$\mathcal{L}(G_{\text{post}})$ est $LL(1)$

Julie Parreaux

2018-2019

Référence du développement : Schwarzentruber [1, p.151]

Leçon où on présente le développement : 923 (Analyse lexicale et syntaxique).

1 Introduction

L'approche $LL(1)$ est une approche gloutonne de l'analyse syntaxique qui permet de la réaliser en temps linéaire. Cependant pour obtenir un tel résultat, il nous faut utiliser des grammaires bien particulières : les grammaires $LL(1)$. Comme toutes les grammaires ne sont pas $LL(1)$, il nous faut montrer qu'un langage est LL(1) en lui trouvant une grammaire $LL(1)$. C'est ce que nous faisons dans ce développement pour le langage engendré par G_{post}.

Remarques sur le développement

Ce développement permet de montrer qu'une grammaire est $LL(1)$ et qu'elle engendre le bon langage.

1. Exhiber une grammaire candidate.
2. Montrer que cette grammaire est bien $LL(1)$ (caractérisation via les premier et les suivant).
3. Montrer que $\mathcal{L}(G_{\text{post}}) \subseteq \mathcal{L}(G_{\text{post}}^{LL(1)})$ (par récurrence forte).
4. Montrer que $\mathcal{L}(G_{\text{post}}) \supseteq \mathcal{L}(G_{\text{post}}^{LL(1)})$ (par récurrence forte).

2 La grammaire G_{post} est $LL(1)$

Définition. La grammaire G_{post} est définie par

$$\begin{array}{lcl} S & \rightarrow & SS+ \\ S & \rightarrow & c \end{array}$$

Proposition. *Le langage engendré par la grammaire G_{post} est $LL(1)$, c'est-à-dire qu'il existe une grammaire $LL(1)$ qui reconnaît exactement le même langage.*

Démonstration. **Étape 1** Exhibons une grammaire candidate.

$$G_{\text{post}}^{LL(1)} : \begin{array}{lcl} S & \rightarrow & ST \\ T & \rightarrow & S+T \\ T & \rightarrow \epsilon & \end{array}$$

Étape 2 Montrons que $G_{\text{post}}^{LL(1)}$ est $LL(1)$.

— S donne une seule règle donc la grammaire est $LL(1)$ en S.

— T : $\text{Premier}(\epsilon) \cap \text{Premier}(S+T) = \emptyset$ et $\text{Suivant}(T) \cap \text{Premier}(S+T) = \emptyset$ donc la grammaire est $LL(1)$ en T.

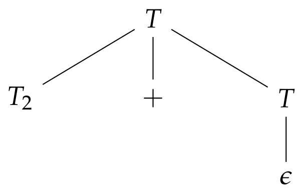
$$\begin{aligned}
&\text{Suivant}(S) = \{+, \propto\} \\
&\text{Suivant}(T) = \text{Suivant}(S) \cup \text{Suivant}(T) = \{+, \propto\} \\
&\text{Premier}(cT) = \{c\} \\
&\text{Premier}(S+T) = \text{Premier}(S) = \{c\} \\
&\text{Premier}(\epsilon) = \{\bowtie\}
\end{aligned}$$

Étape 3 Montrons que $\mathcal{L}(G_{\text{post}}) \subseteq \mathcal{L}(G_{\text{post}}^{LL(1)})$. Raisonnons par récurrence forte sur $n \in \mathbb{N}$ via la propriété P_n : "un mot contenant n occurrences de $+$ engendré par G_{post} est engendré par $G_{\text{post}}^{LL(1)}$".

Initialisation Montrons que P_0. Le seul mot sans occurrence de $+$ par G_{post} est c. Il est également engendré par $G_{\text{post}}^{LL(1)}$: $S \to CT \to c\epsilon \to c$.

Hérédité Soit $n \in \mathbb{N}$ tel que $\forall i \leq n$ P_i est vrai. Montrons que P_{n+1}. Soit $m_1 m_2 +$ un mot engendré par G_{post} ayant $(n+1)$ terminaux $+$.

- Par définition de G_{post}, m_1 et m_2 sont également engendrés par G_{post}.
- Les mots m_1 et m_2 ont au plus n terminaux $+$ chacun. Par hypothèse de récurrence, il existe deux arbres de dérivation T_1 et T_2 de $G_{\text{post}}^{LL(1)}$ tel que $\text{mot}(T_1) = m_1$ et $\text{mot}(T_2) = m_2$.
  * Dans T_1 il existe une branche $T - \epsilon$ (sinon, il resterait des non terminaux T dans le mot).
  * On peut donc prendre celle qui est la plus à droite pour la remplacer par



  * Cet arbre est bien engendré par la grammaire (correspond à $T \to T_2 + T \to T_2 + \epsilon \to T_2$).
  * Le mot des feuilles de cet arbre est $m_1 m_2 +$.
  - On a pris la branche $T - \epsilon$ la plus à droite dans T_1 donc il n'y a pas de lettre après.
  - À gauche on génère m_1.
  - L'arbre inséré génère donc m_2.

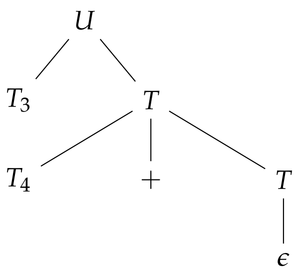
D'où la récurrence.

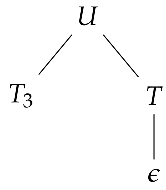
Étape 3 Montrons que $\mathcal{L}(G_{\text{post}}^{LL(1)}) \subseteq \mathcal{L}(G_{\text{post}})$. Raisonnons par récurrence forte sur $n \in \mathbb{N}$ via la propriété P_n : "un mot contenant n occurrences de $+$ engendré par $G_{\text{post}}^{LL(1)}$ est engendré par G_{post}".

Initialisation Montrons que P_0. Le seul mot sans occurrence de $+$ par $G_{\text{post}}^{LL(1)}$ obtenu par $S \to CT \to c\epsilon \to c$ est c. Il est également engendré par G_{post} par $S \to c$.

Hérédité Soit $n \in \mathbb{N}$ tel que $\forall i \leq n$ P_i est vrai. Montrons que P_{n+1}. Soit un mot m avec $(n+1)$ occurrence du non-terminal $+$. Cherchons le $+$ de la racine de l'arbre de dérivation de la grammaire de G_{post}. On parcours en profondeur la branche droite de l'arbre de dérivation de m par $G_{\text{post}}^{LL(1)}$.

- Cette branche se termine $T - \epsilon$.
- Le nœud père de $T - \epsilon$ ne peut pas être S (sinon pas de $+$) donc ce nœud est nécessairement T et on a l'arbre suivant. Si on remplace la branche de racine T par $T - \epsilon$, on a avec T_3 qui possède un $+$ en mot donc par l'hypothèse de récurrence, il existe m_3 un mot de G_{post} correspondant à T_3. De même pour T_4 qui donne m_4 (par hypothèse de récurrence). Donc l'arbre génère $m_3 m_4 + \in \mathcal{L}(G_{\text{post}})$.





□

Références

[1] F. Schwarzentruber R. Legendre. *Compilation : Analyse lexicale et syntaxique du texte à sa structure en informatique*. Reference Sciences. Ellipses, 2015.