TD 10 : Grammaires LALR(1)

À partir de l'automate des contextes $\mathcal{C}_0$ d'une grammaire algébrique, on peut définir une relation d'équivalence $\equiv_0$ sur $(N \uplus \Sigma)^*$ par

$$\delta \equiv_0 \gamma \text{ iff } \text{goto}(q_0, \delta) = \text{goto}(q_0, \gamma) .$$

On appelle alors un k-item $[A \to \alpha_1 \cdot \alpha_2, u]$ *LALR(k)-valide* dans le contexte γ s'il existe $\delta \equiv_0 \gamma$ tel que $[A \to \alpha_1 \cdot \alpha_2, u]$ soit k-valide pour δ, c'est-à-dire s'il existe une dérivation droite

$$S' \underset{\text{rm}}{\Longrightarrow}^* \rho A v \underset{\text{rm}}{\Longrightarrow} \rho \alpha_1 \alpha_2 v = \delta \alpha_2 v$$

avec $u \in \text{First}_k(v)$. Les notions d'automate des contextes LALR(k), et de conflits LALR(k) en découlent comme d'habitude.

Exercice 1 (Construction d'analyseurs LALR(1))**.**

1. La grammaire suivante est-elle LALR(1) ?

$$\begin{aligned} S &\to aAc \mid bAd \mid agd \mid bgc \\ A &\to B \\ B &\to g \end{aligned}$$

2. On souhaite construire directement un analyseur LALR(1) à partir de l'automate des contextes $\mathcal{C}_0$. L'idée est de calculer, pour chaque item de réduction $[A \to \alpha \cdot]$ présent dans un certain état q de $\mathcal{C}_0$, l'ensemble de symboles terminaux a tels que $[A \to \alpha \cdot, a]$ soit LALR(1)-valide dans le même état q. Proposer une telle méthode de construction.

Exercice 2 (Expressivité)**.**

1. On considère le fragment suivant de la grammaire C ANSI (modulo quelques simplifications) :

$$\begin{aligned} \langle \textit{statement} \rangle &\to \langle \textit{labeled\_statement} \rangle \\ &\mid\ \langle \textit{expression\_statement} \rangle \\ \langle \textit{labeled\_statement} \rangle &\to \textit{id} : \langle \textit{statement} \rangle \\ &\mid\ \texttt{case}\ \langle \textit{conditional\_expression} \rangle : \langle \textit{statement} \rangle \\ \langle \textit{expression\_statement} \rangle &\to \langle \textit{conditional\_expression} \rangle \ ; \\ \langle \textit{conditional\_expression} \rangle &\to \textit{id} \end{aligned}$$

Montrer que cette grammaire est LALR(1) mais pas SLR(k).

2. Montrer que la grammaire suivante est SLL(1) mais pas LALR(k) :

$$\begin{aligned} S &\to aA \mid bB \\ A &\to Cc \mid Dd \\ B &\to Cd \mid Dc \\ C &\to FE \\ D &\to FH \\ E &\to \varepsilon \\ F &\to \varepsilon \\ H &\to \varepsilon \end{aligned}$$

Exercice 3 (Travaux pratiques)**.** Un exemple de générateur d'analyseurs syntaxiques LALR(1) est GNU `bison`. Le fichier d'exemple contient une grammaire des expressions. Comment ajouter les expressions ternaires à la C, de forme « `(1-1) ? 2 : 3` » ?

Exercice 4 (Langages d'arbres)**.** Donner des automates d'arbres pour les langages suivants :

1. L'ensemble des arbres d'arité au plus p dont la frontière est dans le langage rationnel de mots R.
2. L'ensemble des arbres d'arité au plus p dont les étiquettes de toutes les branches sont dans un langage rationnel R.
3. L'ensemble des arbres d'arité au plus p dont au moins une branche est étiquetée par un mot du langage rationnel R.