

Sémantique et TDL. Grammaires Attribuées

1 Evaluation d'un nombre entier

On considère la grammaire des nombres entiers :

1. $S \rightarrow I$
2. $I \rightarrow \mathbf{c}$
3. $I \rightarrow I \mathbf{c}$

où S est l'axiome et c un chiffre décimal.

On désire décrire la sémantique du calcul de la valeur de l'entier analysé, en utilisant une grammaire attribuée.

1. Représenter et décorer l'arbre syntaxique associé à l'analyse du nombre 546.
2. Quel(s) attribut(s) faut-il associer aux symboles de cette grammaire ?
3. En déduire les règles sémantiques associées aux règles de production décrivant cette sémantique.
4. En déduire un ordre d'évaluation des règles sémantiques.
5. La grammaire précédente n'est pas LL(1) car elle est récursive à gauche. Donner une grammaire non récursive à gauche acceptant le même langage. Proposer une sémantique attribuée pour cette nouvelle grammaire. Pour cela, vous pouvez soit définir une nouvelle sémantique, soit adapter la sémantique définie précédemment.

2 Construction d'un arbre abstrait

On considère la grammaire décrivant les instructions du langage BLOC :

1. $S \rightarrow B$
2. $B \rightarrow \{ LI \}$
3. $LI \rightarrow I LI$
4. $LI \rightarrow \Lambda$
5. $I \rightarrow T \mathbf{id} = E ;$
6. $I \rightarrow \mathbf{id} = E ;$
7. $I \rightarrow \mathbf{if} (E) B \mathbf{else} B$
8. $I \rightarrow \mathbf{if} (E) B$
9. $I \rightarrow \mathbf{while} (E) B$
10. $T \rightarrow \dots$
11. $E \rightarrow \dots$

On désire décrire la sémantique de construction de l'arbre abstrait, en utilisant une grammaire attribuée. Cette sémantique étends celle des expressions

étudiée en cours. Le non-terminal E possède un attribut sémantique synthétisé *ast* de type **Expression**. Le non-terminal T possède un attribut sémantique synthétisé *ast* de type **Type**.

1. Définir l'arbre abstrait que vous souhaitez construire pour le programme suivant. Représenter l'arbre syntaxique associé à l'analyse de ce programme. Indiquer les parties de l'arbre abstrait qui doivent être construites pour chaque noeud de l'arbre syntaxique.

```
{
    int i = 1;
    int j = 1;
    while ( i < 5 ) {
        j = i * j;
        i = i + 1;
    }
    print j;
}
```

2. Quel(s) attribut(s) sémantique(s) faut-il associer aux symboles de cette grammaire ?
3. En déduire les règles sémantiques nécessaires.
4. En déduire un ordre d'évaluation des règles sémantiques.
5. La grammaire précédente n'est pas LL(1) car elle n'est pas factorisée à gauche. Donner une grammaire factorisée à gauche acceptant le même langage. Adapter la sémantique définie précédemment à cette nouvelle grammaire.