UNIVERSITÉ DE STRASBOURG

Décembre 2009

Examen de Théorie des Langages
Licence Informatique 3e année

durée : 3 heures
documents de cours autorisés
aucun appareil électronique n'est admis

Les 3 parties sont indépendantes. Le soin apporté à la copie sera pris en compte dans la notation.

Partie I *(4 points)*

Montrez que le langage $L=\{a^n b^p \mid 0<n\leq p\leq 2n\}$ est algébrique en donnant un automate à pile le reconnaissant ainsi qu'une grammaire qui engendre les mots de L.

Partie II *(6 points)*

Les automates à une pile forment un mécanisme de reconnaissance des langages algébriques. En utilisant une pile supplémentaire, il est possible de concevoir des automates reconnaissant des langages non algébriques.

a) En reprenant les notations du cours pour les automates à une pile, donnez la définition formelle d'un automate à deux piles.

b) Donnez l'automate à deux piles qui admet le langage $L1=\{a^n b^n c^n \mid n\in\mathbb{N}\}$. Au préalable, décrire informellement (en quelques lignes) le fonctionnement de l'automate.

c) Après avoir démontré que $L2=\{a^n b^n a^n b^n \mid n\in\mathbb{N}\}$ n'est pas algébrique, donnez un automate à deux piles reconnaissant les mots de ce langage.

d) Le langage $L3=\{a^n b^n c^n d^n \mid n\in\mathbb{N}\}$ peut-il être reconnu par un automate à une pile ? A deux piles ? Si non indiquez la nature de l'automate pouvant reconnaître *L3*. Dans l'affirmative, décrivez informellement puis dessinez l'automate à une pile ou à deux piles correspondant.

Partie III *(10 points)*

Une grammaire *G=(V, T, R, S)* est dite sous forme normale de Chomsky si toutes ses règles sont de la forme :

$S \to \epsilon$ ou
$A \to BC$ avec $A, B, C \in V$ ou
$A \to \alpha$ avec $\alpha \in T$

Cette forme normale présente surtout un intérêt théorique et intervient dans certaines démonstrations (lemme de la double étoile par exemple). Nous nous intéressons ici à la mise sous forme normale de Chomsky d'une grammaire quelconque. Tous les algorithmes sont ici à donner de manière informelle. Dans la suite, on appellera *variable* un symbole non terminal.

a) Indiquez en une à deux phrases pourquoi cette forme de grammaire est importante pour la démonstration du lemme de la double étoile.

Une grammaire est dite *réduite* si aucune variable de *V* n'engendre un langage vide et si toutes les variables de *V* peuvent être atteintes à partir de *S*. On rappelle qu'un langage est vide s'il ne contient aucun mot, pas même le mot vide ϵ.

b) Donnez un algorithme qui construit la grammaire réduite d'une grammaire G quelconque. Indication : produire l'ensemble des *variables utiles*, puis, chercher les variables utiles accessibles à partir de *S*. Une variable est utile si elle conduit à un mot formé de symboles terminaux.

Illustrez le fonctionnement de l'algorithme sur la grammaire suivante et donnez le langage engendré par cette grammaire :

$S \to CA \mid BA \mid \epsilon$
$A \to Sa$
$B \to CE \mid AB$
$C \to b \mid Ba$
$D \to ab \mid ba$
$E \to AB$

Une grammaire est dite *propre* si elle ne contient aucune règle de la forme $A \to \epsilon$ avec $A \in V \setminus \{S\}$ (ϵ-production) et aucune règle de la forme $A \to B$ avec $A, B \in V$ (production unitaire).

c) Pour les ϵ-productions, pourquoi doit-on avoir $A \in V \setminus \{S\}$?
La suppression des ϵ-productions est réalisée par un algorithme qui construit un ensemble de *variables annulables* (variables d'où dérivent le mot vide). Ces variables sont ensuite remplacées dans les règles de production. Donnez l'algorithme qui, à partir d'une grammaire, produit une grammaire équivalente sans ϵ-productions.

Montrez le résultat de cet algorithme sur la grammaire suivante :

$$
\begin{array}{lcl}
S & \rightarrow & AA \mid bBCA \\
A & \rightarrow & BB \mid Cb \\
B & \rightarrow & \epsilon \mid aS \\
C & \rightarrow & b \mid Ba
\end{array}
$$

d) Donnez l'algorithme qui, à partir d'une grammaire, produit une grammaire équivalente sans production unitaire. Montrez le résultat sur la grammaire suivante :

$$
\begin{array}{lcl}
S & \rightarrow & aAb \mid A \mid B \\
A & \rightarrow & bSa \mid S \mid B \\
B & \rightarrow & c \mid cc
\end{array}
$$

e) Décrivez l'algorithme qui, à partir d'une grammaire propre, produit une grammaire équivalente sous forme normale de Chomsky.

f) Mettez sous forme normale de Chomsky la grammaire suivante en indiquant les différentes étapes.

$$
\begin{array}{lcl}
S & \rightarrow & aWb \mid aaXb \mid \epsilon \\
W & \rightarrow & aWb \mid YY \mid \epsilon \\
X & \rightarrow & aaXb \mid W \mid Y \\
Y & \rightarrow & WZ \mid ZX \\
Z & \rightarrow & YY
\end{array}
$$