Compilation - corrigé du TD4

Analyse syntaxique LR

2 Résolution des conflits *shift/reduce*

Question 2-1 Entre un shift et un reduce, on prend le shift. Cela correspond en général à favoriser la plus longue production de la grammaire.

Question 2-2 Plusieurs conflits shift/reduce, c'est un shift en conflit avec plusieurs reduce. Donc oui.

Question 2-3 (et suivantes) Dans ce cas cela n'aide pas.

Dans l'exemple $T \to i \mid i[E]$, favoriser le shift échoue sur l'entrée $i + i$ (on va shifter le i sans jamais le réduire en T, donc on ne pourra pas réduire la somme), mais favoriser le reduce échoue sur l'entrée $i[i]$. Il n'y a pas de priorité qui aille dans tous les cas. Et cela répond à la question sur l'arnaque.

Le principe de LR(0) c'est de faire un reduce dès qu'on peut. Le passage à (S)LR(1) permet de faire des shift là où en LR(0) on choisissait de faire des reduce. Cela désambigüise certaines grammaires LR(0). Favoriser le shift aide les grammaires qui restent ambigües au sens LR(1). C'est le cas du else.

Au passage, une grammaire ambigüe n'est jamais LR ou LL ou quoi que ce soit. Par définition une grammaire LR (ou LL, etc) c'est une grammaire qui n'a pas de conflit LR (ou LL, etc), et qui est donc pleinement déterministe. Une ambigüité de la grammaire se traduit forcément par un conflit quelquepart.