

# TD6 : Parsing

Timothée Bernard

21 octobre 2015

## 1 Parsing LL

- Éliminer la récursion (gauche directe) de la grammaire ETF :
$$\begin{aligned} E &\rightarrow T \mid E + T \\ T &\rightarrow F \mid T * F \\ F &\rightarrow a \mid ( E ) \end{aligned}$$
- Donner les ensembles PREMIER et SUIVANT pour chaque non terminal et construire la table de prédiction LL associée à la grammaire obtenue à la question précédente<sup>1</sup>.
- Appliquer une version modifiée de l'algorithme de parsing descendant vu en TD il y a deux semaines, en utilisant la table LL, pour analyser le mot  $a + a * (a + a)$ .
- Soit la grammaire suivante :  $\left\{ \begin{array}{l} S \rightarrow a \mid b \mid ( T ) \\ T \rightarrow T , S \mid S \end{array} \right\}$ 
  - Donnez un arbre de dérivation pour les mots  $(a, b)$  et  $(b, (a, a))$
  - La grammaire est-elle LL(1)?
  - Éliminer la récursivité à gauche.
  - Montrer que la nouvelle grammaire est LL(1). Donner la table d'analyse.

## 2 Parsing LR

- Soit la grammaire suivante :
$$\begin{aligned} S &\rightarrow A B \\ A &\rightarrow a A \mid a \\ B &\rightarrow a B \mid b \end{aligned}$$
  - Calculer les contextes LR(0) de chaque règle.
  - Appliquer l'algorithme naïf (shift-reduce + contextes LR(0)) sur l'entrée  $aab$  (dessiner l'arbre des configurations explorées, en précisant là où les contextes LR(0) ont permis d'éviter de fausses routes).

---

1. Ne pas oublier le marqueur de fin \$.