Contrôle continu Langages formels (LI324)
Aucun document autorisé.
Durée : 2 heures.

1. Soit $X = \{a, b, c, \ldots, z\}$. Proposer un automate **déterministe** et **minimal** qui reconnaisse le langage $X^*issime$[3]. Peut-on proposer une généralisation sur le nombre minimal d'états d'un automate reconnaissant X^*u pour $u \in X^*$, en fonction de la longueur de u ?
2. (a) Soit le langage L_1, formé sur l'alphabet $X_1 = \{0, 1, 2, 3, 4, 5, 7, 8, 9, \textbf{,}\}$, comprenant tous les mots représentant une liste d'entiers séparés par une virgule. Le langage L_1 comprend donc, par exemple, les mots "12", "03,2,414", etc. Proposer une expression rationnelle, un automate et une grammaire régulière qui reconnaît L_1.
   (b) On définit maintenant le langage L_2 formé de tous les mots sur l'alphabet $X_2 = \{$`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, <, >, {, }, a, b, c, d, e, ,`$\}$ qui décrivent un automate fini. Par exemple, le mot "`<{1,2,3},{a,b,c},{1},{3},{<1,a,2>,<2,b,3>,<2,b,2>}>`" appartient à L_2. Proposer une grammaire algébrique qui reconnaît L_2.
   [bonus] Considérons maintenant le langage $L_3 \subset L_2$ des automates *complets*[4]. Décrire avec précision, mais sans en donner le graphe, une machine de Turing qui reconnaît L_3. On supposera dans un premier temps que le nombre d'états est borné, avant de proposer un algorithme plus général.
3. Soit la grammaire suivante $\begin{cases} S \to a \mid b \mid (\ T\ ) \\ T \to T\ ,\ S \mid S \end{cases}$
   (a) Donnez un arbre de dérivation pour les mots (a, b) et $(b, (a, a))$
   (b) La grammaire est-elle LL(1) ?
   (c) Eliminer la récursivité à gauche et factoriser si nécessaire.
   (d) Montrer que la nouvelle grammaire est LL(1). Donner la table d'analyse.
   (e) Expliciter le comportement d'un analyseur descendant sur le mot $(a, (b, a), a)$

[3] Mots formés d'un mot quelconque de X^* suivi des lettres i, s, s, i, m, e.
[4] Rappel : si l'automate est complet, la fonction delta est définie pour **tous** les couples de $Q \times X$.