

Langages formels (LI 3242)
Contrôle continu
Devoir sur table n°2 & Examen
Aucun document autorisé.
Durée : 2 heures.

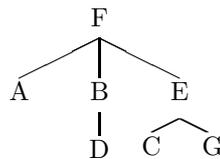
1. Soit la grammaire
- $$\begin{aligned}
 S &\rightarrow Z\$ \\
 Z &\rightarrow aZbC \\
 \del{Z} &\rightarrow \del{ab}C \quad Z \rightarrow ab \\
 C &\rightarrow cC \\
 C &\rightarrow c
 \end{aligned}$$

- (a) Représenter l'analyse *shift/reduce* du mot *aabbbc\$*.
 (b) Donner l'automate LR(0) de cette grammaire.

2. Soit la grammaire
- $$\begin{aligned}
 S &\rightarrow SN\ SV \\
 SN &\rightarrow Jean \mid Marie \\
 SV &\rightarrow V_i \\
 SV &\rightarrow V_a\ que\ S \\
 V_a &\rightarrow\ croit \mid\ pense \\
 V_i &\rightarrow\ dort \mid\ ronfle
 \end{aligned}$$

- (a) Mettre cette grammaire sous forme normale de Chomsky.
 (b) Donner la table des sous-chaînes bien formées produite par un parsing CYK de la phrase *Marie croit que Jean ronfle*.

3. On suppose que l'on a défini un langage pour décrire des arbres. Par exemple, l'arbre suivant pourrait être décrit par le « mot » suivant :



racine: F ;
 F : A, B, E ;
 A : ;
 B : D ;
 D : ;
 E : C, G ;
 C : ;
 G : ;

- (a) Proposer une grammaire qui reconnaisse ce langage.
 (b) Ajouter des actions sémantiques (syntaxe à la `ply` ou `yacc`) pour calculer le nombre maximum de fils d'un nœud dans l'arbre.
 (c) Ajouter des actions sémantiques pour construire une représentation de l'arbre qui permette, une fois le parsing réalisé, d'afficher l'arbre sous une forme linéaire, par exemple sous la forme $F(A, B(D), E(C,G))$.
 (d) [Bonus] Ajouter le code pour produire la forme linéaire.
4. Proposer l'algorithme qui, à partir d'une table des sous-chaînes bien formées remplie (par exemple par une analyse Earley), produit le ou les arbres syntaxiques représenté(s) dans la table.