

TD 8 : Grammaires LL(k)

Exercice 1 (Motifs Objective Caml). Voici un extrait de la syntaxe des motifs d'Objective Caml, sur l'alphabet terminal $\Sigma = \{\text{VN}, -, ::, \text{as}, [,], ;\}$ et l'ensemble de variables $V = \{\langle pat \rangle, \langle patl \rangle\}$:

$$\begin{aligned} \langle pat \rangle &\rightarrow \text{VN} \mid - \mid \langle pat \rangle :: \langle pat \rangle \mid \langle pat \rangle \text{ as VN} \mid [\langle patl \rangle] \\ \langle patl \rangle &\rightarrow \langle pat \rangle \mid \langle patl \rangle ; \langle pat \rangle \end{aligned}$$

Cette grammaire permet de générer des motifs tels que `[a; -; b] :: t as l`

1. Donner une grammaire fortement LL(1) équivalente à ce fragment.
2. Préciser les ensembles First_1 et Follow_1 calculés.
3. Écrire un analyseur récursif descendant dans votre langage favori pour votre grammaire.

Exercice 2 (Grammaires LL(0)).

1. Montrer que si l'automate expansion/vérification associé à une grammaire réduite est déterministe, alors la grammaire est LL(0).
2. Montrer qu'une grammaire LL(0) engendre au plus un mot.

Exercice 3 (Récursivité gauche, ambiguïté).

1. Montrer qu'une grammaire algébrique est LL(k) si et seulement si, pour toutes dérivations gauches $S \xrightarrow{*}_g ux\delta \rightarrow_g u\alpha_1\delta \xrightarrow{*}_g uv_1$ et $S \xrightarrow{*}_g ux\delta \rightarrow_g u\alpha_2\delta \xrightarrow{*}_g uv_2$ avec $\text{First}_k(v_1) = \text{First}_k(v_2)$, nécessairement $\alpha_1 = \alpha_2$.
2. Une variable $x \in V$ est *récursive gauche* s'il existe une dérivation $x \xrightarrow{+} x\alpha$ avec α dans $(V \uplus \Sigma)^*$. Une grammaire réduite est *récursive gauche* s'il existe au moins une variable x *récursive gauche*.
Montrer que si une grammaire réduite est *récursive gauche*, alors elle n'est LL(k) pour aucun k .
3. Montrer que si une grammaire est ambiguë, alors elle n'est LL(k) pour aucun k .

Exercice 4 (Langages fortement LL(k)).

1. Montrer qu'une grammaire algébrique est fortement LL(k) si et seulement si, pour toutes dérivations gauches $S \xrightarrow{*}_g u_1x\delta_1 \rightarrow_g u_1\alpha_1\delta_1 \xrightarrow{*}_g u_1v_1$ et $S \xrightarrow{*}_g u_2x\delta_2 \rightarrow_g u_2\alpha_2\delta_2 \xrightarrow{*}_g u_2v_2$ avec $\text{First}_k(v_1) = \text{First}_k(v_2)$, nécessairement $\alpha_1 = \alpha_2$.
2. Montrer qu'une grammaire est LL(1) si et seulement si elle est fortement LL(1).

3. Montrer que la grammaire suivante est LL(2) mais n'est fortement LL(k) pour aucun k :

$$\begin{aligned} S &\rightarrow aAab \mid bAb \\ A &\rightarrow cAB \mid \varepsilon \mid a \\ B &\rightarrow \varepsilon \end{aligned}$$

4. Montrer que si une grammaire est LL(k), alors il existe une grammaire fortement LL(k) équivalente, qui utilise des variables de la forme (x, F) où $F \subseteq \text{Follow}_k(x)$.
5. Appliquer cette construction à la grammaire précédente.
6. Montrer que si une grammaire est fortement LL(k), alors elle engendre un langage déterministe.

Il existe une hiérarchie stricte des langages LL(k) pour chaque valeur de k . Il existe aussi des langages déterministes qui ne sont pas LL(k).

Exercice 5 (Calcul efficace de First_1 et Follow_1). On appelle *expression relationnelle* un terme décrit par la syntaxe abstraite

$$e ::= r \mid e^* \mid e^{-1} \mid e \cdot e \mid e \cup e$$

où r est une relation "atomique". Une expression relationnelle définit une relation $R(e)$. La *taille* $|e|$ d'une expression relationnelle e est la somme des tailles des relations atomiques qui la composent. Pour une expression relationnelle e de taille finie, on peut calculer une image par sa relation $R(e)$ en temps $O(|e|)$. On souhaite calculer les ensembles First_1 et Follow_1 d'une grammaire G à l'aide d'expressions relationnelles de taille $O(|G|)$, de manière à obtenir des algorithmes en temps linéaire $O(|G|)$.

1. Proposer une expression relationnelle **emptyword** sur $V \times \{\varepsilon\}$ de taille $O(|G|)$ telle que

$$R(\text{emptyword})(x) = \{\varepsilon \mid x \xrightarrow{*} \varepsilon\}.$$

2. Proposer une expression relationnelle **first** sur $V \times (\Sigma \uplus \{\varepsilon\})$ de taille $O(|G|)$ telle que

$$R(\text{first})(x) = \text{First}_1(x).$$

3. Proposer une expression relationnelle **end** sur $V \times \{\varepsilon\}$ de taille $O(|G|)$ telle que

$$R(\text{end})(x) = \{\varepsilon \mid \exists \delta \in (V \uplus \Sigma)^*, S \xrightarrow{*} \delta x\}.$$

4. Proposer une expression relationnelle **follow** sur $V \times (\Sigma \uplus \{\varepsilon\})$ telle que

$$R(\text{follow})(x) = \text{Follow}_1(x).$$

Au moins dans un premier temps, on pourra se contenter d'une expression de taille $O(|G|^2)$.