Compilation - TD4

Analyse syntaxique LR

1 Questions quasi de cours

Question 1-1 Qu'est-ce qui est commun aux méthodes LR(0), LR(1), SLR(1) et LALR(1) ? En quoi diffèrent-elles ?

Question 1-2 Justifiez le 0 de LR(0).

Question 1-3 Y a-t-il des grammaires ambigües LR ?

Question 1-4 Pourquoi empile-t-on des états sur la pile d'un automate LR, alors que lorsqu'on réduit on les jette sans même les lire ?

2 Résolution des conflits *shift/reduce*

Question 2-1 Comment résoudre ces conflits de manière à reconnaître la plus grande chaîne de symboles d'entrée possible ?

Question 2-2 Est-ce que cela marche si on a plusieurs conflits shift/reduce sur le même token ?

Question 2-3 Est-ce que cela marche pour le problème du cours dans le chapitre sur LR(0) (c'était $T \rightarrow i \mid i[E]$) ?

Question 2-4 Est-ce que cela marche pour le problème de l'*else* du C ?

Question 2-5 En cours, on a jeté LR(0) pour (S)LR(1) parce qu'on voulait traiter des grammaire qui avaient des conflits shift/reduce et reduce/reduce en LR(0). Si plus tard le cours vous présente un chapitre intitulé "comment résoudre les conflits *shift/reduce* dans une grammaire LR(1)" (Grune et Bal p.173), devrez-vous crier à l'arnaque ?

3 Items LR(0)

Soit la famille de grammaires G_n définie par

$$\begin{array}{lcll} \mathrm{S} & \rightarrow & A_i b_i & 1 \leq i \leq n \\ A_i & \rightarrow & a_j A_i \mid a_j & 1 \leq i, j \leq n \text{ et } i \neq j \end{array}$$

Question 3-1 Combien de productions a G_n ? Combien a-t-elle d'items LR(0) ? Combien d'ensembles d'items LR(0) ?

Question 3-2 G_n est-elle SLR ? LALR ?

4 Construction de la table d'analyse LALR

Le but de cet exercice est de construire l'ensemble d'items LALR(1) d'une grammaire donnée sans passer par la construction des items LR(1).

On commence par travailler sur des *ensembles réduits d'items* ne comportant pas d'item ayant le point en début de partie droite (on ne met donc pas les productions de la forme $A \rightarrow \epsilon$ – on travaille pour le moment avec des items LR(0)). Ces ensembles réduits sont appelés les *noyaux*

des ensembles LR(0) d'items. À noter qu'il y a un seul noyau comportant une partie droite commençant par un point, à savoir l'axiome $S' \rightarrow .S$.

On veut construire, étant donnée une grammaire, les noyaux des ensembles d'items qui lui sont associés. Pour ce faire, on considère un ensemble d'items I, dont on imagine avoir calculé le noyau. Il est clair que si $[B \rightarrow \gamma.X\delta]$ est dans le noyau de I (X étant un symbole de la grammaire), alors $[B \rightarrow \gamma X.\delta]$ sera dans $\mathit{Transition}(I, X)$. Mais il ne faut pas oublier les items qui ne sont pas notés dans le noyau de I.

Question 4-1 À quelle condition l'item $[A \rightarrow X.\beta]$ sera-t-il également dans $\mathit{Transition}(I, X)$?

Comment calculer par conséquent les ensembles de noyaux d'ensembles d'items à partir d'une grammaire donnée ?

Question 4-2 On appelle G la grammaire suivante :

S' → S
S → G = D | D
G → * D | id
D → G

Donner les ensembles de noyaux pour G.

Question 4-3 On veut maintenant enrichir les items des noyaux en items $LR(1)$. Pour ce faire, on raisonne sur un ensemble I d'items $LR(1)$ contenant un item de la forme $[B \rightarrow \gamma.C\delta, b]$, et on suppose qu'il existe une dérivation droite de la forme $C \Rightarrow_d^* A\eta$. On suppose en outre que la grammaire contient une production $A \rightarrow X\beta$.

Que peut-on dire sur les éléments de $\textsc{Premier}(\eta\delta)$, quant aux items $LR(1)$ qu'il faut rajouter dans $\mathit{Transition}(I, X)$?

Et qu'en est-il lorsque $\eta\delta \Rightarrow^* \epsilon$?

On parlera de symboles de prévision qui se propagent ou qui sont produits spontanément, suivant les cas.

Question 4-4

Utiliser un symbole de prévision factice # pour définir un algorithme qui permet de calculer les symboles de prévision qui se propagent ou qui sont produits spontanément à partir d'un ensemble K d'items $LR(0)$.

Question 4-5 On veut maintenant calculer les noyaux d'ensembles d'items $LALR(1)$ d'une grammaire à partir de ses noyaux $LR(0)$. On admettra (...) que la procédure de la question précédente calcule bien les items $LALR(1)$. Proposer un algorithme exploitant celle-ci.

Question 4-6 Faire tourner tout ça sur la grammaire G présentée ci-dessus.

Question 4-7 Discuter la manière dont pourraient être traitées les actions `décaler` et `réduire`, en ne travaillant que sur des noyaux d'ensembles d'items $LALR(1)$.