Langages formels (LI324)
Contrôle continu & Examen final
Aucun document autorisé.
Durée : 2 heures.

1. Soit la grammaire suivante : $S \to S + S$
   $S \to S * S$
   $S \to a$
   $S \to (S)$

   (a) En considérant le mot $a * a + a$, donner le contenu de la table des sous-chaînes bien formées qui serait obtenue en mettant en œuvre la méthode de parsing tabulaire ascendant, avec les règles init, scan et comp.[5]. On pourra se dispenser de faire figurer dans la table les règles actives non pertinentes, mais il est important que toutes les règles inactives soient indiquées.

   (b) Donner les arbres syntaxiques (y compris partiels) correspondant à cette analyse.

   (c) Donnez l'algorithme qui, étant donnée une table ainsi remplie, fournit le ou les arbres syntaxiques correspondant à l'analyse. On ne demande pas l'algorithme entièrement détaillé, mais les principes généraux doivent être spécifiés précisément.

2. On s'intéresse aux groupes nominaux de la forme illustrée par (1).

   (1) a. le jour de la rentrée de la classe
   b. le jour de la rentrée de ce mois

   (a) Proposer une grammaire algébrique qui engendre ces deux syntagmes en fournissant des arbres syntaxiques qui distinguent les deux cas illustrés. Donner les arbres syntaxiques.

   (b) La grammaire précédente est vraisemblablement ambigüe, et pour éviter cette ambigïté, on peut s'inspirer du traitement "naïf" de la transitivité verbale, en distinguant des noms "intransitifs", qui ne sous-catégorisent pas de PP, et des noms "transitifs", qui en demandent un. Une telle grammaire pourrait contenir les règles $N' \to N_i$ et $N' \to N_t\ PP$.
   Quels sont les avantages et inconvénients d'une telle méthode pour les exemples (1) ?

   (c) Il est naturel de traiter les modifieurs (non sous-catégorisés) comme des adjonctions, par exemple avec des règles de la forme $N' \to N'\ PP$.

      i. En prenant un exemple simple (par exemple le syntagme *le chat de la voisine*), montrer les problèmes posés par ce genre de règle pour le parsing descendant.

      ii. Comment peut-on transformer la grammaire pour résoudre le problème précédent ?

      iii. En supposant la grammaire transformée pour permettre un parsing descendant des adjonctions, esquisser l'algorithme qui permettrait de reconstituer, après l'analyse, un arbre syntaxique satisfaisant (c'est-à-dire conforme à la grammaire avant sa transformation).

[5] On rappelle que l'application de la règle init consiste à insérer systématiquement les règles de la forme $A \to \bullet\beta$; la règle scan s'applique dans les situations de "shift" ; la règle comp est la règle fondamentale du parsing tabulaire *à la* Earley.