Langages formels, calculabilité et complexité

Magistère MMFAI

Examen du 21 mai 2001

durée 3 heures

Les documents sont autorisés.

Toutes les réponses devront être correctement justifiées.

Il va sans dire que la rigueur des raisonnements, la clarté des explications, mais aussi la qualité de la présentation influeront sensiblement sur la note.

Le barème sur 35 n'est donné qu'à titre indicatif et pourra être modifié.

1 Automates boustrophédons (10 points)

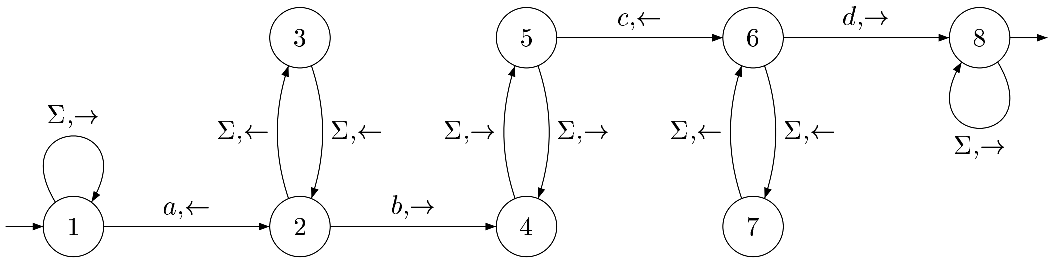
Un automate boustrophédon (automate à double sens) est un automate fini pouvant changer son sens de lecture à tout moment. On peut aussi dire qu'il s'agit d'une machine de Turing ayant simplement une bande d'entrée (sur laquelle elle ne peut pas écrire) et ni bandes de travail ni bande de sortie.

Formellement, un automate boustrophédon est un quintuplet $\mathcal{A} = (Q,\Sigma,T,I,F)$ où Q est l'ensemble fini d'états, I et F sont les états initiaux et finaux, Σ est l'alphabet et $T \subseteq Q \times \Sigma \times Q \times \{\leftarrow,\rightarrow\}$.

Un pas de calcul est de la forme $(p,i) \xrightarrow{a} (q,j)$ avec $(p,a,q,\rightarrow) \in T$ et $j = i+1$ ou $(p,a,q,\leftarrow) \in T$ et $j = i-1$.

Un calcul sur un mot $u = u_1 \cdots u_n$ est une suite $(q_0,i_0) \xrightarrow{u_{i_0}} (q_1,i_1), \xrightarrow{u_{i_1}} \ldots (p_k,i_k)$. Le calcul est acceptant si $q_0 \in I$, $i_0 = 1$, $q_k \in F$ et $i_k = n+1$.

On considère l'automate boustrophédon $\mathcal{B}$ défini par la figure



a) Donner une expression rationnelle pour le langage reconnu par $\mathcal{B}$.

b) Pour chaque entier n, donner un langage rationnel L_n reconnu par un automate déterministe classique ayant n états et ne pouvant pas être reconnu par un automate boustrophédon avec moins de n états.

c) Montrer que tout mot accepté par un automate boustrophédon $\mathcal{A}$ peut l'être par un calcul ne passant pas deux fois par la même configuration.

d) (5 points) Étant donné un automate boustrophédon $\mathcal{A}$, construire un automate non déterministe classique reconnaissant le même langage.

e) Question subsidiaire hors barème : Les automates boustrophédons permettent-ils de reconnaitre certains langages de façon exponentiellement plus concise que les automates non déterministes classiques ?

2 Automates à un pic (12 points)

Un automate à pile est dit à un pic si pour tout calcul valide, la taille de la pile n'augmente plus une fois qu'elle a diminué. La taille de la pile peut donc augmenter (au sens large) pendant une première partie du calcul, puis elle ne fait que diminuer (au sens large).

Un langage est à un pic s'il existe un automate à pile à un pic qui accepte ce langage par pile vide.

a) Montrer que le langage $L = \{a^n b^n \mid n > 0\} \cup \{a^n b^{2n} \mid n > 0\}$ est un langage linéaire.

b) Montrer que L est un langage à un pic.

c) Montrer que tout langage linéaire est un langage à un pic.

d) Montrer que le langage $K = \{a^{i_1} b a^{i_2} b \cdots a^{i_n} b \mid n > 0 \text{ et } \exists j, i_j \neq j\}$ est un langage à un pic.

e) Trouver une grammaire linéaire qui engendre K.

f) soit $\mathcal{A} = (Q,\Sigma,Z,T,(q_0,z_0))$ un automate à pile arbitraire, $q \in Q$ un état de $\mathcal{A}$ et $z \in Z$. Montrer que l'ensemble des mots w tels qu'il existe un calcul de $\mathcal{A}$ sur w à partir de (q,z) dans lequel la taille de la pile ne fait que diminuer (au sens large) est un langage rationnel.

g) (5 points) Montrer que tout langage à un pic est un langage linéaire.

3 Complexité (13 points)

On rappelle que le problème Circuit-Value (CV) est le suivant :

Donnée : Un circuit booléen défini par la suite de ses portes : $\mathrm{porte}_1 ; \mathrm{porte}_2 ; \cdots ; \mathrm{porte}_n$ où chaque porte a l'une des formes suivantes : $\wedge i,j$ ou $\vee i,j$ ou $\neg i$ ou $\top$ (vrai) ou $\perp$ (faux). Les entiers i et j sont codés en binaire et désignent les numéros des portes connectées à la porte considérée. La porte 1 est la porte de sortie du circuit.
Question : Est-ce que la porte de sortie prend la valeur VRAI?

Le problème Monotone-Circuit-Value (MCV) est le même que CV sauf que le circuit ne comporte pas de portes « non ».

a) (3 points) Montrer que le problème CV se réduit à MCV. Que peut-on en déduire?

Une machine de Turing alternante est une machine de Turing M dont l'ensemble Q des états est partitionné en deux ensembles $Q_\wedge$ et $Q_\vee$ représentant les états « et » et les états « ou » de M.

Pour chaque mot x, on considère l'arbre $T(x)$ des calculs de M sur x. La racine de l'arbre est le couple $(c_i(x),0)$ où $c_i(x)$ est la configuration initiale de la machine M sur le mot x. Si c est une configuration de M accessible à partir de $c_i(x)$ en k transitions, alors (c,k) est un nœud de l'arbre de profondeur k. Les fils d'un nœud (c,k) sont les nœuds $(c',k+1)$ tels que c' s'obtient à partir de c en exécutant une transition. Les feuilles de l'arbre sont les couples (c,k) dont la configuration c est acceptante (état OK) ou rejetante (état KO).

Une feuille est ultimement acceptante si son état est OK. Un nœud interne (c,k) associé à un état « et » (respectivement « ou ») est ultimement acceptant si et seulement si tous ses fils (respectivement au moins un de ses fils) sont ultimement acceptants. L'arbre $T(x)$ est acceptant si sa racine est ultimement acceptante.

Le langage $L(M)$ accepté par M est l'ensemble des mots x dont l'arbre des calculs $T(x)$ est acceptant.

On note **AL** l'ensemble des langages acceptés par une machine de Turing alternante utilisant un espace de travail logarithmique par rapport à la taille de sa donnée.

b) (4 points) Montrer que MCV est dans la classe **AL**.

c) (5 points) Montrer que le problème MCV est **AL**-dur.
Indication : Donner une réduction logarithmique qui, à partir d'une machine de Turing alternante M travaillant en espace logarithmique et d'une donnée x de M, construit un circuit booléen monotone évaluant si l'arbre des calculs $T(x)$ est acceptant. Les portes du circuit sont associées aux nœuds (c,k) de $T(x)$.

d) Montrer que $\mathbf{AL} = \mathbf{P}$.