

**L3**  
**ULI6BT Théorie des langages et compilation**  
**durée 2h**

Les notes de cours et TD sont autorisées.

*Chaque candidat doit, en début d'épreuve, porter son nom dans le coin de la copie réservé à cet usage; il le cachettera par collage après la signature de la feuille d'émargement. Sur chacune des copies intercalaires, il portera son numéro de place.*

**Exercice I. Automate Fini**

On considère le langage  $L$  sur l'alphabet  $\{a, b\}$  décrit par l'expression régulière  $(aa + b)^*ab(bb)^*$ .

**Question 1.** Donner tous les mots de longueur cinq appartenant au langage  $L$ .

**Question 2.** En utilisant l'algorithme de GLUSHKOV, donner un automate fini non déterministe qui reconnaît le langage  $L$ .

**Question 3.** Donner une grammaire régulière qui engendre le langage  $L$ .

**Exercice II. Compilation**

On souhaite enrichir le langage de calculette vu en TD/TP avec la structure de contrôle **repeat ... until ...**.

En guise d'exemple, on considère le programme suivant qui détermine le nombre de bits  $d$  de l'écriture en binaire d'un nombre  $nb$ .

```
var nb
var d
nb = 19
d = 0

repeat {
  nb = nb/2
  d = d+1
}
until nb<1

d
```

**Question 4.** On suppose données les règles de la grammaire et les actions de construction de l'AST pour les autres instructions de la calculette.

**4.a.** Définir les règles de la grammaire pour la prise en compte des boucles **repeat ... until ...**.

**4.b.** Ajouter les actions nécessaires pour la construction de l'AST.

**Question 5.** Donner l'AST produit par l'analyse syntaxique du programme donné en exemple.

**Question 6.** Quel code MVàP doit être généré par le compilateur pour réaliser le programme donné en exemple.

**Question 7.** Écrire les règles ANTLR de génération de code pour les boucles **repeat ... until ...**.

### Exercice III. Analyse LL

On considère la grammaire  $G$  d'axiome  $E$  et de terminaux  $\{ + , * , \text{nb} , ( , ) \}$  définie par :

$$\begin{cases} E & \rightarrow \text{nb} \mid ( \text{OP } E \text{ L} ) \\ \text{OP} & \rightarrow + \mid * \\ L & \rightarrow E \text{ X} \\ X & \rightarrow L \mid \epsilon \end{cases}$$

**Question 8.** Donner un arbre d'analyse pour le mot  $( * \text{nb nb nb} )$  .

**Question 9.**

**9.a.** Déterminer les variables effaçables.

**9.b.** Donner la table des ensembles **Premier**.

**9.c.** Donner la table des ensembles **Suivant**.

**Question 10.** Construire la table d'analyse  $LL(1)$  de la grammaire  $G$ . Pourquoi la grammaire  $G$  est-elle  $LL(1)$  ?

**Question 11.** Dérouler l'analyse  $LL(1)$  sur les deux entrées suivantes :

-  $( * \text{nb} )$

-  $( * \text{nb nb nb} )$

### Exercice IV. Analyse ascendante

On considère la grammaire  $G$  précédente.

**Question 12.** Effectuer l'analyse syntaxique ascendante par décalage/réduction du mot  $( * \text{nb nb nb} )$  . Il ne s'agit pas ici de construire la table d'analyse SLR, LR ou LALR de la grammaire  $G$  mais simplement de deviner à chaque étape l'opération de décalage ou de réduction à réaliser.

**Question 13.** Dédurre de cette analyse une dérivation du mot  $( * \text{nb nb nb} )$  .