

TD 12

Grammaires LALR

LALR(k) (pour *look-ahead* LR) est une forme d'analyseur intermédiaire entre SLR et LR(k). Il définit une classe de langages algébriques déterministes strictement contenue entre les deux, en produisant des tables dont la taille est souvent comparable à celle des tables SLR tout en ayant une expressivité suffisante pour la plupart des langages utilisés en pratique. Pour cette raison, la plupart des générateurs d'analyseurs (comme Yacc) utilisent des grammaires LALR(1).

L'idée est la suivante : on commence par produire une table d'analyse LR(k), donc les états sont des ensembles de k -items de la forme $[A \rightarrow \alpha.\beta, u]$. Le cœur d'un état est l'ensemble des 0-items $A \rightarrow \alpha.\beta$ correspondant à des k -items présents dans l'état. Le principe de l'analyse LALR est alors de fusionner les états qui ont le même cœur, en remarquant que la fonction goto ne dépend que du cœur des items. Les actions sont fusionnées aussi, et si aucun conflit ne se produit, on dit que la grammaire considérée est LALR(k).

Exercice 1

1. Formaliser la définition de l'analyseur LALR(k) à partir de l'analyseur LR(k). Montrer que, pour une grammaire qui est effectivement LR(k), la fusion de deux états ne peut jamais produire de conflit shift/reduce, mais seulement des reduce/reduce.
2. Montrer que la grammaire suivante est LALR(1) mais pas SLR(1) :

$$\begin{aligned} S &\rightarrow Aa \mid bAc \mid dc \mid bda \\ A &\rightarrow d \end{aligned}$$

3. Montrer que la grammaire suivante est LR(1) mais pas LALR(1) :

$$\begin{aligned} S &\rightarrow Aa \mid bAc \mid Bc \mid bBa \\ A &\rightarrow d \\ B &\rightarrow d \end{aligned}$$

Exercice 2

Pour chaque $n \geq 1$, soit G_n la grammaire suivante :

$$\begin{aligned} S &\rightarrow A_i b_i && \text{pour } 1 \leq i \leq n \\ A_i &\rightarrow a_j A_i \mid a_j && \text{pour } l \leq i, j \leq n \text{ et } i \neq j \end{aligned}$$

1. Montrer que G_n a $2n^2 - n$ productions et $2^n + n^2 + n$ ensembles de 0-items. En déduire une minoration de la taille d'un analyseur LR en fonction de la taille de la grammaire dans le pire cas.
2. Déterminer si G_n est SLR(1).
3. Déterminer si G_n est LALR(1).