Identification à la limite d'automates probabilistes résiduels avec probabilité un

François Denis[1], Yann Esposito[1]

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

Résumé Les automates probabilistes (PFA) sont des objets permettant de modéliser des distributions de probabilités définies sur des ensembles de mots. Ils ont la même expressivité que les Modèles de Markov Cachés utilisés dans de très nombreuses applications. Pour une sous-classe des PFAs, les automates probabilistes déterministes (PDFA), des algorithmes d'identification à la limite ont été élaborés. Malheureusement les PDFAs sont beaucoup moins expressifs que les PFAs. Aussi étudions-nous une classe d'expressivité intermédiaire : les automates probabilistes résiduels (PRFA). Nous montrons que les PRFAs à paramètres rationnels sont identifiables à la limite avec une probabilité de 1.

1 Introduction

Les automates probabilistes (PA) sont des objets formels permettant de modéliser des distributions de probabilités définies sur des ensembles de mots appelés aussi *langages stochastiques*[1]. Ils sont composés d'une *structure* qui est un automate fini (NFA) et d'un ensemble de *paramètres* associés aux états et aux transitions représentant la probabilité pour un état d'être initial ou terminal, ou la probabilité qu'une transition soit empruntée. À un automate probabiliste A défini sur l'alphabet Σ, on associe naturellement une distribution de probabilités P_A définie sur Σ^*. Les automates probabilistes ont la même expressivité que les Modèles de Markov Cachés (HMM) [2,3,4] et peuvent donc être utilisés dans les mêmes conditions, pour les mêmes très nombreuses applications (reconnaissance de la parole, analyse de séquences biologiques, ...). Les algorithmes définis à propos des HMMs (voir [5] par exemple) peuvent facilement être adaptés aux PAs. Étant donnés un automate probabiliste A et un mot u, on sait calculer efficacement la probabilité $P_A(u)$ (algorithmes forward-backward) ; on sait aussi, parmi tous les chemins de A reconnaissant u, trouver le plus probable (chemin de Viterbi). Étant donnés une structure d'automate probabiliste A et une suite de mots $u_1, \ldots, u_n$ tirés indépendamment selon une même distribution de probabilité P, le problème consistant à trouver les paramètres de A maximisant la vraisemblance des observations est NP-dur [3]. Mais des algorithmes basés sur le principe EM (*Expectation-Maximisation*) [6] permettent de calculer des approximations de ces paramètres jugées souvent satisfaisantes (algorithmes d'apprentissage de Baum-Welch et de Viterbi). En revanche, le problème consistant à inférer un automate probabiliste (structure et paramètres) d'une suite de mots $u_1, \ldots, u_n$ tirés indépendamment selon une même distribution de probabilités P est encore un sujet de recherche largement ouvert. Dans la plupart des applications des automates probabilistes ou des HMMs, soit la connaissance du domaine permet de choisir une structure a priori (Cf le modèle standard pour l'analyse des séquences biologiques [7]), soit l'on part d'un automate possédant une structure de graphe complet. On se doute qu'il y a sans doute mieux à faire et certains travaux ont confirmé cette intuition [8] : rechercher des structures à partir des données est souvent préférable à poser des structures a priori. L'inférence d'automates probabilistes est donc un sujet de recherche prometteur.

Certains auteurs ont choisi de travailler avec des objets plus généraux que les automates probabilistes : les *automates à multiplicité* (MA)[9,10,11]. Un automate à multiplicité possède les mêmes caractéristiques qu'un automate probabiliste sauf celles qui contraignent l'automate à définir une distribution de probabilités : les paramètres sont donc à valeur dans un corps K quelconque (généralement $\mathbb{Q}$). Un automate à multiplicité A définit donc une fonction f_A de Σ^* à valeurs dans K. L'ensemble des fonctions définies par des automates à multiplicité sont apprenables dans une variante du modèle d'apprentissage exact d'Angluin, dans lequel l'apprenant est supposé pouvoir formuler des requêtes d'appartenance et d'équivalence étendues : pour une cible A fixée, en réponse à une *requête d'appartenance* portant sur un mot u, un oracle retourne la valeur $f_A(u)$; en réponse à une *requête d'équivalence* portant sur un automate à multiplicité B, l'oracle retourne *oui* si $f_A = f_B$ et un couple $(u, f_A(u))$ tel que $f_A(u) \neq f_B(u)$ sinon. L'algorithme proposé dans [11] identifie la cible en un temps $O((|\Sigma| + m)r^{3.376})$ en utilisant r requêtes d'équivalence et $O((|\Sigma| + \log m)r^2)$ requêtes d'appartenance, où r est le nombre d'états de la cible et où m est la longueur du plus grand contre-exemple retourné par l'oracle d'équivalence. Comme les automates probabilistes sont des cas particuliers d'automates à multiplicité, on déduit immédiatement du résultat précédent qu'ils sont apprenables par requêtes d'appartenance et d'équivalence.

Ce beau résultat n'est pourtant pas entièrement satisfaisant. On peut juger qu'il n'est pas très réaliste de supposer qu'on dispose d'oracles d'appartenance et d'équivalence. Cette objection peut pourtant être assez facilement levée. D'une part, on sait qu'on peut se passer de l'oracle d'équivalence en passant du modèle d'apprentissage exact au modèle d'apprentissage PAC avec requêtes d'appartenance [12,13]. D'autre part, on sait qu'on peut transformer tout résultat d'apprentissage exact en un résultat d'apprentissage où l'on fait l'hypothèse que l'apprenant dispose d'un échantillon contenant suffisamment d'information sur la cible : la plupart des algorithmes développés en inférence grammaticale reposent sur cette hypothèse [14,15]. En revanche, deux autres caractéristiques de ce résultat s'avèrent beaucoup plus gênantes : l'algorithme d'apprentissage de [11] suppose qu'on dispose exactement de la valeur $f_A(u)$ ce qui n'est jamais le cas en pratique et il retourne un automate à multiplicité qui calcule la fonction définie par la cible mais qui n'est pas lui-même un automate probabiliste : l'apprentissage est *impropre*, selon la terminologie utilisée dans l'apprentissage PAC. La conjonction de ces deux petits défauts en génère un grand. En effet, on peut facilement vérifier si un automate à multiplicité est un automate probabiliste ; en revanche, il est indécidable de savoir si un automate à multiplicité génère une fonction positive. Cela a pour conséquence que même si l'on était capable de modifier l'algorithme défini dans [11] de façon à ce qu'il puisse travailler avec des valeurs approchées $\widehat{P_A}(u)$ de $P_A(u)$ (ce qui ne semble pas si simple), il retournerait une fonction qui ne calculerait pas nécessairement une distribution de probabilités et sans qu'on puisse s'en assurer. Il est donc indispensable de travailler dans un cadre théorique qui suppose que les données sont approximatives et qui impose de retourner une distribution de probabilités.

Le modèle d'apprentissage par identification à la limite de Gold a été étendu de façon à pouvoir rendre compte de l'apprentissage de langages stochastiques. On suppose dans ce modèle qu'une cible A étant fixée, l'apprenant dispose d'une suite d'observations $u_1, \ldots, u_n, \ldots$ tirées indépendamment selon P_A. Un algorithme d'apprentissage $\mathcal{A}$, prenant en entrée $u_1, \ldots, u_n$ retournera un automate A_n. On demande qu'avec une probabilité de 1, la structure d'un automate permettant de générer P_A soit identifiée à partir d'un certain rang et que P_{A_n} converge vers P_A ; on peut aussi demander, par exemple lorsque les paramètres de A sont des rationnels, que P_{A_n} soit égal à P_A à partir d'un certain rang. Si la structure de la cible est un automate déterministe, ces trois points ont été obtenus : l'ensemble des automates déterministes probabilistes à paramètres rationnels sont identifiables à la limite avec une probabilité de 1 [16,17,18]. Ce résultat théorique donne lieu à des implémentations qui peuvent être utilisées dans les cas pratiques. Malheureusement, et contrairement au cas non probabiliste, les automates déterministes probabilistes ont une expressivité très inférieure à celle des automates probabilistes dans leur ensemble : on peut démontrer que les langages stochastiques qu'ils reconnaissent sont exactement ceux qui n'ont qu'un nombre fini de langages résiduels, où le *langage résiduel* d'un langage stochastique P est le langage $u^{-1}P$ défini par $u^{-1}P(v) = P(uv)/P(u\Sigma^*)$. Il semble indispensable de pouvoir générer des langages plus riches.

La classe des automates probabilistes résiduels (PRFA) a été introduite dans [19]. Ces automates sont basés sur des composantes naturelles des langages qu'ils reconnaissent : leurs langages résiduels. Un automate probabiliste A est un PRFA ssi chacun de ses états définit un langage résiduel de P_A. Les langages définis par des PRFAs ont une caractérisation intrinsèque : il s'agit des langages stochastiques P admettant une base finie de langages résiduels, c'est-à-dire tels qu'il existe un ensemble fini de mots $u_1, \ldots, u_n$ tel que pour tout mot v, $v^{-1}P = \sum \alpha_i u_i^{-1}P$ où les poids α_i sont des réels positifs. En reformulant cette définition géométriquement, un langage stochastique peut être engendré par un PRFA ssi ses langages résiduels (éventuellement en nombre infini) sont contenus dans un polytope dont les sommets sont des langages résiduels. Un premier résultat d'inférence de PRFAs a été montré dans [19]. Nous démontrons dans cet article que les PRFAs à paramètres rationnels sont identifiables à la limite avec une probabilité de 1.

2 Préliminaires

2.1 Automates et langages

Soit Σ un *alphabet* fini, et soit Σ^* l'ensemble des mots sur Σ. Le mot vide sera noté ε. On suppose que Σ^* est ordonné d'abord par longueur croissante et pour chaque longueur, selon l'ordre lexicographique. Un *langage* est un sous-ensemble de Σ^*. On désigne le plus grand mot d'un langage fini L par $\max(L)$. Si L est un langage, on note $\mathrm{pref}(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, uv \in L\}$. L est dit *préfixiel* si et seulement si $L = \mathrm{pref}(L)$ Un *automate fini non déterministe* (NFA) est un quintuplet $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ où Q est un ensemble fini d'états, $Q_0 \subseteq Q$ est l'ensemble des états initiaux, $F \subseteq Q$ est l'ensemble des états terminaux, δ est la *fonction de transition* définie de $Q \times \Sigma$ dans 2^Q. Nous noterons aussi par δ l'extension de la fonction de transition définie de $2^Q \times \Sigma^*$ vers 2^Q. Un NFA est *déterministe* (DFA) si Q_0 est un singleton et si $\forall q \in Q$, $\forall x \in \Sigma$, $\mathrm{Card}(\delta(q, x)) \leq 1$.

Un mot $u \in \Sigma^*$ est *reconnu* par un NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ si $\delta(Q_0, u) \cap F \neq \varnothing$ et le langage reconnu par A est $L_A = \{u \in \Sigma^* \mid \delta(Q_0, u) \cap F \neq \varnothing\}$. Soit $q \in Q$. On note $L_{A,q}$ le langage $\{v \in \Sigma^* \mid \delta(q, v) \cap F \neq \varnothing\}$.

Soit L un langage et u un mot. Le *langage résiduel* de L par rapport à u est $u^{-1}L = \{v \mid uv \in L\}$. Un automate *fini à états résiduels (RFSA)* est un NFA $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ tel que pour tout état $q \in Q$, $L_{A,q}$ est un résiduel de L_A [20].

2.2 Automates probabilistes et langages stochastiques

Soit un corps K, un *automate à multiplicité (MA)* est un quintuplet $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ où Q est un ensemble fini d'états, $\varphi : Q \times \Sigma \times Q \to K$, $\iota : Q \to K$, $\tau : Q \to K$ sont respectivement les fonctions de transition, d'initialisation et de terminaison.

On étend la fonction de transition φ à $Q \times \Sigma^* \times Q$ par : $\varphi(q, wa, r) = \sum_{s \in Q} \varphi(q, w, s)\varphi(s, a, r)$ où $a \in \Sigma$ et $\varphi(q, \varepsilon, r) = 1$ si $q = r$ et 0 sinon. On étend encore φ à $Q \times 2^{\Sigma^*} \times 2^Q$ par $\varphi(q, U, R) = \sum_{w \in U} \sum_{r \in R} \varphi(q, w, r)$. La fonction f_A associée à un MA A est définie par $f_A(u) = \sum_{q \in Q} \sum_{r \in Q} \iota(q)\varphi(q, u, r)\tau(r)$.

L'ensemble des états initiaux est défini par $Q_I = \{q \in Q \mid \iota(q) \neq 0\}$. L'ensemble des états terminaux est défini par $Q_T = \{q \in Q \mid \tau(q) \neq 0\}$. Le *support* d'un MA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ est le NFA $\langle \Sigma, Q, Q_I, Q_T, \delta \rangle$ tel que $\delta(q, x) = \{r \mid \varphi(q, x, r) \neq 0\}$. L'ensemble des états atteignables est $Q_{\mathrm{att}} = \delta(Q_I, \Sigma^*)$. Un *Automate Fini Probabiliste (PFA)* est un automate à multiplicité dont les fonctions φ, ι et τ prennent leur valeurs dans $[0, 1]$ et satisfont : $\sum_{q \in Q} \iota(q) = 1$, $\tau(q) + \sum_{a \in \Sigma} \sum_{r \in Q} \varphi(q, a, r) = 1$ pour tout état q, et $\varphi(q, \Sigma^*, Q_T) \neq 0$ pour tout $q \in Q_{\mathrm{att}}$. Un *automate fini déterministe probabiliste (PDFA)* est un PFA dont le support est déterministe.

Un *langage stochastique* sur Σ est une distribution de probabilités sur Σ^*. La fonction P_A associée à un PFA A est un langage stochastique sur Σ qui sera dit *généré* par A. Pour tout état q, on note $P_{A,q}$ le langage stochastique généré par le PFA $< \Sigma, Q, \varphi, \iota_q, \tau >$ où $\iota_q(q) = 1$. Soit P un langage stochastique et soit u un mot tel que $P(u\Sigma^*) \neq 0$. Le *langage résiduel* $u^{-1}P$ est le langage stochastique qui associe à tout mot w la probabilité $u^{-1}P(w) = P(uw)/P(u\Sigma^*)$. Si $\mathcal{P}$ est un ensemble fini de langages stochastiques, alors *l'enveloppe convexe* de $\mathcal{P}$ est définie par $\mathrm{conv}\,\mathcal{P} = \left\{\sum_{P\in\mathcal{P}} \alpha_P P \mid \forall P \in \mathcal{P}, \alpha_P \geq 0 \text{ et } \sum_{P\in\mathcal{P}} \alpha_P = 1\right\}$. Un *résiduel premier* de P est un résiduel R tel que pour tout ensemble fini $\mathcal{R}$ de résiduels de P ne contenant pas R, $p \notin \mathrm{conv}\,\mathcal{R}$. Un *automate fini résiduel probabiliste (PRFA)* est un PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau\rangle$ dont tout état définit un langage résiduel, c'est-à-dire tel que $\forall q \in Q, \exists u \in \Sigma^*, P_{A,q} = u^{-1}P_A$ [19].

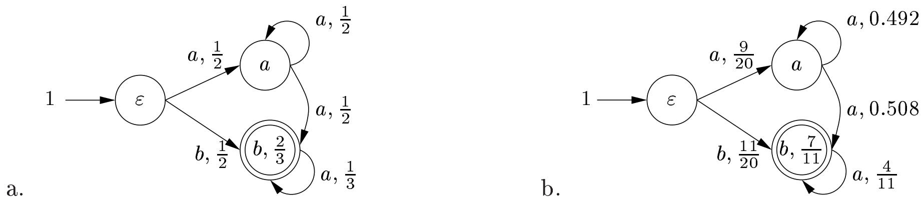

Fig. 1. Un PRFA cible et le PRFA appris par l'algorithme avec une vingtaine de mots.

La figure 1.a représente le PRFA $\langle \Sigma, Q, \varphi, \iota, \tau\rangle$ où $\Sigma = \{a, b\}$, $Q = \{\varepsilon, a, b\}$, $\iota(\varepsilon) = 1$, $\tau(b) = \frac{2}{3}$, $\varphi(\varepsilon, a, a) = \frac{1}{2}$, $\varphi(\varepsilon, b, b) = \frac{1}{2}$, $\varphi(a, a, a) = \frac{1}{2}$, $\varphi(a, a, b) = \frac{1}{2}$ et $\varphi(b, a, b) = \frac{1}{3}$. On voit que $\varepsilon^{-1}P_A = P_{A,\varepsilon}$, $a^{-1}P_A = P_{A,a}$ et $b^{-1}P_A = P_{A,b}$. Les deux PRFAs de la figure 1 possèdent le même support.

Les PDFAs sont des PRFAs. Il a aussi été montré dans [19] que les PFAs sont plus expressifs que les PRFAs, eux-mêmes étant plus expressifs que les PDFAs. Plus précisément, les langages stochastiques générés par les PDFAs sont ceux qui possèdent un nombre fini de résiduels. Les langages générés par les PRFAs sont les langages P dont l'ensemble $\mathcal{P}$ des langages résiduels premiers est fini et dont tout langage résiduel $w^{-1}P$ appartient à $\mathrm{conv}\,\mathcal{P}$. Les langages générés par les PFAs sont les langages stochastiques P tels qu'il existe un ensemble fini de langages stochastiques $\mathcal{P} = \{P_1, \ldots, P_n\}$ tel que pour tout P_i et pour tout résiduel $w^{-1}P_i$, $w^{-1}P_i \in \mathrm{conv}\,\mathcal{P}$ et tel que $P \in \mathrm{conv}\,\mathcal{P}$ (voir [21] pour une étude complète).

3 Apprentissage impropre à partir de données approximatives : une difficulté.

Supposons que nous voulions apprendre le langage stochastique P généré par un PFA. Il est proposé dans [1,10] des algorithmes qui, à partir des probabilités $P(w)$ de certains mots w, renvoient un automate à multiplicité qui génère P. Il semble naturel de vouloir étendre ces méthodes aux cas approchés. Malheureusement, nous ne pouvons pas le faire, et ce pour deux raisons : avec des valeurs approchées, nous ne pouvons obtenir que des MAs approchés qui ne généreront pas nécessairement un langage stochastique aussi proches soient-ils de la cible.

Nous ne connaissons pas de procédure de décision qui permette de savoir si un MA génère un langage stochastique. Nous montrons que savoir si un MA génère une fonction positive est indécidable.

3.1 Problème de robustesse de la représentation en MA.

La représentation de langages stochastiques par des MAs n'est pas robuste. Une extension naturelle d'un des algorithmes donné dans [1,10] prendra en entrée un échantillon de mots générés par un langage P_A et retournera dans le meilleur des cas un MA hypothèse H dont le support sera identique au support de A et dont les valeurs portées par les arêtes seront proches de celles de A. Malheureusement nous avons le résultat négatif suivant : il existe des MAs A et B, ayant même structure et des paramètres arbitrairement proches, dont seul le premier génère un langage stochastique (Cf. figure 2). De plus, savoir si un MA génère une fonction positive est indécidable.

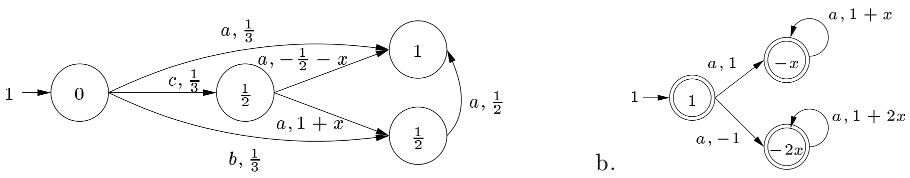

Fig. 2. Deux MAs qui génèrent des langages stochastiques si $x = 0$. Si $x > 0$, le premier génère des valeurs négatives et le second des valeurs non bornées.

3.2 Problème de décision

Proposition 1. *Soit F un MA. Savoir si f_F est une fonction positive est indécidable.*

Démonstration. Soit $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ un PFA accepteur, et soit q_t son unique état terminal. Considérons le MA (figure 3). $F = \langle \Sigma \cup \{\#\}, Q \cup \{q_i, q_h, q_{h1}, q_{h2}, q_c\}$, φ^F, ι^F, $\tau^F\rangle$ où # est une nouvelle lettre, q_i est le seul état initial, q_h est l'état puits générant la fonction nulle sur les mots, q_c est l'état constant qui attribue la valeur 1 à tous les mots, $\iota^F(q) = 1$ si $q = q_i$ et 0 sinon, $\tau^F(q) = 1$ si $q \in \{q_c, q_t, q_{h2}\}$, -1 si $q = q_{h1}$ et 0 sinon, φ^F est identique à φ sur $Q \times \Sigma \times Q$, $\varphi^F(q, \#, q_h) = 1 - \varphi(q, \Sigma, Q)$ pour tout état q de $Q \cup \{q_i, q_c\}$, $\varphi^F(q_i, a, q) = \iota(q)$ pour toute lettre a et tout état de Q, $\varphi^F(q_h, \#, q_{h1}) = \varphi^F(q_h, \#, q_{h2}) = \frac{1}{2}$, $\varphi^F(q_c, a, q_c) = 1$ pour toute lettre $a \in \Sigma$, $\varphi^F(q_{h1}, \#, q_h) = 2$ et φ^F est nulle pour toutes les autres transitions.

Nous avons $P_F(au) = P_A(u) - \lambda$. Sachant que nous avons construit un automate tel que tous les mots qui se terminent par # sont produit avec un valeur nulle, si nous pouvions décider $\exists w \in \Sigma^*, P_F(w) < 0$, alors, nous pourrions décider $\exists w \in \Sigma^*, P_A(w) < \lambda$ qui est indécidable. □

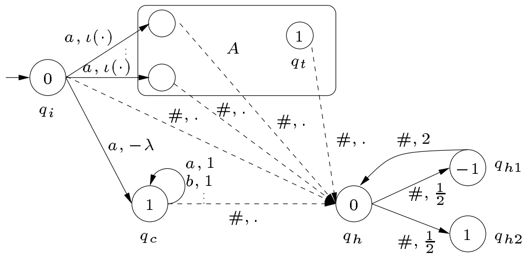

Fig. 3. Un MA qui simule un PFA accepteur.

4 Identification à la limite de la structure des PRFAs.

La plupart des algorithmes d'inférence de PDFAs construisent d'abord un automate arborescent à partir de l'échantillon, recherchent les états qui semblent générer les mêmes langages puis les fusionnent. Notre algorithme construit également un automate arborescent mais recherche plus généralement si le langage défini par un état appartient à la partie convexe des langages définis par les autres états. Dans la suite nous ne considérons que les langages stochastiques générables par des PRFAs.

4.1 Préliminaires

Nous montrons que lorsque la taille de l'échantillon croît, alors, à la limite, travailler avec des valeurs exactes ou approchées est similaire. En particulier nous montrons que les tests d'appartenances d'un langage résiduel à l'enveloppe convexe d'autres langages résiduels deviennent équivalents.

Soit P un langage stochastique. Un *échantillon stochastique* S de P est une suite infinie de mots générés selon la distribution de probabilités P. On notera S_n la séquence composée des n premiers mots de S. Le nombre d'occurrences du mot w dans S_n sera noté $S_n(w)$. Pour tout sous-ensemble $W \subset \Sigma^*$, $S_n(W) = \sum_{w \in W} S_n(w)$. Par abus de notation on écrira $w \in S_n$ ssi $S_n(w) \neq 0$. La séquence S_n définit un langage stochastique P_n où $P_n(w) = S_n(w)/n$.

Soit X un ensemble dénombrable, soit $\mathcal{A} = (A_i)_{i \in I}$ une famille de sous-ensembles de X et soit S un sous-ensemble fini de X. On dit que S est *pulvérisé* par $\mathcal{A}$ si $\{S \cap A_i \mid i \in I\} = 2^S$. La VC-dimension VC($\mathcal{A}$) de $\mathcal{A}$ est le plus grand cardinal d'un ensemble pulvérisé par $\mathcal{A}$. Par exemple, soit $X = \Sigma^*$ et soit $\mathcal{A} = (w\Sigma^*)_{w \in \Sigma^*}$. Si Card($\Sigma$) = 1, on peut facilement montrer que $\{\varepsilon\}$ est pulvérisé par $\mathcal{A}$ mais qu'aucun ensemble contenant deux éléments ne peut être pulvérisé. Si a et b sont deux lettres distinctes de Σ, il est clair que $\{a, b\}$ est pulvérisé par $\mathcal{A}$. Maintenant, soit $r, s, t \in \Sigma^*$ et soit $S = \{r, s, t\}$. Soit u_1 (resp. u_2, u_3) le plus grand préfixe commun de r et s (resp. r et t, s et t). Ces mots sont comparables deux à deux. Supposons que u_1 soit le plus petit : il n'y a pas de mot w tel que $w\Sigma^* \cap S = \{r, s\}$. Aucun sous-ensemble contenant plus de deux éléments ne peut être pulvérisé. Donc, $\mathrm{VC}(\mathcal{A}) = 1$ si $\mathrm{Card}(\Sigma) = 1$ et $\mathrm{VC}(\mathcal{A}) = 2$ si $\mathrm{Card}(\Sigma) > 1$.

Soit P une distribution de probabilités sur X et soit S_n un échantillon contenant n éléments de X indépendamment générés selon P. Il peut être montré [22,23] que pour n'importe quel paramètre de confiance δ, avec une probabilité supérieure à $1 - \delta$, nous avons pour tout $i \in I$,

$$|P_n(A_i) - P(A_i)| \leq c\sqrt{\frac{\mathrm{VC}(\mathcal{A}) - \log\frac{\delta}{4}}{n}} \tag{1}$$

où c est une constante universelle. Soit $\Psi_1(\epsilon, \delta) = \frac{c^2}{\epsilon^2}(2 - \log\frac{\delta}{4})$.

Lemme 1. *Pour tout paramètre de précision ϵ, tout paramètre de confiance δ, pour tout $n \geq \Psi_1(\epsilon, \delta)$, avec une probabilité supérieure à $1 - \delta$ nous avons, $|P_n(w\Sigma^*) - P(w\Sigma^*)| \leq \epsilon$ pour tout $w \in \Sigma^*$.*

Démonstration. Il suffit d'appliquer (1). □

Définition 1. *Soit P un langage stochastique, $U \cup \{v\}$ un ensemble fini de mots tels que $\forall u \in U \cup \{v\}, u^{-1}P$ est défini. On définit le système $E(v, U, P)$ d'équations et d'inéquations linéaires sur les variables $(x_u^v)_{u \in U}$: $x_u^v \geq 0$ pour tout $u \in U$ et $v^{-1}P = \sum_{u \in U} x_u^v u^{-1}P$.*

Soit S un échantillon stochastique de P, $n \in \mathbb{N}$ et $\epsilon > 0$. Supposons que $\forall u \in U \cup \{v\}, u^{-1}P_n$ soit défini. Soit W l'ensemble des mots w tels que : $\exists u \in U \cup \{v\}, P_n(uw\Sigma^*) \neq 0$. On définit le système $I(v, U, S_n, \epsilon)$: $x_u^v \geq 0$ pour tout $u \in U$ et pour tout $w \in W$: $\left|v^{-1}P_n(w\Sigma^*) - \sum_{u \in U} x_u^v u^{-1}P_n(w\Sigma^*)\right| \leq \epsilon$.

On désignera par $\mathrm{sol}(E)$ l'ensemble des solutions du système E.

Ces deux systèmes linéaires permettent de tester l'appartenance d'un résiduel à l'enveloppe convexe d'autres résiduels. En effet si $\mathrm{sol}(E(v, U, P))$ n'est pas vide alors cela signifie que $v^{-1}P \in \mathrm{conv}\left\{u^{-1}P \mid u \in U\right\}$. Les systèmes E nécessitent une connaissance des valeurs exactes des probabilités alors que les systèmes I peuvent s'obtenir à partir d'un échantillon fini.

Exemple 1. Prenons comme exemple le PRFA de la figure 1.a et appelons P le langage qu'il génère. On a alors $\mathrm{sol}(E(a, \{\varepsilon\}, P)) = \mathrm{sol}(E(b, \{\varepsilon, a\}, P)) = \varnothing$, $\mathrm{sol}(E(aa, \{\varepsilon, a, b\}, P)) = \left\{x_\varepsilon^{aa} = 0, x_a^{aa} = \frac{1}{2}, x_b^{aa} = \frac{1}{2}\right\}$ puisqu'on a $aa^{-1}P = \frac{1}{2}a^{-1}P + \frac{1}{2}b^{-1}P$ et de même $\mathrm{sol}(E(ba, \{\varepsilon, a, b\}, P)) = \left\{x_\varepsilon^{ba} = 0, x_a^{ba} = 0, x_b^{ba} = 1\right\}$. Soit S un échantillon associé à P et soit S_{20} composé des 20 mots générés avec répétitions (le nombre entre parenthèses correspond au nombre d'occurrences) : $b(7)$, $aa(4)$, $ba(3)$, $aaa(1)$, $baa(1)$, $aaaa(3)$, $aaaaaaa(1)$. Les préfixes des mots de S_{20} avec leur nombre d'apparition sont : $\varepsilon(20)$, $a(9)$, $b(11)$, $aa(9)$, $ba(4)$, $aaa(5)$, $baa(1)$, $aaaa(4)$, $aaaaa(1)$, $aaaaaa(1)$, $aaaaaaa(1)$. On a $P_n(a\Sigma^*) = \frac{9}{20}$,

$P_n(b\Sigma^*) = \frac{11}{20}$, $a^{-1}P_n(a\Sigma^*) = \frac{9}{9}$, $b^{-1}P_n(a\Sigma^*) = \frac{4}{11}$. Pour $I\left(aa, \{\varepsilon, a, b\}, S_{20}, \frac{1}{8}\right)$ l'inéquation associée au mot a est : $|aa^{-1}P_n(a\Sigma^*) - \sum_{u\in\{\varepsilon,a,b\}} x_u^{aa} u^{-1}P_n(a\Sigma^*)| \leq \epsilon$, c'est-à-dire $\left|\frac{5}{9} - x_\varepsilon^{aa}\frac{9}{20} - x_a^{aa}\frac{9}{9} - x_b^{aa}\frac{4}{11}\right| \leq \frac{1}{8}$.

En particulier, $\mathrm{sol}\left(I\left(\{a\}, \{\varepsilon\}, S_{20}, \frac{1}{8}\right)\right) = \mathrm{sol}\left(I\left(\{b\}, \{\varepsilon, a\}, S_{20}, \frac{1}{8}\right)\right) = \varnothing$, $\left\{x_\varepsilon^{ba} = 0, x_a^{ba} = 0, x_b^{ba} = 1\right\} \in \mathrm{sol}\left(I\left(\{ba\}, \{\varepsilon, a, b\}, S_{20}, \frac{1}{8}\right)\right)$ et finalement nous avons $\{x_\varepsilon^{aa} = 0, x_a^{aa} = 0,492, x_b^{aa} = 0,508\} \in \mathrm{sol}\left(I\left(\{aa\}, \{\varepsilon, a, b\}, S_{20}, \frac{1}{8}\right)\right)$, c'est-à-dire des résultats proches de ceux obtenus avec $E(aa, \{\varepsilon, a, b\}, P)$ et $E(ba, \{\varepsilon, a, b\}, P)$.

Lemme 2. *Lorsque l'échantillon contient un nombre suffisant de mots commençant par v et par tout mot de U, alors avec une probabilité "contrôlée", les solutions du système $E(u, U, P)$ seront des solutions du système $I(v, U, S_n, \epsilon)$. Plus formellement, soit P un langage stochastique, $U \cup \{v\}$ un ensemble fini de mots, $m = \mathrm{Card}\,(U \cup \{v\})$, S un échantillon stochastique de P, $\epsilon > 0$, $\delta > 0$. Supposons qu'il existe un entier n tel que $\forall u \in U \cup \{v\}$, $\mathrm{Card}(u^{-1}S_n) \geq \Psi_1(\epsilon/2, \delta/m)$. Alors, avec un probabilité supérieure à $1-\delta$, $\mathrm{sol}\,(E(v, U, P)) \subseteq \mathrm{sol}\,(I(v, U, S_n, \epsilon))$.*

Démonstration. Soit $(x_u^v)_{u\in U}$ une solution du système $E(v, U, P)$ alors $\sum_{u\in U} x_u^v = 1$ et

$$\begin{aligned}&\left|v^{-1}P_n(w\Sigma^*) - \textstyle\sum_{u\in U} x_u^v \cdot u^{-1}P_n(w\Sigma^*)\right| \\ &\quad\leq \left|v^{-1}P_n(w\Sigma^*) - v^{-1}P(w\Sigma^*)\right| + \textstyle\sum_{u\in U} x_u^v \left|u^{-1}P_n(w\Sigma^*) - u^{-1}P(w\Sigma^*)\right|\end{aligned}$$

pour tout $w \in \Sigma^*$. D'après le lemme 1, pour tout $u \in U \cup \{v\}$, avec une probabilité supérieure à $1 - \frac{\delta}{m}$, nous avons $\left|u^{-1}P_n(w\Sigma^*) - u^{-1}P(w\Sigma^*)\right| \leq \frac{\epsilon}{2}$ pour tout mot w. Soit A_u cet évènement. Nous avons $P\left(\cap_{u\in U\cup\{v\}} A_u\right) = 1 - P\left(\cup_{u\in U\cup\{v\}} \overline{A_u}\right) \geq 1 - \sum_{u\in U\cup\{v\}} p\left(\overline{A_u}\right) \geq 1 - m\left(\frac{\delta}{m}\right) = 1 - \delta$.

Donc avec une probabilité supérieure à $1 - \delta$, nous avons pour tout mot w, $\left|v^{-1}P_n(w\Sigma^*) - \sum_{u\in U} x_u^v u^{-1}P_n(w\Sigma^*)\right| \leq \frac{\epsilon}{2} + \sum_{u\in U} x_u^v \frac{\epsilon}{2} = \epsilon$ pour tout $(x_u^v)_{u\in U}$ solution du système $E(v, U, P)$. □

Lemme 3. *Soit $(X_i)_{i=\{1,\ldots,n\}}$ des variables aléatoires indépendantes suivant des lois de Bernoulli de même paramètre $p > 0$. Soit $Y_n = \sum_{i=1}^n X_i$, $\delta > 0$ et soit $k \in \mathbb{N}$. Si $n \geq \sup(2k/p, -2(\ln\delta)/p^2)$ alors avec une probabilité supérieure à $1 - \delta$, nous avons $Y_n \geq k$.*

Démonstration. Par l'inégalité de Hoeffding, nous avons $P\left(\frac{Y_n}{n} - p \leq -\epsilon\right) = P\left(Y_n \leq n(p - \epsilon)\right) \leq e^{-2n\epsilon^2}$ pour tout $\epsilon > 0$. Soit $\epsilon = \sqrt{-\frac{\ln\delta}{2n}}$. Nous avons $e^{-2n\epsilon^2} = \delta$. Comme $n \geq -2(\ln\delta)/p^2$, nous avons $\epsilon \leq \sqrt{-\frac{p^2\ln\delta}{2(-2\ln\delta)}} = p/2$ et comme $n \geq 2k/p$, nous avons $k \leq n\left(p - \frac{p}{2}\right) \leq n(p - \epsilon)$. □

Définissons $\Psi(\epsilon, \delta, \alpha, m) = \sup\left\{2\Psi_1\left(\frac{\epsilon}{2}, \frac{\delta}{2m}\right)/\alpha, -2\ln\frac{\delta}{2m}/\alpha^2\right\}$.

Lemme 4. *Lorsque l'échantillon est suffisamment grand, alors, avec une probabilité "contrôlée", les solutions des systèmes E seront solutions des systèmes I. Plus formellement, soit P un langage stochastique, $U \cup \{v\}$ un ensemble fini de* *mots,* $\alpha = \min\{P(u\Sigma^*) \mid u \in U \cup \{v\}\}$*,* $m = \mathrm{Card}(U \cup \{v\})$*,* S *un échantillon stochastique de* P*,* $\epsilon > 0$*,* $\delta > 0$*. Supposons* $\alpha > 0$*. Si* $n \geq \Psi(\varepsilon, \delta, \alpha, m)$*, alors avec une probabilité supérieure à* $1 - \delta$*,* $\mathrm{sol}\,(E(v, U, P)) \subseteq \mathrm{sol}\,(I(v, U, S_n, \epsilon))$*.*

Démonstration. En utilisant le lemme précédent, nous avons que pour chaque $u \in U \cup \{v\}$, avec une probabilité supérieure à $1 - \frac{\delta}{2m}$, $\mathrm{Card}(u^{-1}S_n) \geq \Psi_1(\frac{\epsilon}{2}, \frac{\delta}{2m})$. Donc cette propriété est vraie pour tout $u \in U \cup \{v\}$ avec une probabilité supérieure à $1 - \frac{\delta}{2}$. Dans le cas où toutes ces inéquations sont satisfaites, du lemme 2 nous déduisons qu'avec une probabilité d'au moins $1 - \frac{\delta}{2}$, $\mathrm{sol}\,(E(v, U, P)) \subseteq \mathrm{sol}\,(I(v, U, S_n, \epsilon))$ ce qui permet de conclure. □

Définition 2. *Soit* P *un langage stochastique et* U *un ensemble préfixiel tel que pour tout mot* u *de* U*,* $u^{-1}P$ *est défini. Nous définissons la* frontière *de* U *par rapport à* P *par :* $\mathrm{fr}\,(U, P) = \{ux \mid u \in U, x \in \Sigma, ux \notin U, (ux)^{-1}P \text{ est défini}\}$*.*
Maintenant, considérons les systèmes d'inéquations suivant :
$E(U, P) = \bigcup_{v \in \mathrm{fr}(U,P)} E(v, U, P)$*, et* $I(U, S_n, \epsilon) = \bigcup_{v \in \mathrm{fr}(U,P_n)} I(v, U, S_n, \epsilon)$*.*

On verra dans la suite que si $E(U, P)$ admet une solution alors, on peut construire un PRFA A qui génère P avec U comme ensemble d'états et tel que $\forall u \in U, P_{A,u} = u^{-1}P_A$. Et on remarquera que $E(U, P)$ admet une solution est équivalent à dire que pour tout mot $v \in \mathrm{fr}\,(U, P)$, $E(v, U, P)$ admet une solution.

Exemple 2. Soit P le langage généré par le PRFA de la figure 1.a, alors : $\mathrm{fr}\,(\{\epsilon\}, P) = \{a, b\}$, $\mathrm{sol}\,(E(\{\varepsilon\}, P)) = \mathrm{sol}\left(I\left(\{\varepsilon\}, S_n, \frac{1}{8}\right)\right) = \varnothing$. $\mathrm{fr}\,(\{\epsilon, a\}, P) = \{b, aa\}$, $\mathrm{sol}\,(E(\{\varepsilon, a\}, P)) = \mathrm{sol}\left(I\left(\{\varepsilon, a\}, S_n, \frac{1}{8}\right)\right) = \varnothing$. $\mathrm{fr}\,(\{\epsilon, a, b\}, P) = \{aa, ba\}$, $E(\{\varepsilon, a, b\}, P)$ et $I\left(\{\varepsilon, a, b\}, S_n, \frac{1}{8}\right)$ admettent des solutions.

Proposition 2. *Soit* $W = \mathrm{fr}\,(U, P) \cup U$*,* $\alpha = \min\{P(v\Sigma^*) \mid v \in W\}$*,* $m = \mathrm{Card}(W)$*,* $k = \mathrm{Card}(\mathrm{fr}\,(U, P))$*,* $\epsilon > 0$ *et* $\delta > 0$*. Si* $n \geq \Psi(\epsilon, \frac{\delta}{k}, \alpha, m)$ *alors avec une probabilité d'au moins* $1 - \delta$*,* $\mathrm{sol}\,(E(U, P)) \subseteq \mathrm{sol}\,(I(U, S_n, \epsilon))$*.*

Démonstration. D'après le lemme 4, pour tout mot v de $\mathrm{fr}\,(U, P)$, la probabilité pour que $\mathrm{sol}\,(E(v, U, P)) \nsubseteq \mathrm{sol}\,(I(v, U, S_n, \varepsilon))$ est inférieure à $\frac{\delta}{k}$. La probabilité pour que $\forall v \in \mathrm{fr}\,(U, P)$, $\mathrm{sol}\,(E(v, U, P)) \subseteq \mathrm{sol}\,(I(v, U, S_n, \varepsilon))$ est supérieure à $1 - \sum_{i=1}^{k} \frac{\delta}{k} = 1 - \delta$. □

Lemme 5. *Il existe des suites* $(\epsilon_n)_{n\in\mathbb{N}}$*,* $(\delta_n)_{n\in N}$*,* $(\alpha_n)_{n\in\mathbb{N}}$*,* $(m_n)_{n\in\mathbb{N}}$ *et* $(k_n)_{n\in\mathbb{N}}$ *telles que* $\epsilon_n \to 0$*,* $\sum_{n\in\mathbb{N}} \delta_n < \infty$*,* $\alpha_n \to 0$*,* $m_n \to +\infty$*,* $k_n \to +\infty$ *et il existe un* N *tel que* $\forall n > N, n \geq \Psi\left(\epsilon_n, \frac{\delta_n}{k_n}, \alpha_n, m_n\right)$*.*

Démonstration. Par exemple $\alpha_n = \epsilon_n = n^{-\frac{1}{4}}$, $\delta_n = \frac{1}{n^2}$ et $m_n = k_n = n$. □

Note 1. Jusqu'à la fin de l'article $(\epsilon_n)_{n\in\mathbb{N}}$, $(\delta_n)_{n\in\mathbb{N}}$, $(\alpha_n)_{n\in\mathbb{N}}$, $(k_n)_{n\in\mathbb{N}}$ et $(m_n)_{n\in\mathbb{N}}$ seront des suites qui posséderont les propriétés du lemme précédent. Lorsque le contexte ne sera pas ambigu, $I\,(U, S_n, \varepsilon_n)$ sera noté par I_n et $E(U, P)$ par E.

Lemme 6. *[Borel-Cantelli] Soit* $(A_n)_{n\in\mathbb{N}}$ *une suite d'évènements de probabilités supérieures ou égales à* $1 - \delta_n$ *où* $\sum_{n\in\mathbb{N}} \delta_n < \infty$*. Alors avec une probabilité de 1, il existe un entier* N *tel que pour tout* $n > N$*,* A_n *se produit.*

Lemme 7. *Avec probabilité un, il existe une étape après laquelle les solutions des systèmes E sont des solutions des systèmes I_n. Plus formellement, pour tout ensemble fini de mots U tel que $\forall u \in U$, $u^{-1}P$ est défini, avec probabilité* 1, *il existe un N tel que pour tout $n > N$,* $\mathrm{sol}\left(E(U, S_n)\right) \subseteq \mathrm{sol}\left(I\left(U, S_n, \varepsilon_n\right)\right)$.

Démonstration. La preuve se déduit de la proposition 2 et des lemmes 5 et 6. □

Lemme 8. *Avec une probabilité un, il existe une étape après laquelle les systèmes I_n n'admettent une solution que si les systèmes E admettent une solution. Plus formellement, Pour tout ensemble fini de mots U tel que $\forall u \in U$, $u^{-1}P$ est définit et* $\mathrm{sol}\left(E\left(U, P\right)\right) = \varnothing$, *avec probabilité* 1, *il existe un N tel que pour tout $n > N$,* $\mathrm{sol}\left(I\left(U, S_n, \epsilon_n\right)\right) = \varnothing$.

Démonstration. Supposons que $\forall N \in \mathbb{N}, \exists n \geq N$ tel que $I(U, S_n, \epsilon_n)$ ait une solution. Soit $\phi : \mathbb{N} \to \mathbb{N}$ telle que $\phi(n) \geq n$ et $I(U, S_{\phi(n)}, \epsilon_{\phi(n)})$ ait une solution $(x_u^{v,\phi(n)})_{u \in U}$. Comme les solutions appartiennent au compact $[0,1]^{\mathrm{Card}(U)}$, on peut en extraire une sous-suite convergente $(x_u^{v,\sigma(\phi(n))})$ de limite $(x_u^v)_{u\in U}$.
$\left|v^{-1}P(w\Sigma^*) - \sum_{u\in U} x_u^v P(w\Sigma^*)\right| \leq \left|v^{-1}P(w\Sigma^*) - v^{-1}P_{\sigma(n)}(w\Sigma^*)\right| + \left|v^{-1}P_{\sigma(n)}(w\Sigma^*) - \sum_{u\in U} x_u^{v,\sigma(n)} P_{\sigma(n)}(w\Sigma^*)\right| + \left|\sum_{u\in U} x_u^{v,\sigma(n)} P_{\sigma(n)}(w\Sigma^*) - \sum_{u\in U} x_u^v P_{\sigma(n)}(w\Sigma^*)\right| + \left|\sum_{u\in U} x_u^v P_{\sigma(n)}(w\Sigma^*) - \sum_{u\in U} x_u^v P(w\Sigma^*)\right|$. Avec probabilité 1 tous les termes tendent vers 0 ; le premier et le quatrième d'après le lemme 1, le second est inférieur à $\epsilon_{\sigma(\phi(n))}$ et le troisième terme par hypothèse de convergence de $(x_u^{v,\sigma(\phi(n))})$. D'où $(x_u^v)_{u\in U}$ est une solution de $E(U, P)$. □

4.2 L'algorithme principal

L'algorithme principal se divise en trois parties ; dans la première nous calculons les états du PRFA, dans la seconde nous recherchons les transitions et à la fin nous construisons le PRFA en fonction des données obtenues.

Algorithme 1 DEES

`Entrée :` Un échantillon stochastique S_n de taille n.
`Sortie :` Un PRFA préfixe $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$.
$Q \leftarrow$ `calculeEtats` (S_n)
`renvoie construitPRFA` $(Q,$ `calculeTranstions` $(Q, S_n))$.

Soit $\mathrm{Prem}\,(P)$ l'ensemble des mots u tels que $u^{-1}P$ soit premier et qu'il n'existe pas de mot $v < u$ tel que $v^{-1}P = u^{-1}P$. On définit $\mathrm{Pm}\,(P)$ l'ensemble des mots u tels que $u \leq \max\left(\mathrm{Prem}\,(P)\right)$ et $u^{-1}P$ est défini.

Remarque 1. $\mathrm{Pm}\,(P)$ est un ensemble préfixiel.

Il est clair que $\mathrm{sol}\left(E\left(\mathrm{Pm}\,(P), P\right)\right) \neq \varnothing$ et que pour tout ensemble $U = \{u < v\}$ où $v < \max\left(\mathrm{Pm}\,(P)\right)$, alors $\mathrm{sol}\left(E\left(U, P\right)\right) = \varnothing$. Notre algorithme va renvoyer un automate dont l'ensemble d'état sera $\mathrm{Pm}\,(P)$. Comme nous nous intéressons seulement à un résultat d'identification à la limite, on se contentera d'un tel ensemble. En pratique l'algorithme que nous utilisons renvoie un automate dont l'ensemble d'états est $\mathrm{Pm}\,(P)$ défini récursivement comme l'ensemble des mots v tels que $v^{-1}P$ soit défini, et tels que $v^{-1}P \notin \mathrm{conv}\,\{u^{-1}P \mid u \in \mathrm{Pm}\,(P) \text{ et } u < v\}$.

Définition 3. *Un* PRFA préfixe *est un PRFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ dont l'ensemble d'état est un ensemble préfixiel de mots et tel que $\iota(\varepsilon) = 1$, $\forall (v, a, u) \in Q \times \Sigma \times Q$, $\varphi(v, a, u) \neq 0$ si et seulement si $va = u$ ou si $va \notin Q$. Un* PRFA préfixe associé à P *est un PRFA préfixe qui génère P (Cf. figure 1.a).*

Remarque 2. Soit $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ un PRFA préfixe associé à P. On remarque que pour tout $(v, a, u) \in Q \times \Sigma \times Q$, $\varphi(v, a, u) = P(va\Sigma^*)/P(v\Sigma^*)$ si $u = va$. En revanche, si $va \notin Q$, alors $\varphi(u, a, v)$ n'est pas déterminé par P.

On définit la relation sur deux ensembles finis de mots U et V par, $U \preceq V$ si et seulement si $U = \varnothing$ ou $\max(U) < \max(V)$ ou, $\max(U) = \max(V)$ et $U \setminus \max(U) \preceq V \setminus \max(U)$. On montre aisément que $\preceq$ est un ordre total.

Définition 4. *Le* PRFA préfixe réduit *de P est le PRFA préfixe associé à P, $T = \langle \Sigma, \mathrm{Pm}\,(P), \varphi^T, \iota, \tau \rangle$ tel que pour tout $va \in \mathrm{fr}\,(\mathrm{Pm}\,(P), P)$, $\varphi^T(v, a, u) \neq 0$ ssi u est un mot du plus petit sous-ensemble de* $\mathrm{Pm}\,(P)$ *(par rapport à $\preceq$) tel que $\exists (x_u^{va})_{u \in U}$ suite réelle positive, $(va)^{-1}P = \sum_{u \in U} x_u^{va} u^{-1}P$, (Cf. figure 1.a).*

Proposition 3. *Le PRFA préfixe réduit de P est unique et génère P.*

Démonstration. La preuve que le PRFA préfixe réduit génère P est donnée dans [19]. Il suffit de montrer l'unicité. Pour cela, il faut démontrer que les valeurs des transitions sont déterminées par P. Soit $va \in \mathrm{fr}\,(\mathrm{Pm}\,(P), P)$ et U le plus petit sous-ensemble de $\mathrm{Pm}\,(P)$ tel que $\exists (x_u^{va})_{u \in U}$ suite réelle positive,$(va)^{-1}P = \sum_{u \in U} x_u^{va} u^{-1}P$. Supposons que l'on ait $(va)^{-1} P = \sum_{u \in U} x_u^{va} u^{-1}P = \sum_{u \in U} y_u^{va} u^{-1}P$ et $x_{u_0}^{va} \neq y_{u_0}^{va}$ pour $u_0 \in U$. Soit $z_u = (x_u^{va} - y_u^{va}) / (y_{u_0}^{va} - x_{u_0}^{va})$ définie pour tout $u \in U \setminus \{u_0\}$. On a alors $u_0^{-1}P = \sum_{u \in U \setminus \{u_0\}} z_u u^{-1}P$. On a $(va)^{-1} P = \gamma \left(\sum_{u \in U} x_u u^{-1}P\right) + (1 - \gamma) \left(\sum_{u \in U \setminus \{u_0\}}^{m} z_u u^{-1}P\right)$. Lorsque $\gamma = 1$, tous les coefficients sont strictement positifs par hypothèse. Pour $\gamma = 0$ il y a au moins un coefficient négatif sans quoi cela contredirait l'hypothèse de minimalité. Soit $\gamma_0 \in [0, 1]$ le plus petit γ tel que tous les coefficients soient positifs. On a $\gamma_0 > 0$ et au moins un terme $\gamma_0 x_j + (1 - \gamma_0)\,\beta_j = 0$ ce qui contredit l'hypothèse de minimalité. □

Algorithme 2 `calculeEtats`

`Entrée` : Un échantillon stochastique S_n de taille n.
`Sortie` : un ensemble préfixiel de mots.
$Q \leftarrow \{\varepsilon\}$;
`tant que` $\mathrm{sol}\,(I\,(Q, S_n, \epsilon_n)) = \varnothing$ `et` $\mathrm{fr}\,(Q, P_n) \neq \varnothing$ `faire`
    $Q \leftarrow Q \cup \min \mathrm{fr}\,(Q, P_n)$
`retourne` Q

Proposition 4. *Avec une probabilité un, après un rang N, l'algorithme* `calculeEtats` *renvoie* $\mathrm{Pm}\,(P)$.

Démonstration. Soit $U \subsetneq \mathrm{Pm}\,(P)$ alors, d'après la remarque 1 et le lemme 8, avec une probabilité un, il existe un entier N tel que $\forall n > N$, `calculeEtats`(S_n) ne renvoie pas U. D'après la remarque 1 et le lemme 7, il existe un entier N tel que $\forall n > N$, `calculeEtats`(S_n) renvoie $\mathrm{Pm}\,(P)$. $\square$

Algorithme 3 `calculeTransitions`

`Entrée :` Un ensemble préfixiel Q, un échantillon stochastique S_n de taille n.
`Sortie :` une suite $(x_u^v)_{(v,u)\in \mathrm{fr}(Q,P_n)\times Q}$.
$C = \varnothing$, $V = \varnothing$,
`tant que` $V \neq \mathrm{fr}\,(Q, P_n)$ **`faire`**
  $v = \min\,(\mathrm{fr}\,(Q, P_n) \setminus V)$, $V \leftarrow V \cup \{v\}$, $U = \varnothing$,
  `tant que` $U \neq Q$ **`faire`**
    $u = \max\,(Q \setminus U)$, $U \leftarrow U \cup \{u\}$
    `Si` $\mathrm{sol}\,(I(Q, S_n, \epsilon_n) \cup C \cup \{x_u^v = 0\}) \neq \varnothing$ **`alors`** $C \leftarrow C \cup \{x_u^v = 0\}$
  `fin tant que`
`fin tant que.`
`Si` $\mathrm{sol}\,(I(Q, S_n, \epsilon_n) \cup C) \neq \varnothing$ **`alors`**
  `retourne` $(x_u^v) \in \mathrm{sol}\,(I(Q, S_n, \epsilon_n) \cup C)$
`sinon retourne la suite identiquement nulle.`

Pour prouver la convergence de `calculeTransitions` nous avons besoin du lemme suivant :

Lemme 9. *Soit U un ensemble fini de mots, soit $U' \subseteq U$ et soit $C = \{x_u^v = 0 \mid u \in U'\}$ un ensemble d'équations. On pose $E = E(U, P)$ et $I_n = I(U, S_n, \epsilon_n)$. Avec probabilité 1 à partir d'un certain rang,* $\mathrm{sol}\,(E \cup C) = \varnothing \Leftrightarrow \mathrm{sol}\,(I_n \cup C) = \varnothing$.

Démonstration. D'après le lemme 7, avec probabilité un, toute solution de E, en particulier celles qui satisfont C, est solution de I_n à partir d'un certain rang. Cela prouve l'implication de la droite vers la gauche. Si pour tout entier N, il existe $n \geq N$ tel que $I_n \cup C$ ait des solutions, des arguments analogues à ceux du lemme 8 montrent que $E \cup C$ admet aussi des solutions. $\square$

Proposition 5. *Avec probabilité un, après un certain rang* `DEES` *renvoie un PRFA dont le support est celui du PRFA préfixe réduit.*

Démonstration. Soit $T = \left\langle \Sigma, \mathrm{Pm}\,(P), \varphi^T, \iota^T, \tau^T \right\rangle$ le PRFA préfixe réduit associé à P. Alors il est clair en utilisant la proposition 4 et le lemme 9 avec probabilité 1 après un certain rang, l'invariant de boucle suivant est vérifié (dans la boucle interne) :

Soit $w \in \Sigma^*$ et $a \in \Sigma$ tels que $v = wa$, pour tout $u \in Q$, tel que $u \leq \max U$, $\varphi^T(w, a, u) = 0 \iff \{x_u^{wa} = 0\} \in C$.

En particulier en sortie des deux boucles imbriquées cette propriété est vraie pour tout couple de $\mathrm{fr}\left(\mathrm{Pm}\left(P\right), P\right) \times \mathrm{Pm}\left(P\right)$ puisque $Q = \mathrm{Pm}\left(P\right)$. Par construction (`construit_PRFA`), nous avons $\forall (wa, u) \in \mathrm{fr}\left(\mathrm{Pm}\left(P\right), P\right) \times \mathrm{Pm}\left(P\right)$, $\varphi^T(w, a, u) = 0 \Leftrightarrow \varphi(w, a, u)$ ce qui par signifie clairement que le support du PRFA A renvoyé par `DEES` sera égal au support du PRFA préfixe réduit qui génère P. □

Algorithme 4 `construitPRFA`

`Entrée :` Un ensemble prefixiel Q, un échantillon stochastique S_n de taille n,
une suite $(x_u^v)_{(v,u) \in \mathrm{fr}(Q, P_n) \times Q}$.
`Sortie :` un PRFA préfixe $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$.
`Pour tout` $(v, a, u) \in Q \times \Sigma \times Q$ `faire` $\varphi(v, a, u) = 0$.
`Pour tout` $(v, a, va) \in Q \times \Sigma \times Q$ `faire` $\varphi(v, a, va) = v^{-1} P_n(a\Sigma^*)$.
$\iota(\varepsilon) = 1$
`Pour tout` $u \in Q$ `faire` $\tau(u) = P_n(u) / P_n(u\Sigma^*)$.
`Si la suite` (x_u^v) `n'est pas identiquement nulle alors`
   `pour tout couple` $(va, u) \in \mathrm{fr}\left(Q, P_n\right) \times Q$ `où` $a \in \Sigma$
      `faire` $\varphi(v, a, u) = x_u^{va} v^{-1} P_n(a\Sigma^*)$
`Fin si`
`Si` A `est un PFA alors`
   `retourne` $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$
`sinon`
   `Soit` $(0)_{v,u}$ `la suite identiquement nulle`
   `retourne construitPRFA(`Préfixes $\left(S_n\right), S_n, (0)_{v,u}$`)`
`Fin si`

---

5 Convergence vers le PRFA préfixe réduit

Théorème 1. *Soit P un langage stochastique générable par un PRFA, soit A le PRFA préfixe réduit de P et soit Sun échantillon stochastique de P. Avec une probabilité de* 1*,* `DEES` *retourne à partir d'un certain rang un PRFA A_n dont la structure est A. Avec probabilité* 1*, les valeurs portées par les transitions des A_n convergent vers celles portées par les transitions de A.*

Démonstration. D'après la proposition 5, avec probabilité 1, il existe un N tel que $\forall n \geq N$, la suite des automates A_n construit par `DEES` possède le même support que A. Il est clair qu'avec probabilité 1, $\lim_{n \to \infty} v^{-1} P_n\left(a\Sigma^*\right) = v^{-1} P(a\Sigma^*)$ pour tout $va \in Q$. Soit $v \in \mathrm{fr}\left(\mathrm{Pm}\left(P\right), P\right)$ et $v^{-1}P = \sum_{u \in U} x_u^v u^{-1} P$ où U est l'ensemble des mots $u \in \mathrm{Pm}\left(P\right)$ et tel que la contrainte $\{x_u^v = 0\}$ n'appartiennent pas à C. Par la proposition 3, il est clair que $(x_u^v)_{u \in U}$ est une solution unique. Soit $\mathcal{W}$ un ensemble minimal de mots tel que $(x_u^v)_{u \in U}$ soit aussi l'unique solution du système $\forall w \in \mathcal{W}, v^{-1} P\left(w\Sigma^*\right) = \sum_{u \in U} x_u^v u^{-1} P(w\Sigma^*)$ avec les $x_u^v \geq 0$.

Soient les vecteurs $V = \left[v^{-1}P(w\Sigma^*)\right]_{w\in\mathcal{W}}$ et $X = [x_u^v]_{u\in U}$, la matrice $W = \left[u^{-1}P\left(w\Sigma^*\right)\right]_{u\in U,w\in\mathcal{W}}$. Il est clair que W est une matrice carrée inversible et on a $X = W^{-1}V$. Soient $V_n = \left[v^{-1}P_n(w\Sigma^*)\right]_{w\in W}$, $X_n = [x_u^{v,n}]_{u\in U}$, les matrices $W_n = \left[u^{-1}P_n\left(w\Sigma^*\right)\right]_{u\in U,w\in W}$ et E_n des vecteurs colonnes dont tous les éléments sont de valeur absolue inférieure à ϵ_n d'où $\|E_n\| = O\left(\epsilon_n\right)$. On a avec probabilité un après un certain rang, $V_n - W_nX_n = E_n$. Comme avec probabilité 1, $\lim_{n\to\infty} W_n = W$, et que le déterminant est une fonction continue, avec probabilité 1, il existe un rang après lequel W_n est inversible. On a donc : $W_n^{-1}V_n - X_n = W_n^{-1}E_n$ d'où $\left\|W_n^{-1}V_n - X_n\right\| \le \left\|W_n^{-1}\right\| O\left(\epsilon_n\right)$.

Considérons $W_n = W + U_n = W\left(1 + W^{-1}U_n\right)$, d'où nous obtenons que $W_n^{-1} = \left(1 + W^{-1}U_n\right)^{-1} W^{-1} = \left(1 - W^{-1}U_n - (W^{-1}U_n)^2 - \ldots\right) W^{-1}$ d'où $\left\|W^{-1} - W_n^{-1}\right\| = O\left(\|U_n\| \left\|W^{-1}\right\|^2\right)$. Il est clair d'après la proposition 2, le lemme 5 et le lemme 6 qu'avec probabilité 1, il existe un rang après lequel $\|V - V_n\| = O\left(\epsilon_n\right)$ et $\|W_n - W\| = O\left(\epsilon_n\right)$. On a donc :

$$\begin{aligned}
\|X - X_n\| &= \left\|W^{-1}V - W^{-1}V_n + W^{-1}V_n - W_n^{-1}V_n + W_n^{-1}E_n\right\| \\
&\le \left\|W^{-1}V - W^{-1}V_n\right\| + \left\|W^{-1}V_n - W_n^{-1}V_n\right\| + \left\|W_n^{-1}E_n\right\| \\
&\le \left(\left\|W^{-1}\right\| + \|V_n\| + \left\|W_n^{-1}\right\|\right) O\left(\epsilon_n\right) = O\left(\epsilon_n\right) \quad . \qquad \square
\end{aligned}$$

5.1 Complexité

La fonction `calculeEtats` est une boucle dont le nombre maximal d'entrées sera le nombre de préfixes de mots de S_n. À chaque étape, on doit résoudre un système d'inéquations linéaires ce qui est un problème polynomial dans la taille du système, lui-même polynomial dans la taille de S_n. La fonction `calculeTransitions` possède deux boucles imbriquées. La taille de la frontière d'un ensemble de lettre est polynomial dans la taille de cet ensemble, or Q possède une taille polynomiale en la taille de S_n(nombre de préfixes de S_n). On entrera donc un nombre polynomial de fois à l'intérieur des deux boucles. L'opération à l'intérieur des deux boucles est aussi une résolution de système d'équations. Donc `calculeTransitions` est une fonction polynomiale en la taille de S_n. La fonction `construitPRFA` fait un test pour savoir si l'automate renvoyé est un PFA. Ce test peut se réaliser en temps polynomial sur le nombre d'états [4]. Puis la construction est clairement polynomiale en fonction du nombre d'états si le test est vrai et sinon le nombre d'opérations est polynomial dans la taille de S_n.

Étant donné que l'algorithme que nous avons utilisé pour prouver le résultat de convergence à la limite possède une complexité plus élevée que l'algorithme que nous utilisons en pratique, nous ne détaillons pas plus sa complexité. En pratique l'utilisation de l'algorithme du simplexe nous permet de résoudre les systèmes d'inéquations en temps linéaire, ce qui lui donne, en moyenne, une complexité identique à celle des algorithmes d'inférence de PDFAs.

6 Identification à la limite de la classe des PRFAs rationnels

On veut maintenant montrer qu'il existe une méthode pour trouver la valeur exacte des transitions avec probabilité 1, si les arêtes portent des valeurs rationnelles. Dans [18] les auteurs utilisent une méthode basée sur les arbres de Stern-Broco pour identifier les transitions d'un PDFA cible. Nous utilisons ici la méthode des fractions continues.

6.1 Approximations d'un nombre réel par des fractions continues

Tout nombre réel peut être représenté par un développement en fractions continues (tous les résultats mentionnés ci-dessous peuvent être trouvés dans [24]). Si $a_0, \ldots, a_n$ sont des nombres entiers, on notera

$$[a_0, \ldots, a_n] = a_0 + 1/\left(a_1 + 1/\left(\cdots(1/a_n)\cdots\right)\right) \ .$$

Si x est un nombre réel positif, on définit les suites (x_n) et (a_n) par $x_0 = x$, $a_n = \lfloor x_n \rfloor$ et $x_{n+1} = 1/(x_n - a_n)$, tant bien sûr que ces termes sont définis. On définit alors deux suites (p_n) et (q_n) de nombres entiers par $p_0 = a_0$, $p_1 = a_1 a_0 + 1$, $p_{n+2} = a_{n+2}p_{n+1} + p_n$ et $q_0 = 1$, $q_1 = a_1$, $q_{n+2} = a_{n+2}q_{n+1} + q_n$ pour $n \geq 0$, tant que ces termes sont définis. On montre facilement que $[a_0, \ldots, a_n] = \frac{p_n}{q_n}$.

Lemme 10. *Si x est rationnel, alors il existe un entier N pour lequel $x = \frac{p_N}{q_N}$ et si $n < N$ alors $\left|x - \frac{p_n}{q_n}\right| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{q_n^2}$.*

Si $\frac{a}{b}$ est une fraction telle que $\left|\frac{a}{b} - x\right| < \frac{1}{2b^2}$, alors il existe M tel que $\frac{a}{b} = \frac{p_M}{q_M}$.

Pour tout entier A et tout rationnel x, il n'existe qu'un nombre fini de rationnels $\frac{p}{q}$ vérifiant $\left|x - \frac{p}{q}\right| \leq \frac{A}{q^2}$.

Lemme 11. *Soient (ϵ_n) une suite de réels positifs convergent vers 0, x un rationnel, (y_n) une suite de rationnels vérifiant $|x - y_n| \leq \epsilon_n$ à partir d'un certain rang. Soit $\left(\frac{p_{n,m}}{q_{n,m}}\right)$ les développements de y_n en fractions continues. Alors, il existe un entier N tel que pour tout $n \geq N$, il existe M tel que x est égal à $\frac{p_{n,M}}{q_{n,M}}$ et est la seule fraction $\frac{p_{n,m}}{q_{n,m}}$ qui vérifie $\left|y_n - \frac{p_{n,m}}{q_{n,m}}\right| \leq \epsilon_n \leq \frac{1}{q_{n,m}^2}$.*

Démonstration. Soit $x = \frac{a}{b}$, dès que $\epsilon_n \leq \frac{1}{2b^2}$ d'après le lemme 10, il existe m tel que $x = \frac{p_{n,m}}{q_{n,m}}$. D'autre part, supposons que l'on ait $\left|y_n - \frac{p_{n,m}}{q_{n,m}}\right| \leq \epsilon_n \leq \frac{1}{q_{n,m}^2}$. On en déduit que $\left|x - \frac{p_{n,m}}{q_{n,m}}\right| \leq |x - y_n| + \left|y_n - \frac{p_{n,m}}{q_{n,m}}\right| \leq 2\epsilon_n \leq \frac{2}{q_{n,m}^2}$. Il n'existe qu'un nombre fini de fractions $\frac{p_{n,m}}{q_{n,m}}$ vérifiant cette inégalité. On a aussi $\left|x - \frac{p_{n,m}}{q_{n,m}}\right| \leq 2\epsilon_n$. Donc, dès que ϵ_n est assez petit $\left(\epsilon_n < \frac{1}{2b^2}\right)$, on a $x = \frac{p_{n,m}}{q_{n,m}}$. Soit z_n la première fraction $\frac{p_{n,m}}{q_{n,m}}$, s'il en existe, qui vérifie $\left|y_n - \frac{p_{n,m}}{q_{n,m}}\right| \leq \epsilon_n \leq \frac{1}{q_{n,m}^2}$. Il existe un rang après lequel la suite (z_n) est constante égale à x. $\square$

Exemple 3. Si $g_n = \frac{1}{2} - \frac{1}{n}$ et $\epsilon_n = \frac{1}{n}$, on a $g_3 = \frac{1}{6}$, $g_4 = \frac{1}{4}$, $g_5 = \frac{3}{10}$, $g_6 = \frac{1}{3}$, $g_7 = \frac{5}{14}$. Le premier entier n pour lequel $\left| g_n - \frac{p_{n,m}}{q_{n,m}} \right| \leq \frac{1}{n} \leq \frac{1}{q^2}$ admet des solutions est $n = 4$. On a $z_4 = \frac{1}{4}$, $z_5 = \frac{1}{3}$, $z_6 = \frac{1}{3}$ et $z_7 = \frac{1}{2}$ à partir de $n = 7$.

Un PRFA sera dit *rationnel* si le PRFA préfixe réduit associé au langage stochastique qu'il génère ne possède que des valeurs rationnelles sur les transitions.

Théorème 2. *Il existe un algorithme qui pour tout un langage stochastique P générable par un PRFA rationnel, avec une probabilité de* 1*, renvoie le PRFA préfixe réduit associé à P.*

Démonstration. On considère φ la fonction de transition du PRFA préfixe réduit associé à P et φ^n la fonction transition renvoyé par `DEES` à l'étape n. On sait que $|\varphi^n(t) - \varphi(t)| = O(\epsilon_n)$. Soit $\epsilon_n = o(\epsilon'_n)$ et $\lim_{n \to \infty} \epsilon' = 0$. On sait que $|\varphi^n(t) - \varphi(t)| \leq \epsilon'_n$ à partir d'un certain rang. Il suffit alors de rechercher la première fraction continue $\frac{p_{n,m}}{q_{n,m}}$ qui vérifie $\left| \varphi^n(t) - \frac{p_{n,m}}{q_{n,m}} \right| \leq \epsilon'_n \leq \frac{1}{q^2}$ où $\varphi^n(t) = \frac{p}{q}$ et d'après le lemme 11, on sait qu'à partir d'un certain rang on aura $\frac{p_{n,m}}{q_{n,m}} = \varphi(t)$. □

7 Conclusion

Nous avons montré que l'ensemble des PRFAs à paramètres rationnels sont identifiables à la limite avec une probabilité de 1. Étant donné que les PRFAs sont plus expressifs que les PDFAs, nous pouvons espérer obtenir de meilleurs résultats pratiques. L'algorithme est implémenté et des expérimentations sont en cours.

Références

1. Paz, A. : Introduction to probabilistic automata. Academic Press, London (1971)
2. Casacuberta, F. : Some relations among stochastic finite state networks used in automatic speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 691–695
3. Abe, N., Warmuth, M. : On the computational complexity of approximating distributions by probabilistic automata. Machine Learning **9** (1992) 205–260
4. Dupont, P., Denis, F., Esposito, Y. : Links between probabilistic automata and hidden markov models : probability distributions, learning models and induction algorithms. Technical Report RR 2003-02, UCL (2003)
5. Bréhélin, L., Gascuel, O. : Modèles de Markov cachés et apprentissage de séquences. In : Le temps, l'espace et l'évolutif - Ecole thématique Document et Evolution. (2000)
6. Dempster, A., Laird, N.M., Rubin, D.B. : Maximum likelyhood from incomplete data via the em algorithm. Journal of the Royal Statistical Society **39** (1977) 1–38

7. Baldi, P., Brunak, S. : Bioinformatics : The Machine Learning Approach. MIT Press (1998)
8. Freitag, D., McCallum, A. : Information extraction with HMM structures learned by stochastic optimization. In : AAAI/IAAI. (2000) 584–589
9. Bergadano, F., Varricchio, S. : Learning behaviors of automata from multiplicity and equivalence queries. In : Italian Conference on Algorithms and Complexity. (1994)
10. Beimel, A., Bergadano, F., Bshouty, N.H., Kushilevitz, E., Varricchio, S. : On the applications of multiplicity automata in learning. In : IEEE Symposium on Foundations of Computer Science. (1996) 349–358
11. Beimel, A., Bergadano, F., Bshouty, N.H., Kushilevitz, E., Varricchio, S. : Learning functions represented as multiplicity automata. Journal of the ACM **47** (2000) 506–530
12. Valiant, L. : A theory of the learnable. Commun. ACM **27** (1984) 1134–1142
13. Angluin, D. : Learning regular sets from queries and counterexamples. Information and Computation **75** (1987) 87–106
14. Goldman, S.A., Mathias, H.D. : Teaching a smarter learner. Journal of Computer and System Sciences **52** (1996) 255–267
15. de la Higuera, C. : Characteristic sets for polynomial grammatical inference. Machine Learning **27** (1997) 125–137
16. 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
17. Carrasco, R.C., Oncina, J. : Learning deterministic regular grammars from stochastic samples in polynomial time. RAIRO (Theoretical Informatics and Applications) **33** (1999) 1–20
18. de la Higuera, C., Thollard, F. : Identification in the limit with probability one of stochastic deterministic finite automata. Volume 1891 of Lecture Notes in Artificial Intelligence., Springer (2000) 141–156
19. 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)
20. Denis, F., Lemay, A., Terlutte, A. : Residual Finite State Automata. Fundamenta Informaticae **51** (2002) 339–368
21. Denis, F., Esposito, Y. : Residual languages and probabilistic automata. In : to appear in Proceedings of ICALP2003. (2003)
22. Vapnik, V.N. : Statistical Learning Theory. John Wiley (1998)
23. Lugosi, G. : Pattern classification and learning theory. In : Principles of Nonparametric Learning. Springer (2002) 1–56
24. Hardy, G.H., Wright, E.M. : An introduction to the theory of numbers. Oxford University Press (1979)