Thème Automates à pile.

1 Grammaire non contextuelle et Automates à pile

Exercice 1 Soit la grammaire non-contextuelle suivante pour un langage de Dyck :

$$\begin{array}{ll} 1. & S \to \{\ S\ \} \\ 2. & S \to S\ S \\ 3. & S \to \Lambda \end{array}$$

1. Construire un automate à pile simple indéterministe qui accepte le même langage ;
2. Appliquer cet automate sur le mot {{}{}} ;
3. Donner un automate à pile déterministe qui accepte le même langage ;
4. Appliquer cet automate sur le mot {{}{}}.

Exercice 2 Soit une grammaire $G = (A, V, S, P)$, la grammaire étendue de G à l'ordre k est la grammaire $G' = (A \cup \{\$\}, V \cup \{S'\}, S', P \cup \{S' \to S\$^k\}$ avec $\$ \notin A$ et $S' \notin V$ où \$ est le symbole de fin de mot.

Les symboles directeurs à l'ordre k d'une règle de production, c'est-à-dire les préfixes de taille k des dérivations commençant par cette règle, sont définis par :

$$SD_k(X \to \gamma) = \{m_X \in \bigcup_{0 \leq i \leq k} A^i\,\$^{k-i} \mid \begin{array}{l} S' \Rightarrow^\star m_P\,X\,\delta \Rightarrow m_P\,\gamma\,\delta \Rightarrow^\star m_P\,m_X\,m_S \in A^\star\$^k, \\ m_P \in A^\star, m_S \in A^\star\$^k \cup \bigcup_{0 \leq i \leq k} \$^i \end{array} \}$$

Les premiers à l'ordre k d'une production γ, c'est-à-dire les préfixes de taille k des dérivations de γ, sont définis par :

$$P_k(\gamma) = \{m \in \bigcup_{0 \leq i \leq k} A^i \mid \gamma \Rightarrow^\star m\}$$

Les suivants à l'ordre k d'un non-terminal X, c'est-à-dire les suffixes de taille k de X dans les dérivations de l'axiome, sont définis par :

$$S_k(X) = \{m \in \bigcup_{0 \leq i \leq k} A^i\,\$^{k-i} \mid S' \Rightarrow^\star \alpha X m \beta \Rightarrow^\star m', \alpha \in (A \cup V)^\star, \beta \in (A \cup V \cup \{\$\})^\star, m' \in A^\star\$^k\}$$

Une grammaire G est LL(k) si et seulement si :

1. G n'est pas récursive à gauche (directement ou indirectement) ;
2. Les symboles directeurs à l'ordre k des différentes règles de production d'un même non terminal sont distincts deux à deux.

Nous nous limitons par la suite à l'ordre $k = 1$.

Soit une grammaire simplifiée des expressions arithmetiques en langage ADA $G_0 = (A,\ V,\ E,\ P)$ composée des non-terminaux $V = \{E, L\}$, de l'axiome E, des terminaux $A = \{\texttt{ident}\ (\ )\ ,\}$ et de l'ensemble P de règles suivantes :

$$\begin{array}{ll} 1. & E \to \texttt{ident} \\ 2. & E \to E\ (\ L\ ) \\ 3. & L \to E\ ,\ L \\ 5. & L \to E \end{array}$$

1. Construire G_1 la grammaire augmentée à l'ordre 1 de G_0 ;

2. Déterminer les ensembles des symboles directeurs à l'ordre 1 pour les règles de production de G_1 ;
3. Proposer une définition inductive selon la structure des productions qui permettent de calculer mécaniquement les premiers, les suivants et mes symboles directeurs en se limitant à l'ordre 1 ;
4. Est ce que la grammaire G_1 est récursive à gauche ? Si oui, éliminer la récursivité à gauche et construire la grammaire G_2. Si G_1 n'est pas récursive à gauche, alors G_2 est par la suite égale à G_1 ;
5. Calculer les ensembles des premiers et des suivants à l'ordre 1 pour les non-terminaux de G_2 ;
6. Calculer les symboles directeurs à l'ordre 1 associés aux différentes règles de production de G_2 ;
7. G_2 est-elle LL(1) ? Pourquoi ? Si G_2 n'est pas LL(1), transformer G_2 en G_3 pour la rendre LL(1) et calculer les symboles directeurs à l'ordre 1 de G_3 ;
8. Proposer un programme CaML réalisant l'analyse descendante récursive pour la grammaire G_3.

Exercice 3 Soit une grammaire simplifiée des expressions du langage CaML $G = (A, V, E, P)$ composée des non-terminaux $V = \{E\}$, de l'axiome E, des terminaux $A = \{$`ident` `true` `false` `number` $(\ ) = + - * /\}$ et de l'ensemble P de règles suivantes :

$$
\begin{array}{lrl}
1. & E & \to ER = E \\
2. & E & \to ER \\
3. & ER & \to ER + T \\
4. & ER & \to ER - T \\
5. & ER & \to T \\
6. & T & \to T * F \\
7. & T & \to T \ / \ F \\
8. & T & \to F \\
9. & F & \to - F \\
10. & F & \to (\ E \ ) \\
11. & F & \to \texttt{ident} \\
12. & F & \to \texttt{true} \\
13. & F & \to \texttt{false} \\
14. & F & \to \texttt{number}
\end{array}
$$

1. Construire une grammaire LL(1) équivalente à G. Calculer les symboles directeurs à l'ordre 1 de ses règles de production.