

Contrôle continu - Compilation (L3 Info) Durée : 2h - documents interdits

Conventions

- Axiome : symbole non terminal à gauche de la première production de la grammaire
- Symboles non terminaux : lettres *MAJUSCULES ITALIQUES*
- Symboles terminaux : lettres **minuscules true-type** ou caractères spéciaux simples

Exercice 1 - Vrai ou faux (4 pts)

Pour chacune des affirmations suivantes dites si elle est vraie ou fausse en justifiant brièvement et de façon convaincante votre réponse (seules les réponses accompagnées d'une justification seront prises en compte).

1. Les grammaires hors-contexte récursives à gauche ne sont pas LL
2. Tout langage hors-contexte est reconnaissable par un automate à pile déterministe
3. Les grammaires non factorisées à droite ne sont pas LL
4. Les grammaires hors-contexte récursives sont ambiguës

Exercice 2 - Écriture de grammaire (6 pts)

Une grammaire hors-contexte est un quadruplet $\langle N, \Sigma, P, S \rangle$ où N est l'alphabet non terminal (un symbole non terminal sera représenté par une lettre majuscule), Σ est l'alphabet terminal (un symbole terminal sera représenté par une lettre minuscule), P est l'ensemble des règles de production et S est l'axiome. Voici un exemple de grammaire hors-contexte : $\langle \{ S \}, \{ a, b \}, \{ S \rightarrow aSb, S \rightarrow \varepsilon \}, S \rangle$

1. Ecrire une grammaire G qui permet de générer toutes les grammaires hors-contexte. Le vocabulaire terminal de G est le suivant :
a ... z (symboles terminaux) A ... Z (symboles non terminaux) $\varepsilon, \rightarrow, ', ', <, >, \{, \}$.
Votre grammaire doit être la plus simple possible, indiquez à quoi correspond chaque symbole non terminal de cette dernière.
2. Dessiner l'arbre de dérivation que la grammaire G associe au mot suivant :
 $\langle \{ S \}, \{ a, b \}, \{ S \rightarrow aSb, S \rightarrow \varepsilon \}, S \rangle$
3. Est ce que G permet de vérifier que l'axiome appartient au vocabulaire non terminal? Si ce n'est pas le cas, est il possible de modifier G de manière qu'elle le vérifie? à quoi ressemblerait la grammaire transformée?

Exercice 3 - Analyse LL(1) 6 pts

Soit la grammaire G_1 :

- 1 $A \rightarrow BA$
- 2 $A \rightarrow \varepsilon$
- 3 $B \rightarrow CDE$
- 4 $C \rightarrow a$
- 5 $C \rightarrow b$
- 6 $D \rightarrow c$
- 7 $D \rightarrow \varepsilon$
- 8 $E \rightarrow e$
- 9 $E \rightarrow f$

1. Décrivez en une phrase ou deux ou en notation mathématique le langage engendré par G_1
2. Calculez les ensembles PREMIER et SUIVANT de G_1
3. Construisez la table LL(1) de G_1
4. Simulez l'analyse $LL(1)$ du mot **acebf**. Les configurations sont représentées par le triplet (w, α, y) , où w est la partie de la bande de lecture qui commence au caractère sous la tête de lecture (ce qui reste à analyser), α est la pile et y est la séquence de symboles de sortie (numéros des productions appliquées).

Exercice 4 - Récursivité à gauche 4 pts

Soit la grammaire G_2 :

$$\begin{aligned} X &\rightarrow YZ \mid X \mathbf{b} \\ Y &\rightarrow ZZ \mid XZ \\ Z &\rightarrow XZ \mid YX \mid \mathbf{a} \end{aligned}$$

1. Ecrire une grammaire G'_2 non récursive à gauche telle que $L(G_2) = L(G'_2)$, en respectant l'ordre suivant sur les symboles non terminaux : $X < Y < Z$.