Langages et Compilation

analyse syntaxique descendante

1 Définitions

Soit $G = (V_t, V_n, Z, P)$ une grammaire hors-contexte. On définit les notions suivantes :

— soit $\alpha \in (V_t \cup V_n)*$, le prédicat **Vide**($\alpha$) vaut vrai ssi ε peut être dérivé de α par G :

$$\textbf{Vide}(\alpha) \equiv \alpha \Rightarrow^*_G \varepsilon$$

— soit $\alpha \in (V_t \cup V_n)*$, **Premiers**($\alpha$) est l'ensemble des symboles terminaux qui peuvent débuter une dérivation de α par G :

$$\textbf{Premier}(\alpha) = \{a \in V_t \mid \alpha \Rightarrow^*_G a\beta\}$$

— soit $X \in V_n$, **Suivants**(X) est l'ensemble des symboles terminaux qui peuvent suivre une occurrence de X dans une dérivation de Z par G :

$$\textbf{Suivants}(X) = \{a \in V_t \mid Z \Rightarrow^*_G \alpha X a\beta\}$$

— soit $X \to u_1.u_2 \ldots u_n$ une règle de P, avec $u_i \in (V_t \cup V_n)$. **Directeurs**($X \to u_1.u_2 \ldots u_n$) est l'ensemble des symboles terminaux qui peuvent débuter une dérivation par cette règle :

$$\begin{array}{rcl}\textbf{Directeurs}(X \to u_1.u_2 \ldots u_n) & = & \textbf{Premiers}(u_1) \\ & & \cup \text{ (si } \textbf{Vide}(u_1) \text{ alors } \textbf{Premiers}(u_2) \text{ sinon } \emptyset) \\ & & ... \\ & & \cup \text{ (si } \textbf{Vide}(u_1.u_2...u_{i-1}) \text{ alors } \textbf{Premiers}(u_i) \text{ sinon } \emptyset) \\ & & ... \\ & & \cup \text{ (si } \textbf{Vide}(u_1 \ldots u_n) \text{ alors } \textbf{Suivants}(X) \text{ sinon } \emptyset)\end{array}$$

Une grammaire hors-contexte $G = (V_t, V_n, Z, P)$ est dite *LL(1)* si et seulement si l'ensemble des règles de P définissant un même symbole non-terminal ont des directeurs disjoints :

$$\forall X \in V_n.\ \forall X \to \alpha, X \to \beta \in P.\ \textbf{Directeur}(X \to \alpha) \cap \textbf{Directeur}(X \to \beta) = \emptyset$$

Un langage hors-contexte est dit LL(1) si et seulement si il existe une grammaire LL(1) qui le reconnait. L'ensemble des langages LL(1) est *strictement inclus* dans l'ensemble des langages hors-contexte.

2 Exemple : langage $L1 = \{a^n b^n c, n \geq 1\}$

On considère la grammaire G_1 qui reconnaît le langage L_1 :

$$\begin{array}{rcl} Z & \to & XY\# \\ X & \to & ab \\ X & \to & aXb \\ Y & \to & c \end{array}$$

Calculons les premiers et suivants de chaque non-terminal puis les directeurs de chaque règle.

$$\begin{array}{rcl}
\textbf{Premier}(Z) & = & \textbf{Premier}(X) \\
\textbf{Premier}(X) & = & \{a\} \\
\textbf{Premier}(Y) & = & \{c\} \\
\\
\textbf{Suivant}(Z) & = & \emptyset \\
\textbf{Suivant}(X) & = & \{b\} \cup \textbf{Premier}(Y) \\
\textbf{Suivant}(Y) & = & \# \\
\\
\textbf{Directeurs}(Z \to XY\#) & = & \{a\} \\
\textbf{Directeurs}(X \to ab) & = & \{a\} \\
\textbf{Directeurs}(X \to aXb) & = & \{a\} \\
\textbf{Directeurs}(Y \to c) & = & \{c\}
\end{array}$$

La grammaire G_1 n'est donc pas LL(1). Pour essayer de la rendre LL(1) nous pouvons factoriser la règle dérivant de X.

Soit la grammaire G_1' qui reconnaît le même langage L_1 :

$$\begin{array}{ccl}
Z & \to & XY\# \\
X & \to & aT \\
T & \to & b \\
T & \to & Xb \\
Y & \to & c
\end{array}$$

$$\begin{array}{rcl}
\textbf{Directeurs}(Z \to XY\#) & = & \{a\} \\
\textbf{Directeurs}(X \to aT) & = & \{a\} \\
\textbf{Directeurs}(T \to b) & = & \{b\} \\
\textbf{Directeurs}(T \to Xb) & = & \{a\} \\
\textbf{Directeurs}(Y \to c) & = & \{c\}
\end{array}$$

La grammaire G_1' est LL(1).

3 Exemple : langage $L_2 = \{a^n b^n c, n \geq 0\}$

On considère la grammaire G_2 qui reconnaît le langage L_2 :

$$\begin{array}{ccl}
Z & \to & XY\# \\
X & \to & \varepsilon \\
X & \to & aXb \\
Y & \to & c
\end{array}$$

$$\begin{aligned}
\textbf{Directeurs}(Z \to XY\#) &= Premier(X) = \{a\} \cup Suivant(X) \\
\textbf{Directeurs}(X \to \varepsilon) &= Suivant(X) = Premier(Y) \cup \{b\} = \{b, c\} \\
\textbf{Directeurs}(X \to aXb) &= \{a\} \\
\textbf{Directeurs}(Y \to c) &= \{c\}
\end{aligned}$$

La grammaire G_2 est LL(1).

4 Exemple : langage $L_3 = \{ba^n, n \geq 0\}$

On considère la grammaire G_3 qui reconnaît le langage L_3 :

$$\begin{aligned}
Z &\to S\# \\
S &\to b \\
S &\to Sa
\end{aligned}$$

$$\begin{aligned}
\textbf{Directeurs}(S \to b) &= \{b\} \\
\textbf{Directeurs}(S \to Sa) &= Premier(S) \text{ et } b \in Premier(S)
\end{aligned}$$

Cette grammaire n'est pas LL(1), de même que toute grammaire récursive à gauche.

Par élimination de la récursivité à gauche on peut toutefois la transformer en une grammaire G'_3 qui reconnait le même langage L_3 :

$$\begin{aligned}
Z &\to S\# \\
S &\to bT \\
T &\to \varepsilon \\
T &\to aT
\end{aligned}$$

On a alors :

$$\begin{aligned}
\textbf{Directeurs}(Z \to S\#) &= \{b\} \\
\textbf{Directeurs}(S \to bT) &= \{b\} \\
\textbf{Directeurs}(T \to aT) &= \{a\} \\
\textbf{Directeurs}(T \to \varepsilon) &= Suivant(T) = Suivant(S) = \{\#\}
\end{aligned}$$

G'_3 est donc une grammaire LL(1).

Remarque : Eliminer la récusivité à gauche ne permet pas toujours d'obtenir une grammaire LL(1).

Par exemple, considérons la grammaire G_4 qui reconnaît le langage $L_4 = \{ba^n, n \geq 1\}$:

$$\begin{aligned}
Z &\to Sa\# \\
S &\to b \\
S &\to Sa
\end{aligned}$$

Après élimination de la récursivité à gauche, nous obtenons la grammaire G'_4 :

$$\begin{array}{rcl} Z & \rightarrow & Sa\# \\ S & \rightarrow & bT \\ T & \rightarrow & \varepsilon \\ T & \rightarrow & aT \end{array}$$

$$\begin{array}{rcl} \textbf{Directeurs}(Z \rightarrow Sa\#) & = & \{b\} \\ \textbf{Directeurs}(S \rightarrow bT) & = & \{b\} \\ \textbf{Directeurs}(T \rightarrow aT) & = & \{a\} \\ \textbf{Directeurs}(T \rightarrow \varepsilon) & = & Suivant(T) = Suivant(S) = \{a\} \end{array}$$

La grammaire G'_4 n'est pas LL(1). Mais nous pouvons écrire une grammaire LL(1) en nous y prenant autrement :

$$\begin{array}{rcl} Z & \rightarrow & bA\# \\ A & \rightarrow & aA \\ A & \rightarrow & a \end{array}$$

qui une fois factorisée donne :

$$\begin{array}{rcl} Z & \rightarrow & bA\# \\ A & \rightarrow & aX \\ X & \rightarrow & \varepsilon \\ X & \rightarrow & A \end{array}$$

$$\begin{array}{rcl} \textbf{Directeurs}(X \rightarrow \varepsilon) & = & Suivant(X) = Suivant(A) = \{\#\} \\ \textbf{Directeurs}(X \rightarrow A) & = & Premier(A) = \{a\} \end{array}$$