UNIVERSITÉ DE TUNIS EL MANAR
FACULTÉ DES SCIENCES DE TUNIS
————
DÉPARTEMENT DES SCIENCES DE L'INFORMATIQUE

2019-2020

COMPILATION
T.D.N°3
(Analyse Sémantique)

Section : IF4
Enseignantes : Yousra Hlaoui et Faten Abbessi

Exercice 1

Soit la Définition Dirigée par la Syntaxe (DDS) suivante :

Production	***Règle sémantique***
$S' \rightarrow S$	$S^{(0)}.x := 0$ $Retourner\ S^{(0)}.y$
$S \rightarrow a\ S\ a$	$S^{(1)}.x := 2 * S^{(0)}.x + 1$ $S^{(0)}.y := S^{(1)}.y + 1$
$S \rightarrow b\ S\ c$	$S^{(1)}.x := S^{(0)}.x + 5$ $S^{(0)}.y := 3 * S^{(1)}.y - 10$
$S \rightarrow d$	$S^{(0)}.y := S^{(0)}.x - 10$

Donner l'arbre décoré du mot $aabadacaa$. Quelle est la valeur retournée ?

Exercice 2

On considère la grammaire suivante qui décrit des expressions arithmétiques préfixées :

$$exp \rightarrow\ chiffre \mid (\ +\ exp\ exp\ ) \mid (\ *\ exp\ exp\ )$$
$$chiffre \rightarrow\ 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

1. Donner une DDS pour cette grammaire de façon à calculer la valeur de l'expression analysée.
2. Donner l'arbre décoré du mot $(+1(*3(+23)))$ et l'évaluer.

Exercice 3

Soit la grammaire suivante :

$$E \rightarrow E + T \mid T$$
$$T \rightarrow nb\,.\,nb \mid nb$$

Cette grammaire engendre des expressions arithmétiques formées de constantes entières et réelles et de l'opérateur $+$. Lorsque l'on additionne deux entiers, le résultat est un entier, dans les autres cas c'est un réel.

1. Écrire une définition dirigée par la syntaxe (DDS) qui détermine le type de chaque sous-expression.
2. Donner un arbre syntaxique décoré pour la chaîne $5 + 6.1 + 3$

Exercice 3 **Examen Mai2018**

On s'intéresse à des arbres avec des nœuds étiquetés par des entiers de 0 à 9. Ces arbres sont représentés par une expressions parenthésée dans laquelle un nœud est représenté par son étiquette suivie de la liste de ses fils entre parenthèses.

Exemple : L'expression $1(2(3())4())$ représente l'arbre ayant 1 comme racine qui elle même possède deux fils 2 et 4. Le nœud 2 possède 3 comme fils. (La représentation graphique de cet arbre est donnée en (a) de l'exemple 1 de l'exercice 2)

Le langage des arbres est généré par la grammaire G suivante [1] :

$$R \rightarrow N$$
$$N \rightarrow E\,(\,L\,)$$
$$L \rightarrow N\,L$$
$$L \rightarrow \varepsilon$$
$$E \rightarrow 0 \mid 1 \mid \ldots \mid 9$$

On cherche, dans ce qui suit, à répondre à des questions concernant des arbres générés par la grammaire G. On vous demande, alors, de définir des attributs ainsi que des actions sémantiques sur cette grammaire permettant de calculer la valeur de ces attributs. Tous les attributs sont à valeurs entières. Pour calculer la valeur des attributs vous pouvez utiliser les opérations arithmétiques, les opérations booléennes, la fonction binaire $max(x, y)$ et le prédicat $egal(x, y)$ qui vaut 1 si $x = y$ et vaut 0 sinon.
Pour chaque attribut défini, il faut indiquer :

a. ce qu'il représente,

b. s'il est hérité ou synthétisé,

c. sa valeur pour chaque nœud de l'arbre de la question (1).

1. La première règle n'est pas nécessaire, mais elle sera utile pour la suite

1. Construire l'arbre de dérivation correspondant à l'arbre $1(2(3())4())$.
2. Calculer le nombre de nœuds d'un arbre.
   Remarque : *Pour chacune des questions suivantes, vous pouvez utiliser les attributs que vous avez définis pour répondre aux questions qui la précèdent.*
3. Quel est le degré de chaque nœud ?
4. Quelle la profondeur de chaque nœud ?
5. Quelle est la hauteur de l'arbre
6. Calculer le nombre de feuilles d'un arbre.
7. Chaque nœud N a-t-il le même degré que la racine ?
8. L'arbre est-il équilibré ? (Tous les nœuds internes sont de même degré)

Exercice 4 *(4 points)*

Soit la grammaire suivante :

$$
\begin{array}{l}
P \rightarrow D \ ; \ I \\
D \rightarrow D \ ; \ D \mid \textbf{id} \ : \ T \\
T \rightarrow \textbf{caractère} \mid \textbf{entier} \mid \textbf{booléen} \mid \textbf{tableau} \ [\ \textbf{nb} \ ] \ \textbf{de} \ T \mid \uparrow T \\
I \rightarrow I \ ; \ I \mid \textbf{id} := E \mid \textbf{si} \ E \ \textbf{alors} \ I \mid \textbf{tant que} \ E \ \textbf{faire} \ I \\
E \rightarrow \textbf{littéral} \mid \textbf{nb} \mid \textbf{id} \mid E = E \mid E \ \textbf{mod} \ E \mid E \ [\ E \ ] \mid E \uparrow
\end{array}
$$

1. Donner une DDS permettant de déterminer le type des types T, des expressions E et celui des instructions I.
2. À l'aide de cette DDS calculer les types des expressions dans le fragment du programme ci-dessous. Spécifier le type à chaque nœud de l'arbre de l'analyse syntaxique.

```
i : entier ; j : entier ; k : entier ;
tant que i = j mod 2 faire
    k := i ;
    i := j mod i ;
    j := k
```

3. En utilisant les fonctions de génération de code à 3 adresses pour machine à registre, écrivez la séquence de code produite pour le fragment du programme ci-dessus.

Exercice 5 **Examen Mai2017**

Soit la grammaire suivante :

$$
\begin{array}{l}
P \rightarrow D \ ; \ I \\
D \rightarrow D \ ; \ D \mid \textbf{id} \ : \ T \\
T \rightarrow \textbf{caractère} \mid \textbf{entier} \mid \textbf{booléen} \mid \textbf{tableau} \ [\ \textbf{nb} \ ] \ \textbf{de} \ T \mid \uparrow T \mid \ T \rightarrow T \\
I \rightarrow I \ ; \ I \mid \textbf{id} := E \mid \textbf{si} \ E \ \textbf{alors} \ I \mid \textbf{tant que} \ E \ \textbf{faire} \ I \\
E \rightarrow \textbf{littéral} \mid \textbf{nb} \mid \textbf{id} \mid E \ \textbf{mod} \ E \mid E \ [\ E \ ] \mid E \uparrow \mid E \ (\ E \ )
\end{array}
$$

— La production et l'action sémantique ci-dessous permettent la déclaration de types fonctionnels :

$$T \rightarrow T \ '\rightarrow' \ T \qquad \{T^{(0)}.type := T^{(1)}.type \rightarrow T^{(2)}.type\}$$

— Les apostrophes qui encadrent la flèche, servant de constructeur de fonction, permettent de la distinguer de la flèche utilisée comme méta-symbole dans une production

— L'expression E (E) représente l'application d'une fonction à un argument. La règle sémantique pour contrôler le type d'une application de fonction est :

$$E \rightarrow E\,(\,E\,) \qquad \{E^{(0)}.type := \textbf{si } E^{(2)}.type = s \textbf{ et } E^{(1)}.type = s \rightarrow t \textbf{ alors } t \textbf{ sinon } erreur\_de\_type\}$$

1. Donner une DDS permettant de déterminer le type des types T, des expressions E et celui des instructions I.
2. À l'aide de cette DDS calculer les types des expressions dans le fragment du programme ci-dessous. Spécifier le type à chaque nœud de l'arbre de l'analyse syntaxique.

```
f : entier → booléen;
i : entier ; j : entier ; k : entier ;
tant que f ( i ) faire
        k := i ;
        i := j mod i ;
        j := k
```