

Analyse syntaxique

1

On traite des arbres binaires; aux feuilles sont attachées des valeurs entières. On décide de noter $(A . B)$ l'arbre binaire composé des sous-arbres A et B ; si un arbre est réduit à une feuille, il est noté par sa valeur; s'il est vide, il est noté nil .

Ainsi la notation $((1 . 2) . (3 . nil))$ correspond à l'arbre:

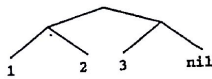


Figure 1 : Arbre $((1 . 2) . (3 . nil))$.

On traite également des listes. On décide de noter une liste (A, B, \dots, E) où A, B, \dots, E sont les éléments successifs de la liste. Les éléments peuvent être des listes ou des arbres et donc aussi des valeurs entières ou nil . Ainsi, $((3 . 2), nil, nil, 4)$ est une liste.

Pour avoir une représentation commune aux arbres et aux listes, on décide que la liste (A, B, \dots, E) sera représentée par l'arbre $(A . (B . (\dots . (E . nil) \dots)))$.

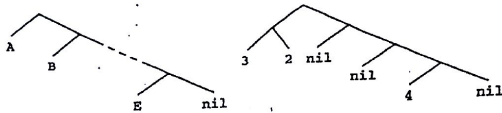


Figure 2 : Arbre (A, B, \dots, E) et Arbre $((3 . 2), nil, nil, 4)$.

1. Donner une grammaire décrivant la syntaxe des arbres et des listes; voici des exemples de notations correctes:

- $(1, (2 . 2), 3)$
- nil
- 13
- $((3 . 4), (2, (3 . 5), nil))$
- (1)

1

2. Peut-on réaliser un analyseur LL(1) pour la grammaire présentée à la question précédente ?
3. Dans l'analyseur LL(1) ainsi construit, ajouter des fonctions sémantiques permettant de construire la notation binaire d'une phrase du langage (c'est-à-dire avec la notation pointée uniquement).
4. Peut-on réaliser un analyseur SLR(1) pour la grammaire présentée à la question précédente ?
5. En supposant qu'on dispose d'un analyseur SLR(1), ajouter des fonctions sémantiques permettant de construire la notation binaire d'une phrase du langage (c'est-à-dire avec la notation pointée uniquement).

2 Analyse ascendante

Soit la grammaire suivante:

$$\begin{array}{ll}
 S \rightarrow G = D & V_t = \{ =, *, idf \} \\
 & D \\
 G \rightarrow * D & V_n = \{ G, D \} \\
 & idf \\
 D \rightarrow G & P = \rightarrow
 \end{array}$$

Peut-on construire des analyseurs SLR(1), LR(1) et LALR(1) ?

2