

**Théorie des langages et de la compilation**  
**Année académique 2004-2005**  
**1ère licence en Informatique et**  
**2ème année du grade l'ingénieur civil informaticien**  
**Examen de première session**

## Remarques préliminaires

- On vous demande de répondre à chaque question sur des feuilles séparées (les correcteurs pouvant être différents).
- N'oubliez pas d'inscrire vos nom et prénom sur chacune des feuilles.
- Vous ne pouvez pas utiliser de notes.
- L'examen comporte cinq questions pour un total de 16 points, les projets (4 points) complétant la note.
- Vous disposez de quatre heures. Pensez à gérer correctement votre temps.
- Si rien n'est spécifié pour les formalismes, les conventions utilisées habituellement dans le cours sont prises.

## Question 1 (6pts)

Soit le langage  $L = \{0^n 1^n \mid n \geq 1\}$

1. Parmi les formalismes suivants :

- (a) les automates finis déterministes
- (b) les automates à pile
- (c) les machines de Turing
- (d) les grammaires
- (e) les expressions régulières

lesquels permettent de définir le langage  $L$  (précisez si nécessaire, mais sans démonstration).

2. Choisissez 3 formalismes parmi ceux-ci et donnez une définition formelle complète pour  $L$  dans ces formalismes.

3. Pour une grammaire  $G = \langle N, \Sigma, P, S \rangle$ , on a  $L(G) = \{w \mid S \xRightarrow{*} w\}$

Définissez formellement  $\alpha \xRightarrow{*} \beta$  et  $\alpha \Rightarrow \beta$

4. Énoncez et démontrez formellement le lemme de pompage (*pumping lemma*) des langages réguliers.

5. Peut-on définir  $L$  avec une grammaire ayant des règles dans les 3 formats suivants :

- $A \rightarrow a$
- $A \rightarrow B a$
- $A \rightarrow a B$

avec  $A, B$  des variables (non terminaux) et  $a$  un terminal quelconque. (Dé)montrez.

6. Peut-on définir  $L$  avec une expression régulière ? (Dé)montrez.

## Question 2 (3pts)

1. Transformez la grammaire dont les règles de production suivent en grammaire  $LL(1)$  "nettoyée" en justifiant chaque étape, et expliquant les algorithmes utilisés.

$$\begin{aligned} S &\rightarrow a E \mid b F \\ E &\rightarrow b E \mid \epsilon \\ F &\rightarrow a F \mid a G \mid a H D \\ G &\rightarrow G c \mid d \\ H &\rightarrow C a \\ C &\rightarrow H b \\ D &\rightarrow a b \end{aligned}$$

2. Construisez l'analyseur  $LL(1)$  récursif (utilisez un formalisme C-like) pour cette grammaire nettoyée en expliquant votre démarche.

## Question 3 (2pts)

Construisez l'analyseur  $SLR(1)$  de la grammaire suivante :

$$\begin{aligned} S' &\rightarrow S\$ \\ S &\rightarrow S a S b \mid c \mid \epsilon \end{aligned}$$

en expliquant les fonctions et algorithmes utilisés

## Question 4 (3pts)

Expliquez le rôle de l'analyse sémantique et les outils qu'elle a à sa disposition.

## Question 5 (2pts)

1. Définissez langage
  - (a) de classe 0 dans la hiérarchie de Chomsky,
  - (b) de classe 1 dans la hiérarchie de Chomsky,
  - (c) Récursivement Enumérable,
  - (d) Récursif.
2. Donnez le principe de la preuve qui montre que tout langage de classe 0 est Récursivement Enumérable.
3. Donnez le principe de la preuve qui montre que tout langage de classe 1 est Récursif.