \leftarrow \{(\varepsilon, v)\};$ 
   $\lambda \leftarrow v;$ 
  for  $x \in \Sigma$  do
     $v \leftarrow \sum_{i,j=1}^n e_i \varphi(q_i, x, q_j) s_j;$ 
    if  $v > 0$  then
       $S \leftarrow S \cup \{(x, v)\};$ 
       $\lambda \leftarrow \lambda + v;$ 
  for  $(x, v) \in S$  do  $(x, v) \leftarrow (x, v/\lambda);$ 
   $x \leftarrow Draw(S)$  /* Draw randomly an element  $(x, p)$  of  $S$  with probability  $p$  */;
  if  $x = \varepsilon$  then
     $finished \leftarrow True;$ 
  else
     $u \leftarrow ux;$ 
    for  $i = 1, \dots, n$  do  $e_i \leftarrow \sum_{j=1}^n e_j \varphi(q_j, x, q_i);$ 

```

Algorithm 3: Algorithm drawing a word according to the distribution p_r .