

Théorie des langages et de la compilation
Année académique 2006-2007
1ère licence en Informatique et
2ème année du grade de 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 (4pts)

1. Décrivez, pour chaque classe de la hiérarchie de Chomsky, tous les formalismes correspondant que nous avons étudiés au cours. Définissez les éléments qui les composent.
2. Quels formalismes a-t-on utilisé pour construire et faire fonctionner un compilateur ? Expliquez.
3. Définissez, dans chacun des formalismes, le langage L des nombres réels en notation scientifiques (un nombre comprend optionnellement un signe, une partie entière obligatoire, un point décimal obligatoire, une partie fractionnaire obligatoire et un exposant optionnel. La partie exposant, si elle, existe comprend un E , un signe optionnel et un nombre entier).

Question 2 (3pts)

Voici un extrait de la grammaire du langage PASCAL qui définit les expressions :

$$E \rightarrow E \text{ OpArith } E$$

$$E \rightarrow E \text{ OpBool } E$$

$$E \rightarrow E \text{ OpComp } E$$

$$E \rightarrow \text{NOT } E$$

$$E \rightarrow (E)$$

$$\text{OpArith} \rightarrow + \mid - \mid * \mid / \mid \text{DIV} \mid \text{MOD}$$

$$\text{OpBool} \rightarrow \text{AND} \mid \text{OR}$$

$$\text{OpComp} \rightarrow = \mid < > \mid < \mid < = \mid > \mid > =$$

La priorité des opérateurs est fixée comme suit, par ordre croissant de priorité (deux opérateurs sur la même ligne ont la même priorité) :

NOT

*, /, DIV, MOD, AND

+, -, OR

=, <>, <, <=, >, >=

L'associativité est fixée à gauche.

Cette grammaire est-elle LL(1)? Respecte-t-elle la priorité et l'associativité des opérateurs? Si oui, justifiez. Si non, transformez-la en grammaire LL(1) qui respecte la priorité et l'associativité des opérateurs, en détaillant les étapes de la transformation. Donnez les tables de l'analyseur LL(1) correspondant.

Question 3 (3pts)

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

$$S' \rightarrow S$$
$$S \rightarrow SxSy$$
$$S \rightarrow z$$
$$S \rightarrow \varepsilon$$

Simulez-en le fonctionnement sur la chaîne $xzxyy$.

Question 4 (3pts)

Expliquez 2 méthodes pour déterminer si 2 langages réguliers (décrits sous forme d'automates finis déterministes) sont équivalents. Expliquez les algorithmes utilisés.

Question 5 (3pts)

Lors de la compilation ou l'exécution d'un programme impératif classique (comme JAVA ou C++), à quels moments et pourquoi a-t-on besoin d'une pile (*stack*)? Détaillez votre réponse.