Licence d'informatique – Université Paris 7

Grammaires et Analyse

Examen du 25 janvier 2002, durée : 2 heures

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

Notes de cours manuscriptes autorisées. Livres non autorisés

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

Exercice 1 (12 points)

Soit la grammaire G suivante d'axiome P et symboles non terminaux : P, D, E, T, L (tous les autres sont des terminaux) :

$$\begin{array}{lcl} P & \rightarrow & D; E \\ D & \rightarrow & D; D \mid \textbf{id} : T \\ T & \rightarrow & \textbf{liste\_de}\ T \mid \textbf{car} \mid \textbf{entier} \\ E & \rightarrow & (L) \mid \textbf{litter} \mid \textbf{nb} \mid \textbf{id} \\ L & \rightarrow & E, L \mid E \end{array}$$

Note (qui peut être utile pour comprendre la semantique de la grammaire, mais qui est absolument **sans rapport** avec la resolution de l'exercice) :
Cette grammaire engendre des programmes, représentés par le non terminal P, qui consistent en une séquence de déclarations D suivie d'une expression représentée par le non terminal E. Une déclaration donne un identificateur et son type, représenté par le non terminal T. Une expression peut être un littéral (constante de type **car**), un nombre (constante de type **entier**), un identificateur (variable de type **car** ou **entier**) ou une liste de tous ceux-ci, y compris une liste de listes (ce qui explique la règle : $T \rightarrow \textbf{liste\_de}\ T$).

a) Eliminer les récursions gauches de cette grammaire et la factoriser à gauche.

b) Calculer les fonctions *Premier* et *Suivant* pour chaque symbole non terminal de la grammaire G' ainsi obtenue.

c) Calculer la table d'analyse syntaxique correspondante à G'.

d) Simuler, à l'aide de la table, l'analyse de la chaîne :

$$\textbf{id} : \textbf{entier}\ ;\ \textbf{id} : \textbf{liste\_de liste\_de entier}\ ;\ ((\textbf{nb}), (\textbf{id}, \textbf{nb}))$$

Si la table que vous avez obtenue n'est pas déterministe (certaines cases contiennent plusieurs règles de production), on pourra "déviner" la règle appropriée à utiliser lorsque un choix non déterministe s'impose.

e) Dessiner l'arbre de dérivation ayant cette chaîne comme frontière et qui est obtenu par l'analyse de la chaîne.

Exercice 2 (8 points)

Soit l'automate à pile défini par les transitions suivantes (q_0 est l'état initial) :

$$\begin{array}{ll} \delta(q_0, a, Z_0) = \{(q_0, AAZ_0)\} & \delta(q_0, a, A) = \{(q_0, AAA)\} \\ \delta(q_0, b, A) = \{(q_1, \epsilon)\} & \delta(q_1, b, A) = \{(q_1, \epsilon)\} \\ \delta(q_1, b, Z_0) = \{(q_0, Z_0)\} & \delta(q_0, \epsilon, Z_0) = \{(q_0, \epsilon)\}. \end{array}$$

a) Quel est le langage reconnu par cet automate par pile vide ? Justifiez votre reponse.

b) Par la méthode donnée en cours, construire directement une grammaire qui engendre le langage L reconnu par l'automate par pile vide.
Suggestion. Pour simplifier les calculs, vous pouvez ne developper que les productions issues des symboles dont l'accessibilité a déjà été établie.

c) Simplifier la grammaire ainsi obtenue en la rendant propre et reduite.
Suggestion. Avant d'appliquer les algorithmes sur la grammaire, regarder si certains symboles sont visiblement inutiles et les éliminer d'emblée. Il est aussi conseillé de renommer les symboles avant d'appliquer les algorithmes.