DS 3

Option informatique, deuxième année

Julien Reichert

Partie 1 : Automates probabilistes

On fixe dans cet exercice un alphabet $\Sigma = \{0, 1\}$.

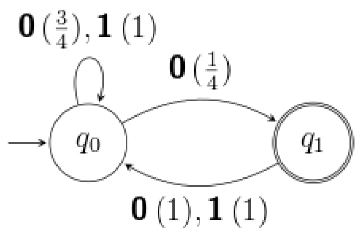
Un *automate probabiliste* sur l'alphabet Σ est un quadruplet $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$, où :

- Q est un ensemble fini non vide d'*états* ;
- $q_0 \in Q$ est appelé *état initial* ;
- $F \subseteq Q$ est un ensemble dont les éléments sont appelés *états finals* ;
- Pr: $Q \times \Sigma \times Q \to [0; 1]$ est une application appelée *fonction probabiliste de transition*, pour laquelle on suppose que pour tout $q \in Q$, pour tout $a \in \Sigma$, on a $\sum_{q' \in Q} \mathrm{Pr}(q, a, q') = 1$. On note $\mathrm{Pr}(q \xrightarrow{a} q')$ pour $\mathrm{Pr}(q, a, q')$.

Une *transition* est un triplet $(q, a, q') \in Q \times \Sigma \times Q$, noté $(q \xrightarrow{a} q')$, tel que $\mathrm{Pr}(q \xrightarrow{a} q') > 0$.

On représente un automate probabiliste de manière graphique, de façon similaire à la représentation des automates non-déterministes classiques : les états sont représentés par des cercles, l'état initial par une flèche arrivant sur le cercle correspondant, les états finals par des cercles doubles (ou une flèche sortant des cercles correspondants). La fonction probabiliste de transition est représentée par une flèche entre états, dont l'étiquette est cette fois $a(p)$ pour a la lettre et p la probabilité si elle est non nulle.

Ainsi, l'automate $\mathcal{A}_0 = (\{q_0, q_1\}, q_0, \{q_1\}, \mathrm{Pr})$ représenté ci-dessous :



a pour fonction probabiliste de transition Pr la fonction suivante (seules les valeurs non nulles sont mentionnées) :

q	α	q'	$\mathrm{Pr}(q \xrightarrow{\alpha} q')$
q_0	**0**	q_0	3/4
q_0	**0**	q_1	1/4
q_0	**1**	q_0	1
q_1	**0**	q_0	1
q_1	**1**	q_0	1

Étant donné un automate probabiliste $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ sur Σ, un *chemin* ρ est une suite finie de transitions

$q_{i_1} \overset{a_1}{\rightarrow} q_{i_2} \overset{a_2}{\rightarrow} \dots \overset{a_n}{\rightarrow} q_{i_{n+1}}$; on dit que ρ a pour *étiquette* le mot $a_1 a_2 \dots a_n \in \Sigma^*$, pour *état de départ* l'état $q_{i_1} \in Q$ et pour *état d'arrivée* l'état $q_{i_{n+1}} \in Q$. La *probabilité* de ρ, notée $\mathrm{Pr}(\rho)$, est définie comme $\Pi_{k=1}^{n} \mathrm{Pr}(q_{i_k} \overset{a_k}{\rightarrow} q_{i_{k+1}})$. Un état peut être vu comme un chemin de longueur nulle, la probabilité d'un chemin de longueur nulle est égale à 1. Un *chemin pour le mot* $u \in \Sigma^*$ est un chemin dont l'étiquette est u et l'état de départ est q_0. Ce chemin est *acceptant* si l'état d'arrivée est un état de F, *non-acceptant* sinon. La *probabilité d'un mot* $u \in \Sigma^*$, notée $\mathrm{Pr}(u)$, est par définition la somme des probabilités de tous les chemins acceptants pour le mot u.

Question 1 : Calculer les probabilités des mots ε (le mot vide), 0 et 010 pour l'automate $\mathcal{A}_0$.

Question 2 : Montrer, pour tout automate probabiliste $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ et tout mot $u \in \Sigma^*$, l'égalité suivante en utilisant une récurrence sur la longueur du mot u : $\mathrm{Pr}(u) = 1 - \Sigma_{\rho \text{ chemin non acceptant sur } u} \mathrm{Pr}(\rho)$.

Question 3 : On revient à l'automate probabiliste $\mathcal{A}_0$. Quels sont les mots u dont la probabilité $\mathrm{Pr}(u)$ pour $\mathcal{A}_0$ est égale à 0 ? Quels sont ceux dont la probabilité est égale à 1 ?

Question 4 : Proposer (sans justification) une expression rationnelle pour le langage des mots u dont la probabilité $\mathrm{Pr}(u)$ pour $\mathcal{A}_0$ est non nulle.

Question 5 : Montrer que pour tout automate probabiliste $\mathcal{A}$, il existe un automate non nécessairement déterministe $\mathcal{A}'$ qui accepte exactement les mots dont la probabilité pour $\mathcal{A}$ est non nulle.

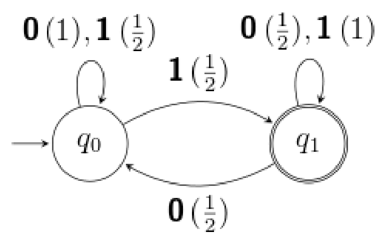
Question 6 : Appliquer la construction de la question précédente à l'automate $\mathcal{A}_0$ pour obtenir un automate non-déterministe qui accepte exactement les mots dont la probabilité pour $\mathcal{A}_{/}$ est non nulle. Déterminiser cet automate.

Soit $\mathcal{A} = (Q, q_0, F, \mathrm{Pr})$ un automate probabiliste sur Σ. Pour un réel $\eta \in [0; 1[$, le η-langage reconnu par $\mathcal{A}$, noté $L_\eta(\mathcal{A})$, est défini par : $L_\eta(\mathcal{A}) = \{u \in \Sigma^* \mid \mathrm{Pr}(u) > \eta\}$. On dit qu'un langage $L \subseteq \Sigma^*$ est *stochastique* s'il existe un automate probabiliste et un réel $\eta \in [0; 1[$ tel que L soit le η-langage reconnu par l'automate.

Question 7 : Démontrer que tout langage rationnel est stochastique.

Étant donné un mot $\alpha_1 \alpha_2 \dots \alpha_n$ sur l'alphabet $\{0, 1\}$, on dit que l'expression $\overline{0, \alpha_1 \alpha_2 \dots \alpha_n}^2$ est une écriture (finie) en base deux du nombre réel $\Sigma_{i=1}^{n} 2^{-i} \alpha_i$.

On considère maintenant l'automate $\mathcal{A}_1 = (\{q_0, q_1\}, q_0, \{q_1\}, \mathrm{Pr})$ ci-dessous :



Question 8 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(q_0 \overset{1}{\rightarrow} q_0 \overset{1}{\rightarrow} q_1 \overset{1}{\rightarrow} q_1 \overset{0}{\rightarrow} q_0 \overset{1}{\rightarrow} q_1)$ et en donner une écriture finie en base deux.

Question 9 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(10)$ et en donner une écriture finie en base deux.

Question 10 : Dans l'automate $\mathcal{A}_1$, calculer $\mathrm{Pr}(1101)$ et en donner une écriture finie en base deux.

Question 11 : Soit $u \in \{0, 1\}^*$ un mot arbitraire. Montrer que $\mathrm{Pr}(u)$ pour $\mathcal{A}_1$ admet une écriture finie en base deux, et en donner une expression. Prouver que cette écriture est correcte.

Question 12 : Soit $\eta \in [0; 1[$. Prouver l'égalité suivante :

$$L_\eta(\mathcal{A}_1) = \{\alpha_1 \alpha_2 \dots \alpha_n \in \{0, 1\}^* \mid \overline{0, \alpha_n \alpha_{n-1} \dots \alpha_1}^2 > \eta\}.$$

Question 13 : En déduire qu'il existe des langages stochastiques qui ne sont pas rationnels.

Partie 2 : Automates et fonctions

On utilise les notations usuelles du cours, ainsi que les notations $|w|$ pour représenter la longueur du mot w et $|w|_a$ pour représenter le nombre d'occurrences de la lettre a dans w. On considère dans cet exercice l'alphabet $\Sigma = \{a, b\}$. Soit f une application quelconque définie sur $\mathbb{N}$ et à valeurs dans $\mathbb{N}$. On note $L(f)$ l'ensemble des mots u appartenant à Σ^* vérifiant l'égalité $|u|_a = f(|u|_b)$.

Question 14 : On considère la fonction f_1 définie par : $\forall n \in \mathbb{N}, f_1(n) = 2$. Dessiner un automate reconnaissant le langage $L(f_1)$.

On considère la fonction $f_2(n) = n + 1$ modulo 2 (qui est aussi l'indicatrice de l'ensemble des entiers pairs).

Question 15 : Décrire $L(f_2)$ par une expression rationnelle de la forme $\alpha(bab+a+b)\beta$, où α et β sont des expressions rationnelles à déterminer. Justifier la réponse.

Question 16 : Dessiner un automate non nécessairement déterministe reconnaissant le langage décrit par l'expression rationnelle $bab + a + b$. Cet automate devra nécessairement posséder un seul état initial et un seul état final.

Question 17 : En s'appuyant sur l'expression rationnelle obtenue à la question 15, compléter l'automate obtenu à la question précédente pour obtenir un automate non déterministe reconnaissant le langage $L(f_2)$. Cet automate devra nécessairement posséder un seul état initial et un seul état final.

Question 18 : Déterminiser l'automate obtenu à la question précédente. On utilisera un algorithme vu en cours et on ne fera apparaître que les états accessibles depuis l'état initial.

Question 19 : Montrer que si f n'est pas majorée par une constante, alors $L(f)$ n'est pas rationnel.

Question 20 : On considère le langage $L_=$ sur Σ défini par $L_= = \{u \in \Sigma^*, |u|_a = |u|_b\}$. Le langage $L_=$ est-il rationnel ?

Question 21 : On considère le langage $L_\leq$ sur Σ défini par $L_\leq = \{u \in \Sigma^*, |u|_a \leq |u|_b\}$. Le langage $L_\leq$ est-il rationnel ? On utilisera le résultat de la question précédente.

Question 22 : On considère le langage $L_<$ sur Σ défini par $L_< = \{u \in \Sigma^*, |u|_a < |u|_b\}$. Le langage $L_<$ est-il rationnel ? On utilisera le résultat de la question précédente.

Question 23 : Montrer que la réciproque de la proposition énoncée dans la question 19 est fausse.

Indication : on pourra admettre que le langage des mots de la forme b^n où b est une lettre et n est un entier premier n'est pas rationnel.

Partie 3 : Homomorphismes de langages

Le but de cet exercice est de prouver qu'un homomorphisme de langage préserve la structure de langage régulier : l'image et l'image réciproque d'un langage régulier par un homomorphisme de langage sont des langages réguliers.

On suppose les définitions de base sur les automates, les langages et les expressions régulières connues. Soient deux alphabets Σ et Σ_2, un homomorphisme de langage h est une application de Σ^* dans Σ_2^* telle que $h(\varepsilon) = \varepsilon$ et pour $w_1, w_2 \in \Sigma^*$, on a $h(w_1 w_2) = h(w_1)h(w_2)$. Il est dit ε-libre si aucun mot non vide n'a pour image le mot vide.

L'image d'un langage $L \subseteq \Sigma^*$ par un homomorphisme h est $h(L) = \{h(w) \mid w \in L\} \subseteq \Sigma_2^*$, et l'image réciproque d'un langage $L \subseteq \Sigma_2^*$ est $h^{-1}(L) = \{w \mid h(w) \in L\} \subseteq \Sigma^*$.

Nous décomposons l'étude de la préservation de la structure de langage régulier par les homomorphismes de langage en plusieurs étapes qui reposent sur une restriction des homomorphismes de langage à un alphabet.

Soient deux alphabets Σ et Σ_2, soit une application $h : \Sigma \to \Sigma_2^*$, nous notons $\hat{h}$ l'extension (dite de Kleene) de h à Σ^* définie par $\hat{h}(\varepsilon) = \varepsilon$ et $\forall \sigma \in \Sigma, w \in \Sigma^*, \hat{h}(\sigma w) = h(\sigma)\hat{h}(w)$.

Question 24 : Montrer qu'on définit ainsi un homomorphisme de langage, qui est ε-libre si aucune lettre n'a pour image ε.

Question 25 : Montrer que l'extension de Kleene de la restriction à Σ d'un homomorphisme de langage est l'homomorphisme lui-même.

Pour montrer que l'image d'un langage régulier par un homomorphisme de langage est un langage régulier, nous exploitons la définition des langages réguliers sous la forme d'expressions régulières.

Un homomorphisme d'expression régulière s'appuyant sur une fonction h de Σ dans Σ_2^* est une application de domaine l'ensemble des expressions régulières d'alphabet Σ, notée $\tilde{h}$, par laquelle l'image de l'expression vide (à ne pas confondre avec le mot vide) est l'expression vide, l'image de l'expression ε est ε, l'image d'une lettre correspond à son image par h, et, pour deux expressions régulières e_1 et e_2, l'image de $e_1 + e_2$ est $\tilde{h}(e_1) + \tilde{h}(e_2)$, l'image de $e_1.e_2$ est $\tilde{h}(e_1).\tilde{h}(e_2)$ et l'image de e_1^* est $(\tilde{h}(e_1))^*$.

Question 26 : Soient les alphabets $\Sigma = \{a, b\}$ et $\Sigma_2 = \{0, 1\}$, soit $h : \Sigma \to \Sigma_2^*$ définie par $h(a) = 10$ et $h(b) = 0$, calculer $\tilde{h}(a(ba + aba)^*)$.

Question 27 : Montrer que l'image d'une expression régulière par un homomorphisme d'expression régulière est une expression régulière.

Question 28 : Avec les notations précédentes, montrer que $L(\tilde{h}(e)) = \hat{h}(L(e))$.

Question 29 : Montrer que l'image par un homomorphisme ε-libre de langage d'un langage régulier est un langage régulier.

La deuxième partie de l'énoncé (réciproque) a été retirée.