Feuille de TD n° 2 (séance 4) : analyseurs SLR et OCAMLYACC

Exercice 1 : Soit la grammaire suivante :

$$\begin{array}{ll} E & \to E{+}T \\ E & \to T \\ T & \to \mathbf{id} \end{array}$$

1. Construire la table d'analyse LR(0). Il y a-t-il des conflits ?
2. Calculer `nullable`, `FIRST` et `FOLLOW` pour chaque non-terminal de cette grammaire.
3. Changer la table LR(0) pour obtenir un table SLR en utilisant le `FOLLOW` calculé.
4. Simuler le fonctionnement de la table SLR sur les chaînes $\boxed{\mathbf{id}{+}\mathbf{id}}$ puis $\boxed{\mathbf{id}{+}\mathbf{id}{+}\mathbf{id}}$. Quel est l'arbre de dérivation pour la deuxième chaîne ? Est-il unique et pourquoi ?

Exercice 2 : Étant donnée la grammaire suivante :

$$S \to \mathbf{i}S\mathbf{e}S \mid \mathbf{i}S \mid \mathbf{a}$$

1. Construire la table d'analyse SLR.
2. Résoudre les conflits éventuels de manière à associer chaque occurrence de **e** à l'occurrence de **i** la plus proche (par quel choix entre *décaler* et *réduire* cela se traduit-il ?)
3. Simuler l'analyseur sur la chaîne $\boxed{\mathbf{i\ i\ a\ e\ a}}$ et indiquer l'arbre de dérivation obtenu.

Exercice 3 : Montrer que la grammaire suivante ne peut pas avoir un analyseur SLR.

$$\begin{array}{ll} S & \to L{=}R \\ S & \to R \\ L & \to {*}R \\ L & \to \mathbf{id} \\ R & \to L \end{array}$$

Exercice 4 : Écrire les fichiers OCAMLLEX et OCAMLYACC donnant la définition de chacune des grammaires précédentes de façon à ce que les conflits soient résolus.

Utiliser l'option `-v` de la commande OCAMLYACC et comparer les fichiers `.output` produits avec les tables construites dans les exercices précédents.

Ceci est une correction à l'usage des enseignants, car incomplète.

<u>Correction</u> 1 : C'est l'exemple qui se trouve à la page 62 d'Apple. Il faut introduire la production de début.

1. Oui, c'est un conflit shift-reduce sur le "+".
2. `nullable` est vide. `FIRST`$(E, T) = \{id\}$. `FOLLOW`$(E, T) = \{+\}$.
3. voir livre Apple.
4. Pour la deuxième chaîne, l'arbre est unique et correspond à une dérivation gauche grâce à la grammaire qui est récursive gauche.

<u>Correction</u> 2 : Exemple page 66 d'Apple mais pas traité complètement.

1. todo
2. C'est un conflit *décaler-réduire* pour lequel on donne faveur à *décaler*.
3. facile

<u>Correction</u> 3 : Exemple 3.26 page 65 d'Apple mais qui est traité directement avec LR(1). On ne peut pas avoir une table SLR car il y a des conflits *décaler-réduire* qui ne peuvent pas être enlevés par le calcul de FOLLOW. MS a sur papier la correction. Il faut donc un analyseur LR(1) ou LALR(1).