**INPG** **ENSIMAG**
**Année Spéciale** **Année 2005-06**

# *Compilation*

**TD4 - analyse syntaxique descendante**

## Récursivité à gauche

**Exercice 1** En vous autorisant l'utilisation des productions vides ($C \longrightarrow \epsilon$), expliquez comment transformer automatiquement certaines grammaires pour éliminer tout récursivité à gauche ($\alpha, \beta, \gamma \in V_T$) :

$$A \longrightarrow A\alpha \mid \beta$$

Étendez la méthode lorsque la récursivité est indirecte :

$$\begin{array}{rcl} A & \longrightarrow & B\alpha \\ B & \longrightarrow & A\beta \mid \gamma \end{array}$$

**Exercice 2** Réécrivez la grammaire des expressions arithmétiques en sa variante « récursive à droite ».

## Grammaires LL(1)

**Exercice 3** Rappellez les définitions des ensembles *first* et *follow*. Donnez leurs valeurs pour la grammaire de l'exercice 2.

**Exercice 4** Déterminez si les grammaires suivantes sont LL(1).

– G1 :

$$\begin{array}{rcl} Z & \longrightarrow & S\# \\ S & \longrightarrow & Xa \\ S & \longrightarrow & \epsilon \\ X & \longrightarrow & Xb \\ X & \longrightarrow & Y \\ Y & \longrightarrow & aX \\ Y & \longrightarrow & \epsilon \end{array}$$

– G2 :

$$\begin{array}{rcl} Z & \longrightarrow & S\# \\ S & \longrightarrow & XaY \\ X & \longrightarrow & W \\ X & \longrightarrow & T \\ W & \longrightarrow & bc \\ T & \longrightarrow & ac \\ Y & \longrightarrow & eY \\ Y & \longrightarrow & f \\ Y & \longrightarrow & \epsilon \end{array}$$

**Exercice 5** Modifiez les grammaires suivantes pour les rendre LL(1). Vérifiez.

– G1 :

$$\begin{array}{rcl} Z & \longrightarrow & S\# \\ S & \longrightarrow & Sa \\ S & \longrightarrow & b \end{array}$$

– G2 :

$$\begin{array}{rcl} Z & \longrightarrow & S\# \\ S & \longrightarrow & aXa \mid bXb \mid a \mid b \mid \epsilon \\ S & \longrightarrow & b \end{array}$$

**Exercice 6** Soit la grammaire suivante ($V_T = \{:, =, i, e, ;\}$). Cette grammaire est-elle LL(1) ? Peut-on la rendre LL(1) ? Comment résoudre ce problème ?

$$\begin{array}{rcl} Z & \longrightarrow & S\ ; \\ S & \longrightarrow & V := e \\ S & \longrightarrow & LS \\ L & \longrightarrow & i : \\ V & \longrightarrow & i \end{array}$$

## Implémentation

**Exercice 7** De quelle façon implémenteriez vous un parser descendant ?