Analyse syntaxique automatique

LI035, 2ème semestre 2000/2001

Laura Kallmeyer

Partiel du 2.4.2001

1. Considérez une grammaire hors contexte avec symbole initiale S, V = {S, NP, VP, V}, T = {Jean, Marie, aime} et les productions

   1. S $\rightarrow$ NP VP 2. NP $\rightarrow$ Jean 3. NP $\rightarrow$ Marie 4. VP $\rightarrow$ V NP 5. V $\rightarrow$ aime

   Pour analyser *Jean aime Marie*, dans quel ordre les productions sont-elles appliquées dans un algorithme

   (a) top-down de gauche à droite?

   (b) bottom-up de gauche à droite?

   (c) coin-gauche?

2. Considérez la grammaire hors contexte $G = \langle V, T, S, P \rangle$ avec V = {S, A, B, C}, T = {a, b, c}, S = S et les productions P = {S $\rightarrow$ A B, B $\rightarrow$ B C, A $\rightarrow$ a, B $\rightarrow$ b, C $\rightarrow$ c}.

   Quelle est la langue $L(G)$ générée par cette grammaire?

   Quels sont les problèmes d'un parsing coin-gauche non-déterministe avec cette grammaire si, dans l'algorithme principal, on fait (dans le cas ou un `move` ne donne rien)

   (a) d'abord un `reduce`, et si cela ne donne rien, un `remove` (version du cours)?

   (b) d'abord un `remove`, et si cela ne donne rien, un `reduce`?

   Pour chacun de ces deux cas, donnez comme exemple une phrase $w \in L(G)$ que l'algorithme ne reconnaît pas.

3. Considérez les trois grammaires hors contexte suivantes:

   - G_1 = ⟨{S, A, B}, {a, b, c},S, {S $\rightarrow$ A S B, S $\rightarrow$ c, A $\rightarrow$ a, B $\rightarrow$ b}⟩
   - G_2 = ⟨{A, B, C, D}, {a, b, c, d},A, {A $\rightarrow$ A B D, A $\rightarrow$ a, B $\rightarrow$ B C, B $\rightarrow$ b, C $\rightarrow$ c, D $\rightarrow$ d}⟩
   - G_3 = ⟨{S, A, B, C}, {a, c},S, {S $\rightarrow$ A B C, A $\rightarrow$ a, B $\rightarrow$ A, C $\rightarrow$ c}⟩

   Quels algorithmes de parsing parmi top-down, bottom-up et coin-gauche

   (a) peut-on utiliser pour G_1, G_2 et G_3 respectivement?

   (b) permettent un parsing déterministe sans faire du backtracking ou poursuivre plusieures possibilités en parallèle pour G_1, G_2 et G_3 respectivement?

   Jusitifiez vos réponses.

4. Chart Parsing bottom-up

   (a) Expliquez en quelques phrases l'algorithme de Cocke, Kasami et Younger.

   (b) Quelle est la différence principale entre l'algorithme général de chart-parsing bottom-up vu au cours et l'algorithme de Cocke Kasami Younger? Autrement dit, pourquoi l'algorithme de Cocke Kasami et Younger est-il plus efficace?

5. Look-ahead

   (a) Comment est-ce qu'on peut intégrer un look-ahead dans un parsing coin-gauche?

(b) Intégration d'un look-ahead dans un algorithme top-down : Voici un algorithme `recognize` top-down breadth-first (le dictonnaire n'est pas séparé de la grammaire, c'est-à-dire les mots $w_1, \ldots, w_n$ d'une phrase sont des terminaux de la grammaire) :

Algorithme recognize

`top-down / gauche-droit / breadth-first`

```
Données globales :  CFG G = ⟨V,T,S,P⟩
Input :
 une phrase w = [w1 ... wn],  n ≥ 0
 une dérivation d (valeur initiale [S])
Output :  true/false
Méthode :
   if d = [] ∧ w = []
   then return(true)
   else
      if d = []
      then return(false)
      else
         if first(d) = first(w)
         then recognize(rest(w), rest(d))
         else
            if first(d) ∉ T
            then
              for all X → α from P:
                 if X =first(d)
                 then
                    if recognize(w, α+rest(d))
                    then return(true)
              return(false)
            else return(false)
```

Améliorez cet algorithme en y intégrant l'utilisation d'un look-ahead.