

# Partiel

## Documents interdits

*Durée: 2h*

14 mars 2013

## 1 Grammaires LL

Soit la grammaire  $G$  :

$$\begin{aligned} S_1 &\rightarrow S_2S_3 \\ S_2 &\rightarrow aS_2 \mid \varepsilon \\ S_3 &\rightarrow S_4S_5 \\ S_4 &\rightarrow bS_4 \mid \varepsilon \\ S_5 &\rightarrow cS_5d \mid \varepsilon \end{aligned}$$

**Q.1.1.** Quel est le langage engendré par  $G$  ?

**Q.1.2.** Écrire les fonctions PREMIER et SUIVANT pour les non-terminaux de  $G$ .

**Q.1.3.** Écrire la table  $LL(1)$  de  $G$ .

**Q.1.4.** Simulez l'analyse  $LL(1)$  du mot  $abcd$

On considère maintenant la grammaire  $G'$  :

$$\begin{aligned} S_1 &\rightarrow S_2S_3 \\ S_2 &\rightarrow aS_2 \mid \varepsilon \\ S_3 &\rightarrow S_4S_5 \\ S_4 &\rightarrow bS_4 \mid \varepsilon \\ S_5 &\rightarrow aS_5b \mid \varepsilon \end{aligned}$$

**Q.1.5.** Écrire la table  $LL(1)$  de  $G'$ .

**Q.1.6.** Que s'est-il passé ?

## 2 Déclaration et initialisation de tableaux en C

Une variable de type tableau en C peut être initialisée lors de sa déclaration à l'aide d'une liste. Cela se fait de la façon suivante<sup>1</sup> :

```
int t[5] = {1,2,3,4,5}
```

1. Le langage C permet de ne pas spécifier la dimension des tableaux et de la calculer lors de la compilation grâce au nombre d'éléments dans la liste d'initialisation. Nous ignorerons ici cette possibilité.

Les éléments constituant la liste doivent être des constantes<sup>2</sup>. Il peut y avoir moins d'éléments dans la liste que la taille du tableau. En revanche, s'il y en a plus, une erreur est signalée.

Lorsque le tableau comporte des sous-tableaux, c'est à dire lorsque c'est un tableau à plusieurs indices, la liste d'initialisation peut comporter des sous-listes, indiquées à leur tour par des accolades, comme dans les exemples suivants :

```
int t1[3][3]= {{1,2,3}, {4,5,6}, {7,8,9}}
int t1[2][2][2]= {{{1,2},{3,4}},{5,6},{7,8}}}
```

**Q.2.1.** Ecrire une grammaire  $G$  générant des déclarations de tableaux éventuellement accompagnées de leurs initialisations.

- Pour simplifier, on ne s'autorise que les types de base `int`, `char` et `double`.
- On considère que les dimensions des tableaux sont des constantes entières.
- On ne s'autorisera pas l'usage des crochets et des accolades dans la grammaire pour représenter l'optionnalité et la répétition.

**Q.2.2.** Dessiner l'arbre de dérivation que  $G$  associe au mot

```
int t1[2][2]= {{1,2},{3,4}}
```

**Q.2.3.** On désire signaler le cas où le nombre d'éléments dans l'initialisation est supérieur à la dimension du tableau. Proposez des attributs ainsi qu'un schéma de traduction dirigé par la syntaxe permettant de détecter ce type d'erreurs. Pour simplifier, on se limitera aux tableaux de dimension 1. Vous pouvez par conséquent vous servir d'une grammaire plus simple que  $G$ .

### 3 Grammaires ambiguës

Soit la grammaire suivante :

$$\begin{aligned} S &\rightarrow S'S' \\ S' &\rightarrow aS'a \mid bS'b \mid \varepsilon \end{aligned}$$

**Q.3.1.** Montrer que cette grammaire est ambiguë.

**Q.3.2.** Quel est le langage engendré par cette grammaire ?

**Q.3.3.** Expliquer pourquoi ce langage ne peut pas être généré par une grammaire non ambiguë.

---

2. En réalité il s'agit d'expressions constantes, mais, pour simplifier, on considèrera ici qu'il s'agit de constantes.