Analyse syntaxique automatique

LI035, 2ème semestre 2000/2001

Laura Kallmeyer

Examen du 7.6.2001

Contrôle continu : questions 2–6, contrôle terminal : questions 1–6.

(La question 6 est une question supplémentaire, c'est-à-dire pour obtenir 20 une réponse à 6 n'est pas nécessaire, mais 6 donne des points supplémentaires qui peuvent compenser d'autres erreurs.)

1. Parsing top-down, bottom-up et coin-gauche

   (a) Considérez une grammaire hors contexte avec symbole initial S, V = {S, NP1, NP2, VP, Det, N, V}, T = {le, chat, regarde, Jean} et les productions

   1. S → NP1 VP   2. NP1 → Det N   3. Det → le   4. N → chat
   5. VP → V NP2   6. V → regarde   7. NP2 → Jean

   Pour analyser *le chat regarde Jean*, dans quel ordre les productions sont-elles appliquées dans un algorithme

   i. top-down de gauche à droite?
   ii. bottom-up de gauche à droite?
   iii. coin-gauche?

   (b) Considérez les deux grammaires hors contexte suivantes:

   - $G_1 = \langle\{S, A, B\}, \{a, b\}, S, \{S \rightarrow A\ B, A \rightarrow a\ A, A \rightarrow a, B \rightarrow b\ B, B \rightarrow b\}\rangle$
   - $G_2 = \langle\{S, A, B\}, \{a, b\}, S, \{S \rightarrow A\ B, A \rightarrow A\ a, A \rightarrow a, B \rightarrow B\ b, B \rightarrow b\}\rangle$

   Quelles sont les langues générées par ces deux grammaires?

   Quels algorithmes de parsing parmi top-down et coin-gauche

   i. peut-on utiliser pour G_1 et G_2 respectivement?
   ii. permettent un parsing déterministe sans faire du backtracking ou poursuivre plusieures possibilités en parallèle pour G_1 et G_2 respectivement?

   Jusitifiez vos réponses.

2. La relation $FIRST$.

   Considérez une grammaire hors contexte avec symbole initial S, V = {S, X, Y, A, B, C}, T = {a, b, c} et les productions

   S → X Y   X → A B   X → B C   Y → B C   Y → C
   A → a   A → a A   A → ϵ B → b   B → ϵ C → c   C → c C

   Calculez pour cette grammaire les ensembles suivants :

   (a) $FIRST$(a A)
   (b) $FIRST$(B)
   (c) $FIRST$(X)
   (d) $FIRST$(S)

3. Lookahead dans l'algorithme de Earley :

   (a) Pour intégrer un lookahead dans l'algorithme Earley-2 du cours, quelle procédure faut-il modifier? Donnez la nouvelle version de cette procédure.

   (b) Considérez une grammaire hors contexte avec symbole initial NP, V = {NP, AP}, T = {det, n, a} et les productions (dans cet ordre)

   NP → det n   NP → det AP n   NP → n   AP → a   AP → a AP

   Le dictionnaire D contient des entrées ⟨*le*, det⟩, ⟨*petit*, a⟩, ⟨*chien*, n⟩.

   Montrez pour un parsing de la phrase *le petit chien* avec cette grammaire dans quel ordre quels arcs sont entrés par quelles procédures (`expand`, `shift` ou `complete`) dans le chart si on utilise

i. l'algorithme Earley-2 sans lookahead (comme au cours)

ii. l'algorithme Earley-2 avec lookahead que vous venez de développer.

4. Des grammaires LR(0)

   Les deux grammaires suivantes, sont-elles LR(0)? Justifiez votre réponse en montrant que la grammaire satisfait ou ne satisfait pas la définition d'une grammaire LR(0).

   (a) $G_1 = \langle V, T, P, S \rangle$ avec S = NP, V = {NP, AP}, T = {det, n, a} et les productions
   1. NP → det n 2. NP → det AP n 3. AP → a 4. AP → AP a

   (b) $G_2 = \langle V, T, P, S \rangle$ avec S = NP, V = {NP, AP}, T = {det, n, a} et les productions
   1. NP → det n 2. NP → det AP n 3. AP → a 4. AP → a AP

5. Table de parsing LR.

   Construisez une table de parsing LR pour la grammaire G_2 de la question précédante en utilisant l'algorithme `CLR(1)` du cours. Montrez en détail quels arcs se trouvent dans quels états.

   (Comme NP est le symbole initial, on ajoute la production S' → NP.)

6. Des grammaires LR(k) :

   Démontrez que la grammaire suivante n'est pas LR(1).

   $G = \langle V, T, P, S \rangle$ avec S = S, V = {S, A, B}, T = {a, b} et les productions

   S → A B A → a B A → a B → b A B → b