Oral Informatique ULC (Ulm Lyon Cachan) 2008

Nathanaël FIJALKOW

1 Exercice sur les langages (Gérardin Maxime)

Soit X un alphabet (fini), $\mathcal{L}$ un langage, notons $\sqrt{\mathcal{L}} = \{u \in X^*, u.u \in \mathcal{L}\}$.

1. Montrer que si $\mathcal{L}$ est rationnel, il en est de même de $\sqrt{\mathcal{L}}$.
2. On se donne un automate déterministe reconnaissant le premier. Donner un automate déterministe reconnaissant $\sqrt{\mathcal{L}}$.
3. Donner un (autre) automate non déterministe reconnaissant $\sqrt{\mathcal{L}}$, mais avec moins d'états.
4. Reprendre les questions précédentes avec $\sqrt[k]{\mathcal{L}} = \{u \in X^*, u^k \in \mathcal{L}\}$
5. Soit $\mathcal{L}$ rationnel, le langage $M = \bigcup_{k \in \mathbb{N}^*} \sqrt[k]{\mathcal{L}}$ est-il rationnel ?

Solution

1. Etant donné un automate déterministe reconnaissant $\mathcal{L}$, noté $\alpha = (\{1 \dots n\}, \{1\}, \{n\}, \delta)$, où 1 est le seul état initial, et n le seul état final, notons $\alpha_{p,q} = (\{1 \dots n\}, \{p\}, \{q\}, \delta)$, et $\mathcal{L}_{p,q}$ le langage qu'il reconnaît. Alors $u \in \sqrt{\mathcal{L}}$ équivaut à $\exists q, u \in \mathcal{L}_{1,q} \cap \mathcal{L}_{q,n}$. Donc $\sqrt{\mathcal{L}} = \bigcup_{1 \leqslant p \leqslant n} (\mathcal{L}_{1,q} \cap \mathcal{L}_{q,n})$, et $\sqrt{\mathcal{L}}$ est rationnel.
2. Construisons un automate reconnaissant $\sqrt{\mathcal{L}}$ à partir de α.
   Soit $\beta = (\{1 \dots n\}^n, \{(1 \dots n)\}, G, \bullet)$ où depuis l'état $(e_1 \dots e_n)$ et la lettre a, $(e_1 \dots e_n) \bullet a = (e_1.a \dots e_n.a)$, ce qui permet de suivre les chemins parcourus en lisant le mot u depuis chaque état, et $G = \{(e_1 \dots e_n), e_{e_1} \in F\}$, on choisit les états i pour lesquels le chemin parcouru depuis 1 en lisant le mot u finit sur un état e_1, tel que le chemin parcouru depuis e_1 en lisant le mot u finit sur un état final. Cet automate est déterministe et reconnaît $\sqrt{\mathcal{L}}$.
3. Soit $\chi = (\{(p,q,r)_{1 \leqslant p,q,r \leqslant n}\}, \{(1,e,e)_{1 \leqslant e \leqslant n}\}, H, \bullet)$ où depuis l'état (p,q,r) et la lettre a, $(p,q,r) \bullet a = (p.a, q.a, r)$ et $H = \{(e,f,e), f \in F\}$. Même principe que précédemment, le troisième état sert "d'indicateur".

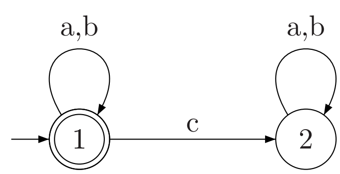
4. De même, $\sqrt[k]{\mathcal{L}} = \bigcup\limits_{1\leqslant p\leqslant n} \bigcap\limits_{0\leqslant p\leqslant k-1} \mathcal{L}_{q_p,q_{p+1}}$, avec $q_0 = 1$, ce qui montre que $\sqrt[k]{\mathcal{L}}$ est rationnel. On construit l'automate β de même, si ce n'est l'ensemble G qui devient $G_k = \{(e_1 \dots e_n), e_{\ddots_{e_1}} \in F\}$, avec k imbrications d'indices.

5. Construisons un automate reconnaissant M, à savoir β_∞. Seul G est modifié, et devient $G_\infty = \bigcup\limits_{k\in\mathbb{N}^*} G_k$.

2 Les automates de Büchi (Fijalkow Nathanaël)

Il s'agit de définir un nouvel objet, l'automate de Büchi. Contrairement aux automates classiques, les automates de Büchi servent à reconnaître des mots infinis. Nous noterons A^ω l'ensemble des mots de longueur infinie sur l'alphabet fini A. Un automate de Büchi est un quadruplet (Q, Δ, I, F), où Q est l'ensemble de ses états, Δ une partie de $Q \times A \times Q$ représentant les transitions, I l'ensemble des états initiaux, et F l'ensemble des états répétés. Un mot sera dit reconnu par un automate de Büchi lorsque une infinité de ses préfixes est reconnue par l'automate au sens classique, ou plus formellement, en notant $\{q_n, n \in \mathbb{N}\}$ l'ensemble des états visités au cours de la lecture du mot u, lorsque $\{k, q_k \in F\}$ est infini, ie l'automate se retrouve une infinité de fois dans un état répété en lisant le mot u. La distinction déterministe / non déterministe se définit de même que dans le cas classique.

Question 1 a) Quel est le langage des mots reconnus par l'automate suivant :



b) Dessiner un automate reconnaissant le langage des mots possédant une infinité de a (sur l'alphabet $\{a, b\}$).

c) Dessiner un automate reconnaissant le langage, noté K des mots possédant un nombre fini de b (sur l'alphabet $\{a, b\}$).

Question 2 On s'intéresse aux propriétés de clôture de l'ensemble des langages reconnus par un automate de Büchi.

a) Montrer la stabilité par réunion.

b) Montrer la stabilité par intersection.

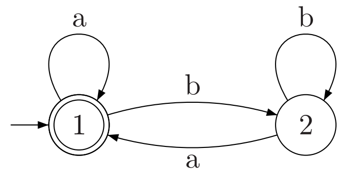
c) Discuter la stabilité par passage au complémentaire.

Question 3 Montrer qu'on ne peut pas toujours déterminiser un automate de Büchi. Discuter.

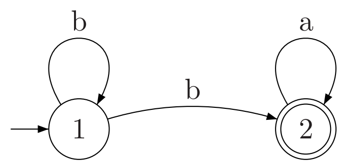
Solution

1 a) $(a|b)^*$, puisqu'un mot contenant un c ne peut rester qu'un nombre fini de fois sur l'état 1.

b) On peut proposer :



c) On peut proposer :



2 a) La construction du produit cartésien de deux automates convient : soit A_1 et A_2 deux automates, on définit $A_1 \oplus A_2 = (Q_1 \times Q_2, \Delta, I_1 \times I_2, F_1 \times Q_2 \cup Q_1 \times F_2)$, où $((p,p'), a, (q,q')) \in \Delta$ dès que $(p, a, q) \in \Delta_1$ et $(p', a, q') \in \Delta_2$. Alors u est reconnu par ce nouvel automate ssi, en notant $\{(p_n, q_n), n \in \mathbb{N}\}$ l'ensemble des états visités au cours de sa lecture, $\{k, (p_k, q_k) \in F_1 \times Q_2 \cup Q_1 \times F_2\}$ est infini, ce qui équivaut à ce que $\{k, (p_k, q_k) \in F_1 \times Q_2\}$ ou $\{k, (p_k, q_k) \in Q_1 \times F_2\}$ soit infini, ce qui équivaut encore à $\{k, p_k \in F_1\}$ ou $\{k, q_k \in F_2\}$ est infini, ie u reconnu par A_1 ou A_2.

b) La construction du produit cartésien de deux automates ne convient plus : en effet le fait que $\{k, p_k \in F_1\}$ et $\{k, q_k \in F_2\}$ soient infinis est une condition nécessaire mais non suffisante pour affirmer que $\{k, (p_k, q_k) \in F_1 \times F_2\}$ est infini. Il se peut, par exemple, que tous les préfixes impairs soient reconnus (au sens classique) par l'automate A_1, et que tous les préfixes pairs soient reconnus par l'automate A_2. Une idée est de conserver de l'information sur les états de l'automate, quitte à en augmenter le nombre. Ajoutons un bit, et prenons pour ensemble d'état $Q_1 \times Q_2 \times \{0,1\}$. Intuitivement, un état sera affecté du bit 0 si le dernier état répété rencontré appartenait à F_1, et 1 sinon. Prenons pour états initiaux $I_1 \times I_2 \times \{0\}$, et pour états finaux $F_1 \times Q_2 \times \{1\}$ (les esthètes choisiront la réunion avec le terme symétrique $Q_1 \times F_2 \times \{0\}$). Pour Δ, soit $(p, a, q) \in \Delta_1$ et $(p', a, q') \in \Delta_2$, si $q \in F_1$ alors $((p, p', x), a, (q, q', 0)) \in \Delta$, symétriquement si $q' \in F_2$ alors $((p, p', x), a, (q, q', 1)) \in \Delta$, et sinon $((p, p', x), a, (q, q', x)) \in \Delta$. Avec les notations précédentes, u est reconnu par ce nouvel automate ssi $\{k, (p_k, q_k, x_k) \in F_1 \times Q_2 \times \{1\}\}$ est infini. Ceci implique d'abord que $\{k, p_k \in F_1\}$ soit infini, puis, en remarquant qu'à chaque fois qu'on croise un état de $F_1 \times Q_2 \times \{1\}$, alors le dernier état répété croisé appartenait à F_2, donc $\{k, q_k \in F_2\}$ est infini, et u est reconnu à la fois par A_1 et A_2.

c) Partons d'un automate quelconque, et tentons de construire un automate reconnaissant son complémentaire. Une premiére remarque, la construction classique (compléter l'automate, inverser les états finaux et non finaux) ne donne pas le résultat. La premiére étape, elle, ne pose pas de difficultés, compléter un automate se fait comme dans le cas classique, et convient. L'idée est qu'un mot n'est pas reconnu si à partir d'un moment, il n'atteint plus d'états de F, et donc qu'il peut se promener librement sur l'automate Q auquel on a supprimé tous les états répétés. Reprenant l'idée de question 1 c) : complétons l'automate par l'état puit 0, notons $\widetilde{Q}$ le répliqué de Q, et choisissons $Q \cup (\widetilde{Q - F})$ pour ensemble d'états. Prenons pour états initiaux I, et pour états répétés $\widetilde{Q - F} \cup \{0\}$. Notons Γ l'ensemble des transitions, c'est la réunion de Δ et de l'ensemble ainsi construit : soit $(p, a, q) \in \Delta$ telles que $q \notin F$ alors $(p, a, \widetilde{q}) \in \Gamma$ et si de plus $p \notin F$ alors $(\widetilde{p}, a, \widetilde{q}) \in \Gamma$. L'état 0 ne mène que vers lui-même, pour toutes les lettres de l'alphabet. Un mot est reconnu s'il monte dans $\widetilde{Q - F}$ et qu'il s'y plaît, ou s'il tombe dans le puit. Cette construction ne convient que si l'automate initial était déterministe (le nouveau ne l'est d'ailleurs généralement plus).

3 Par exemple, montrons que le langage K des mots ne comportant qu'un nombre fini de b, dont une expression rationnelle est $(a|b)^* a^\omega$ ne peut être reconnu par un automate de Büchi déterministe. ...

3 Le jeu de Nim (Henri Sébastien)

Il s'agit d'étudier le jeu de Nim : on dispose de m vases, dont chacun contient un certain nombre de fruits (nombre noté respectivement $x_1, \ldots, x_m$). Un tour de jeu consiste à prendre un nombre non nul de fruits dans un seul vase. Le joueur qui gagne est celui qui prend le dernier fruit. On note G la fonction qui à $(x_1, \ldots, x_m)$ associe a où a est défini comme ceci : si $a[i]$ est le i-ème bit dans l'écriture binaire de a, alors $a[i] = (\sum_{l=1}^{m} x_l[i]) \bmod 2$, où $x_l[i]$ désigne le i-ème bit de x_l dans l'écriture binaire.

Question 1 Calculer $G(4, 1, 9, 10)$.

Question 2 Il s'agit de montrer que le joueur dont c'est le tour gagne la partie si et seulement si $G(x_1, \ldots, x_m) \neq 0$, c'est-à-dire que si $G(x_1, \ldots, x_m) \neq 0$, alors il existe une stratégie gagnante, et réciproquement, si $G(x_1, \ldots, x_m) = 0$, l'adversaire a une stratégie gagnante. On note Y_0 la position où

$G(x_1, \ldots, x_m) = 0$, Y_1 où $G(x_1, \ldots, x_m) \neq 0$.

a) Montrer que si le jeu est en position Y_0, quelque soit le coup joué, on passe en position Y_1.

b) Réciproquement, montrer que si le jeu est en position Y_1, alors il existe un coup tel que le jeu passe en position Y_0.

c) Conclure, et donner une stratégie gagnante si le jeu est en position Y_1.

Question 3 Ecrire un programme qui joue la stratégie gagnante à partir d'une position donnée (le cas échéant).

SOLUTION

1 Présentons sous forme d'un tableau :

Vase	x_i	Ecriture binaire
1	4	100
2	1	1
3	9	1001
4	10	1010
	a	0110

Donc $G(4, 1, 9, 10) = 6$

2 a) Supposons que l'on soit dans la position Y_0, et qu'un joueur retire un certain nombre de fruits du vase i. Alors l'un des bits de x_i est modifié, considérons donc le calcul effectué dans le bit correspondant $G(x_1, \ldots, x_m)[i]$, seul a été changé ce bit, ce qui modifie la parité, et $G(x_1, \ldots, x_m)[i] = 1$, donc $G(x_1, \ldots, x_m) \neq 0$

b) Supposons que l'on soit dans la position Y_1, et notons i le vase contenant le plus de fruits. L'objectif est de modifier chacun des bits qui sont à 1 dans $G(x_1, \ldots, x_m)$, et de ne pas toucher aux autres. Notons $(i_1, \ldots, i_k)$ ces colonnes (représentation en tableau). Considérons le nombre y tel que $y[i]$ vaut 1 si $i \in (i_1, \ldots, i_k)$, 0 sinon. $y \leqslant x_i$, un coup permettant de se placer en position Y_0 consiste donc à retirer y fruits du vase i.

c) Il suffit de faire la synthése de ce qui a été dit. Le jeu se finit en un nombre fini d'étapes (le nombre total de fruit décroît strictement), et la position $(0, \ldots, 0)$ appartient à Y_0, donc est atteinte lors du tour de l'adversaire, c'est donc bien que l'on a remporté la partie.

3 La question 2 b) fournit un algorithme de calcul d'un coup satisfaisant.

4 Complexité de la déterminisation (Lupu Titus)

Question 1 On dispose d'un automate déterministe avec n états. Montrer qu'il existe un automate déterministe reconnaissant le même langage avec au plus 2^n états.

Question 2 a) On prend $A = \{a, b\}$. Donner un automate déterministe reconnaissant A^*aA^2.

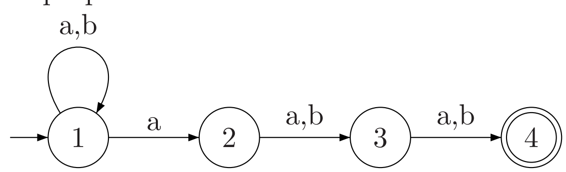
b) Donner un automate non déterministe reconnaissant $L_n = A^*aA^{n-2}$. Montrer que l'algorithme de déterminisation conduit à un automate à 2^{n-1} états.

c) Montrer que tout automate déterministe reconnaissant L_n a au moins 2^{n-1} états.

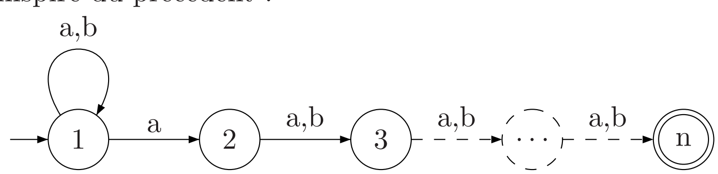
Solution

1 Il s'agit de présenter l'algorithme de déterminisation au programme, l'algorithme des parties. Soit (Q, Δ, I, F) un automate, on le déterminise comme suit : construisons l'automate $(2^Q, \Gamma, \widetilde{I}, \widetilde{F}$, où 2^Q est l'ensemble des parties de Q, $\widetilde{I} = \cup_{i \in I}\{i\}$, $\widetilde{F} = \{A \in 2^Q, A \cap F \neq \emptyset\}$, et $(X, a, Y) \in \Gamma$ dès qu'il existe $x \in X$ et $y \in Y$ tel que $(x, a, y) \in \Delta$. Ces deux automates reconnaissent le même chemin, puisqu'ils parcourent les mêmes chemins à la lecture du même mot. Le second est déterministe, et posséde 2^n états (au plus, éventuellement moins en l'émondant).

2 a) On peut proposer :



b) On s'inspire du précédent :



Montrons que l'automate déterminisé par l'algorithme des parties posséde 2^{n-1} états. En effet, chaque partie de $\{1 \dots n\}$ contenant 1 est un état utile de l'automate, il y en a 2^{n-1}.

c) L_n est l'ensemble des mots dont la (n-1) ième lettre avant la fin est un a. Considérons un automate déterministe reconnaisssant L_n. Considérons l'application qui à tout mot de taille $n-1$ associe l'état dans lequel se trouve l'automate en lisant ce mot. Montrons l'injectivité de cette application : considérons deux mots u et v qui conduisent au même état. Supposons $u \neq v$, on factorise $u = xay$ et $v = xbz$ avec $|y| = |z| = k \leqslant n-2$. Le mot ua^{n-2-k} est dans L_n, mais va^{n-2-k} ne l'est pas, or ils ménent au même état, contradiction. Donc l'application est injective, et l'automate posséde au moins 2^{n-1} états, le cardinal de l'ensemble des mots de longueur $n-1$ sur l'alphabet $\{a, b\}$.