

TD sur l'analyse syntaxique

1<sup>er</sup> février 2008

**Prérequis** : TD15 + arbre syntaxique, analyseur syntaxique LL(1) récursif, analyseur syntaxique LL(1) non-récursif.

**Durée** : 1 h 50

## TD 16 – Analyse syntaxique descendante

Analyseurs LL(1) récursif et non-récursif

### Exercice 1 : Arbres syntaxiques

Soit la grammaire  $G = (\{X, Y, Z\}, \{a, b\}, \rightarrow, X)$  où

$$\begin{aligned} X &\rightarrow aYbX \quad \textcircled{1} \mid bZaX \quad \textcircled{2} \mid \varepsilon \quad \textcircled{3} \\ Y &\rightarrow aYbY \quad \textcircled{4} \mid \varepsilon \quad \textcircled{5} \\ Z &\rightarrow bZaZ \quad \textcircled{6} \mid \varepsilon \quad \textcircled{7} \end{aligned}$$

où les règles ont été numérotées de 1 à 7.

1. Dessiner l'arbre syntaxique du mot *abba* et écrire cet arbre sous forme postfixée en utilisant les numéros des règles.
2. Dessiner les arbres syntaxiques correspondant aux formes postfixées suivantes : *bb7a76a32* et *b7ab7a322*.

### Exercice 2 : Analyse LL(1) récursive

On considère la grammaire  $G$  de l'exercice précédent, on a montré que cette grammaire était LL(1). Construire un analyseur syntaxique en définissant pour chaque non terminal une procédure, donner aussi le programme principal et la procédure correspondant à un terminal (on appelle cette approche la descente récursive).

### Exercice 3 : Analyse LL(1) non récursive

On considère la grammaire définie par les règles suivantes :

$$\begin{aligned} S &\rightarrow a \mid b \mid (T) \\ T &\rightarrow ST' \\ T' &\rightarrow \rightarrow, ST' \mid \varepsilon \end{aligned}$$

1. Construire la table d'analyse de cette grammaire
2. En utilisant cette table et un automate à pile, analyser le mot  $(b, a, b)\$$ .