

# Introduction à la compilation – TD 1 :

## Analyse LL(1)

Université Paris Diderot – Licence 3

(2012-2013)

### Exercice 1

$$\begin{aligned} S &\rightarrow E\$ \\ E &\rightarrow OEE \mid A \\ O &\rightarrow + \mid * \\ A &\rightarrow n \mid (E) \end{aligned}$$

1. Calculez la fonction *FIRST* pour tous les non terminaux de cette grammaire.
2. La grammaire est-elle LL(1) ? Si oui, quelle est la table d'analyse correspondante ? Si non, existe-t-il une grammaire équivalente LL(1) ? Dans ce cas, donnez-en la table d'analyse.
3. Écrivez un programme CAML qui interprète cette table pour déterminer si un mot est dans le langage engendré par cette grammaire.
4. Écrivez un programme CAML qui implémente cette table à l'aide de quatre fonctions mutuellement récursives.
5. Explicitez les étapes de la reconnaissance de la chaîne  $+( * 21 ) ( + ( + 11 ) 2 ) \$$ .
6. Explicitez les étapes du rejet de la chaîne  $( + 1 ( * 2 ) ) \$$ .
7. (★) Comment adapter votre analyseur syntaxique pour qu'il produise un arbre de production si il existe ?
8. (★) Comment obtenir un arbre de syntaxe abstraite à partir de l'arbre de production produit par votre analyseur syntaxique ?

□

### Exercice 2

$$\begin{aligned} S &\rightarrow E\$ & E &\rightarrow T + E \mid T! \\ T &\rightarrow F * T \mid F! & F &\rightarrow n \mid (E) \end{aligned}$$

1. Calculez la fonction *FIRST* pour tous les non terminaux de cette grammaire.
2. La grammaire est-elle LL(1) ? Si oui, quelle est la table d'analyse correspondante ? Si non, existe-t-il une grammaire équivalente LL(1) ? Dans ce cas, donnez-en la table d'analyse.

□

### Exercice 3

Considérez la grammaire suivante :

$$\begin{aligned} S &\rightarrow E\$ & E &\rightarrow E + T \mid E - T \mid T \\ T &\rightarrow T * F \mid F & F &\rightarrow id \mid (E) \mid (num) \end{aligned}$$

1. Pourquoi cette grammaire n'est pas LL(1) ?
2. Éliminez les récursions gauches.
3. Est-ce que la grammaire obtenue ainsi est LL(1) ?

□

**Exercice 4** On considère la grammaire :

$$\begin{array}{ll} S \rightarrow E\$ & E \rightarrow TE' \\ E' \rightarrow +TE' \mid \epsilon & T \rightarrow FT' \\ T' \rightarrow *FT' \mid \epsilon & F \rightarrow (E) \mid id \end{array}$$

1. Calculez les symboles annulables (c'est-à-dire les symboles qui ont  $\epsilon$  dans *FIRST*)
2. Calculez *FIRST* des symboles non terminaux.
3. Calculez *FOLLOW* des symboles non terminaux.
4. Donnez une table d'analyse descendante pour cette grammaire. La grammaire est-elle *LL(1)* ? Expliquez.

□

**Exercice 5** On considère la grammaire :

$$S \rightarrow A \mid B \quad A \rightarrow aAb \mid 0 \quad B \rightarrow aBbb \mid 1$$

1. Quel est le langage généré par cette grammaire ?
2. Calculer les ensembles *FIRST* de *S*, *A* et *B*.
3. La grammaire est-elle *LL(1)* ?
4. Pouvez-vous construire un automate à pile déterministe qui reconnaît le langage généré par la grammaire ?
5. (★) Pouvez-vous écrire un programme CAML qui reconnaît ce langage ?

□

**Exercice 6** On considère la grammaire suivante :

$$S \rightarrow iEtSS' \mid a \quad S' \rightarrow eS \mid \epsilon \quad E \rightarrow b$$

Expliquez pourquoi cette grammaire n'est pas *LL(1)*.

La grammaire est-elle ambiguë ?

□

**Exercice 7** Considérez la grammaire suivante :

$$\begin{array}{ll} S \rightarrow S'\$ & S' \rightarrow S'AB \mid S'BC \mid BB \\ A \rightarrow aAa \mid \epsilon & B \rightarrow bB \mid \epsilon \quad C \rightarrow cC \mid \epsilon \end{array}$$

Calculez les symboles annulables et les ensembles *FIRST* et *FOLLOW*. Est-ce que la grammaire est *LL(1)* ?

□