Magistère d'Informatique 2007-2008
ENS de Cachan
Langages formels

TD 9
–
Analyse descendante (LL)

Exercice 1

1. Montrer que la grammaire suivante est LL(2) mais pas LL(1) :
$$\begin{array}{lcl} S & \rightarrow & abA \mid \varepsilon \\ A & \rightarrow & Saa \mid b \end{array}$$
2. Montrer que la grammaire suivante n'est LL(k) pour aucun k :
$$\begin{array}{lcll} S & \rightarrow & A & \mid B \\ A & \rightarrow & aAb & \mid 0 \\ B & \rightarrow & aBbb & \mid 1 \end{array}$$

Exercice 2

1. Montrer que si l'automate expansion/vérification associé à une grammaire est déterministe, alors la grammaire est LL(0).
2. Montrer qu'une grammaire LL(0) engendre au plus un mot.
3. Montrer que si G est en FNPG et que pour toutes règles $x \rightarrow a\alpha$ et $x \rightarrow b\beta$, avec $a, b \in \Sigma$, on a $a \neq b$ ou $\alpha = \beta$, alors G est LL(1).
4. Montrer que la réciproque est fausse.
5. Soit G une grammaire réduite. Montrer que si G est récursive gauche, alors elle n'est pas LL(k).
6. Montrer qu'un langage LL(k) est non-ambigu.

Exercice 3

Montrer qu'une grammaire est LL(1) si et seulement si elle est fortement LL(1).

Exercice 4

Montrer que la grammaire usuelle pour le langage de Dyck D_n^* sur n paires de parenthèses est LL(1). Donner sa table d'analyse LL(1).
$S \rightarrow \varepsilon \mid a_1Sb_1S \mid \ldots \mid a_nSb_nS$

Exercice 5

Soit la grammaire définie par :

$$E \quad \to \quad E \vee E \quad | \quad E \wedge E \quad | \quad \neg E \quad | \quad (E) \quad | \quad \mathsf{v} \quad | \quad \mathsf{f}$$

Donner une grammaire équivalente en supprimant l'ambiguïté et la récursivité gauche. Construire la table de l'analyseur LL(1) de cette grammaire, et simuler le comportement de l'analyseur sur le mot $\neg(\mathsf{v} \wedge \mathsf{f}) \vee \mathsf{v}$.

Exercice 6

Transformer l'analyseur fortement LL(k) de G en un automate à pile déterministe classique (sans lookahead).

Exercice 7

1. Construire un analyseur déterministe avec lookahead pour une grammaire LL(k).
2. Montrer qu'un langage LL(k) est déterministe.