Analyse d'une grammaire LL(1)

Benjamin Hellouin

Aho-Sethi-Ullman, 4.4

Soit G une grammaire hors contexte sur Σ, V. On veut construire un analyseur prédictif non récursif, c'est-à-dire un algorithme permettant de décider "gloutonnement" à la lecture d'un mot si celui-ci est engendré par G. Ici, cet algorithme prend la forme d'un automate à pile.

L'alphabet d'entrée est Σ, l'alphabet de pile $V \cup \Sigma$, symbole initial S. A chaque instant, l'automate décide si on peut dériver ce qu'il reste en entrée des symboles de la pile. On se donne une *table d'analyse* T : une application $V \times \Sigma \to G \cup \bot$ qui a (A, a) associe une règle du type $A \to \omega$. Alors l'automate se comporte de la manière suivante :

- l'entrée est a, la pile est b : si $a = b$, on dépile et on consomme une lettre. Sinon, erreur.
- l'entrée est a, la pile est A : on applique la règle $T(A, a)$ (erreur si $\bot$) et on empile ω.

Il est clair que si l'automate accepte, le mot est reconnu. On cherche à construire une table d'analyse qui fournisse les règles à appliquer dans chaque situation.

Définition 1. *Pour chaque mot α, on note $PREMIER(\alpha)$ l'ensemble des terminaux en tête d'une chaîne dérivable à partir de α. Pour chaque variable A, on note $SUIVANT(A)$ l'ensemble des terminaux a tels qu'il existe une dérivation $S \to^* \alpha A a \beta$.*

L'idée est la suivante : supposons qu'on cherche à dériver un mot commençant par a depuis un mot commençant par A. On ne peut utiliser une règle $A \to \alpha$ que si :

- $a \in PREMIER(\alpha)$;
- $\alpha \to \varepsilon$ et $a \in SUIVANT(A)$.

On peut ainsi facilement construire une table des seules règles possiblement utilisables, en supposant qu'on a calculé $PREMIER$ et $SUIVANT$. Si une telle table comporte au plus une entrée par case, la grammaire est dite LL(1).

Calcul de PREMIER.

On commence par le calculer sur les variables.

Étape 1 Si $A \to \varepsilon$, ajouter ε à $PREMIER(a)$;

Étape 2 Si $A \to a\alpha$, ajouter a à $PREMIER(A)$;

Étape 3 Si $A \to A_1 \dots A_n$, ajouter $PREMIER(A_1)$ à $PREMIER(A)$. Si de plus $\varepsilon \in PREMIER(A_1)$, ajouter $PREMIER(A_2)$, etc.

On répète ces étapes tant qu'on rajoute des éléments à un des ensembles, ce qui garantit la terminaison. Pour les mots, la définition est similaire à l'étape 3. La correction est intuitive : dans chaque cas, on n'ajoute que des éléments appartenant effectivement à l'ensemble ; et si un élément appartient à l'ensemble, on prouve facilement par récurrence sur la longueur de la plus courte dérivation que cet algorithme le trouve.

Calcul de $SUIVANT$.

Étape 1 Si $A \to \alpha B \beta$, on ajoute $PREMIER(\beta)$ à $SUIVANT(B)$;

Étape 2 Si $A \to \alpha B$ ou $A \to \alpha B \beta$ avec $\varepsilon \in PREMIER(\beta)$, on ajoute $SUIVANT(A)$ à $SUIVANT(B)$.

Ce calcul fonctionne en grande partie comme le précédent et sa correction se démontre de la même façon.