INF105
Contrôle de connaissances

Théorie des langages

7 février 2017

Consignes.

Les exercices sont totalement indépendants. Ils pourront être traités dans un ordre quelconque, mais on demande de faire apparaître de façon très visible dans les copies où commence chaque exercice.

Le sujet étant long pour le temps imparti, il ne sera pas nécessaire de traiter toutes les questions pour obtenir la totalité des points.

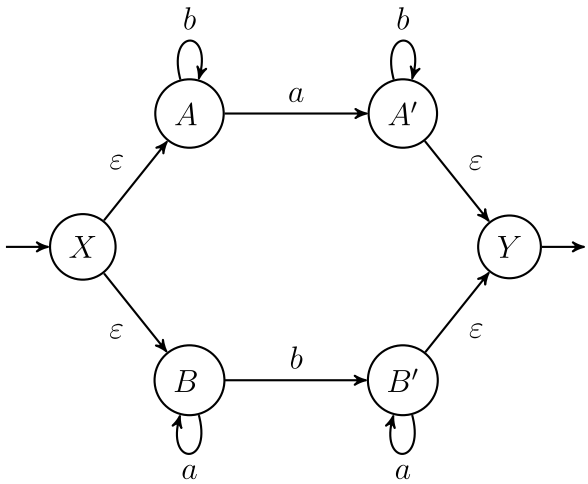
L'usage de tous les documents (notes de cours manuscrites ou imprimées, feuilles d'exercices, livres) est autorisé.

L'usage des appareils électroniques est interdit.

Durée : 1h30
Barème *indicatif* : 8 points par exercices
Cet énoncé comporte 4 pages (page de garde incluse)

Exercice 1.

On considère l'automate fini M sur l'alphabet $\Sigma = \{a, b\}$ représenté par la figure suivante :



(0) De quelle sorte d'automate s'agit-il ? (Autrement dit : est-il déterministe ou non ? avec transitions spontanées ou non ?)

(1a) Décrire brièvement, en français, le langage L reconnu (=accepté) par l'automate M, puis donner une expression rationnelle qui le dénote. (On pourra préférer traiter la question (1b) d'abord.)

(1b) Pour chacun des mots suivants, dire s'ils sont dans L ou non : ε, a, b, ab, aa, aab, $aabb$, $abab$, $ababa$. (Note : il est recommandé de réutiliser ces mots pour vérifier rapidement les réponses aux questions suivantes et ainsi détecter d'éventuelles erreurs lors des transformations des automates.)

(2) Éliminer les transitions spontanées de l'automate M. (On supprimera les états devenus inutiles.) On appellera M_2 l'automate obtenu.

(3) Déterminiser l'automate M_2 obtenu en (2), si nécessaire. (On demande un automate déterministe complet.) On appellera M_3 l'automate déterminisé.

Pour simplifier le travail du correcteur, on demande de représenter M_3 de sorte que les transitions étiquetées par a soient, dans la mesure du possible, horizontales de la gauche vers la droite, et celles étiquetées par b, verticales du haut vers le bas.

(4) Minimiser l'automate M_3 obtenu en (3), si nécessaire (justifier).

(5) Donner un automate (de n'importe quelle sorte) qui reconnaît le langage $\overline{L} = \Sigma^* \setminus L$ complémentaire de L.

(6) Décrire brièvement, en français, ce langage complémentaire $\overline{L}$.

(7) (Question bonus, plus longue, à ne traiter qu'en dernier.) Calculer une expression rationnelle qui dénote ce langage complémentaire $\overline{L}$. (Ne pas hésiter à introduire des notations intermédiaires.)

Exercice 2.

Soit Σ un alphabet (fini, non vide) fixé. Les questions suivantes sont indépendantes (mais on remarquera leur parallélisme). Ne pas hésiter à décrire les algorithmes de façon succincte et informelle.

(1) Expliquer, au moyen des résultats vus en cours, pourquoi il existe un algorithme A_1 qui, étant donnée une expression rationnelle r sur Σ, décide si le langage L_r dénoté par r est différent du langage Σ^* de tous les mots sur Σ. (Autrement dit, l'algorithme A_1 doit prendre en entrée une expression rationnelle r, terminer en temps fini, et répondre « vrai » s'il existe un mot $w \in \Sigma^*$ tel que $w \notin L_r$ et « faux » si $L_r = \Sigma^*$. On ne demande pas que l'algorithme soit efficace.)

(2) Expliquer pourquoi il existe un algorithme A_2 qui, étant donnée une grammaire hors contexte G sur Σ, « semi-décide » si le langage L_G engendré par G est différent du langage Σ^* de tous les mots. (« Semi-décider » signifie que l'algorithme A_2 doit prendre en entrée une grammaire hors contexte G, terminer en temps fini en répondant « vrai » s'il existe un mot $w \in \Sigma^*$ tel que $w \notin L_G$, et ne pas terminer [1] si $L_G = \Sigma^*$.) Indication : on peut tester tous les mots possibles.

(3) Expliquer pourquoi il existe un algorithme A_3 comme suit : on lui fournit en entrée un algorithme T qui décide un langage $L_T \subseteq \Sigma^*$ (c'est-à-dire que T termine toujours en temps fini quand on lui présente un mot sur Σ, et répond « vrai » ou « faux », et L_T est le langage des mots sur lesquels il répond « vrai »), et l'algorithme A_3 doit semi-décider si L_T est différent de Σ^*. (C'est-à-dire que A_3 doit terminer en répondant « vrai » s'il existe un mot $w \in \Sigma^*$ tel que $w \notin L_T$, et ne pas terminer si $L_T = \Sigma^*$.) Indication : la même approche permet de traiter les questions (2) et (3).

(4) Expliquer pourquoi il *n'existe pas* d'algorithme A_4 qui, dans les mêmes conditions que A_3, décide (au lieu de seulement semi-décider) si L_T est différent de Σ^*. (C'est-à-dire que A_4 est censé terminer toujours, et répondre « vrai » s'il existe un mot $w \in \Sigma^*$ tel que $w \notin L_T$, et « faux » si $L_T = \Sigma^*$.) Indication : expliquer comment on pourrait utiliser un tel A_4 pour résoudre le problème de l'arrêt, en cherchant à fabriquer un T qui rejette un mot précisément si un programme donné s'arrête.

Exercice 3.

On considère la grammaire hors-contexte G d'axiome S et de nonterminaux $N = \{S, T\}$ sur l'alphabet $\Sigma = \{a, b, c\}$ donnée par

$$\begin{aligned} S &\rightarrow TS \mid \varepsilon \\ T &\rightarrow aSbSc \end{aligned}$$

On notera $L(S) = L_G$ le langage qu'elle engendre, et $L(T)$ le langage des mots

1. On peut admettre qu'il termine parfois en répondant « faux », mais ce ne sera pas utile.

qui dérivent de T (c'est-à-dire, si on préfère, le langage engendré par la grammaire identique à G mais ayant T pour axiome).

(0) Donner quelques exemples de mots de $L(S)$ et de $L(T)$ (au moins deux de chaque).

(1) Expliquer brièvement pourquoi $L(S)$ est l'ensemble des mots de la forme $u_1 \cdots u_k$ avec $u_i \in L(T)$, et pourquoi $L(T)$ est l'ensemble des mots de la forme $awbw'c$ avec $w, w' \in L(S)$. (Par conséquent, $L(T)$ est l'ensemble des mots de la forme $au_1 \cdots u_k bu'_1 \cdots u'_\ell c$ avec $u_1, \ldots, u_k, u'_1, \ldots, u'_\ell \in L(T)$.)

(2) Comment exprimer $L(S)$ à partir de $L(T)$ au moyen d'une ou plusieurs opérations rationnelles [2] ? Y a-t-il inclusion de l'un dans l'autre ?

(3) Montrer que tout mot w appartenant à $L(S)$ ou à $L(T)$ a le même nombre de a, de b et de c, c'est-à-dire $|w|_a = |w|_b = |w|_c$ où $|w|_x$ désigne le nombre total d'occurrences de la lettre x dans le mot w.

(4) Montrer par récurrence sur la longueur $|u|$ d'un mot $u \in L(T)$ que si v est un préfixe de u de longueur $0 < |v| < |u|$, alors on a $|v|_c < |v|_a$. Pour cela, on pourra écrire u sous la forme $au_1 \cdots u_k bu'_1 \cdots u'_\ell c$ obtenue en (1), considérer un préfixe [3] d'une telle expression, et appliquer la question (3) et l'hypothèse de récurrence.

(5) Déduire des questions (3) et (4) que si $u \in L(T)$ et si v est un préfixe de u autre que u lui-même, alors [4] $v \notin L(T)$. En déduire que $u \in L(T)$ et $z \in \Sigma^*$, alors u est l'***unique*** préfixe du mot $w := uz$ qui appartienne à $L(T)$ (autrement dit, aucun préfixe de w de longueur $<|u|$ ni $>|u|$ n'appartient à $L(T)$).

(6) En déduire que si un mot w s'écrit $w = u_1 \cdots u_k$ avec $u_1, \ldots, u_k \in L(T)$ alors cette factorisation est unique. (Comment peut-on caractériser u_1 comme préfixe de w ?) Montrer de même la conclusion analogue pour $u_1 \cdots u_k bu'_1 \cdots u'_\ell$ (on pourra noter que $bz \notin L(T)$ quel que soit $z \in \Sigma^*$).

(7) En déduire que la grammaire G est inambiguë.

---

2. C'est-à-dire : union, concaténation, étoile de Kleene.

3. On signale à toutes fins utiles le fait évident suivant : un préfixe non vide de $t_1 \cdots t_n$ où $t_1, \ldots, t_n \in \Sigma^*$ s'écrit sous la forme $t_1 \cdots t_{i-1} y$ où $y \neq \varepsilon$ est un préfixe de t_i.

4. L'énoncé d'origine comportait par erreur la question $u \notin L(T)$ ici.