Compilation - TD3

Analyse syntaxique LL(1)

1 Analyse récursive et dérécursivation à gauche

Question 1-1 Rappelez le principe de l'analyse (grammaticale) récursive. Quel rapport avec les récursions à gauche ?

Question 1-2 Donner l'algorithme de dérécursivation à gauche d'une production du type :

$$A \rightarrow A\alpha_1|\cdots|A\alpha_n|\beta_1|\cdots|\beta_n$$

Question 1-3 Le faire fonctionner sur la grammaire $A \rightarrow AXYz|Az|z|By$.

Question 1-4 Construire un algorithme qui dérécursive une grammaire récursive à gauche indirecte.

Question 1-5 Le tester sur la grammaire suivante :

A → B x y — x
B → C D
C → A — c
D → b

2 Analyse prédictive et factorisation à gauche

Question 2-1 Rappelez le principe de l'analyse prédictive (connue aussi sous le sobriquet LL(1)). Quel rapport avec la factorisation à gauche ?

Question 2-2 Au fait, et les grammaires récursives à gauche, elles vont ou elles vont pas ?

Question 2-3 Quel est le problème avec la grammaire suivante des `if` en C ?

IfElseStatement → `if` (Expression) Statement ElseTailOpt
ElseTailOpt → `else` Statement | ϵ
Statement → ... | IfElseStatement | ...

Une solution dans ce cas est d'exprimer une priorité dans la grammaire pour résoudre le conflit LL(1). Espliquez.

Question 2-4 Donnez la bonne vieille grammaire des expressions algébriques avec les priorités habituelles, et bricolez-là à gauche pour en faire une version LL(1).

Question 2-5 Après avoir trafiqué notre grammaire pour qu'elle soit LL(1), on ne la reconnait plus. Il arrive qu'elle nous construise des arbres qui ne sont plus ceux qu'on voulait. Dans quel cas ? Donnez un exemple archirépandu. Suggérez des solutions.

3 L'automate à pile LL(1)

Question 3-1 Donnez le principe de l'automate à pile LL(1).

Question 3-2 Définissez-le complètement pour la grammaire :

$$\begin{array}{lcl} E & \rightarrow & T \mid E+T \\ T & \rightarrow & \text{ID} \mid (E) \end{array}$$