

TD numéro 2 : Syntaxe, analyse sémantique

Assembleur – Compilation, ENSIIE

Semestre 3, 2024–25

Exercice 1 : Arbres de Syntaxe Abstraite

1. Donner la syntaxe abstraite de l’instruction $y := 6 + (x * \text{tab}[i] - 2)$
2. Proposer une syntaxe concrète pour les séquences d’instructions. Attention : la syntaxe donnée dans la fiche “Syntaxe abstraite de Pseudo Pascal” est ambiguë.
3. On souhaite rajouter des boucles for à la syntaxe de Pseudo Pascal. Proposer une syntaxe similaire à celle en C.
4. Donner un exemple de suite d’instructions contenant au moins une boucle for et une séquence d’instructions :
 - a) en syntaxe concrète,
 - b) en syntaxe abstraite.

Exercice 2 : Analyse sémantique

Soit le programme Pseudo-Pascal suivant :

```
1 var x : integer;
2 function f(y : integer, x : array of integer) : integer;
3 var z : integer;
4 begin
5   x := y + z;
6   f := new array of integer [y]
7 end;
8
9 procedure g(u : integer);
10 begin
11   u := u + z;
12   g := u
13 end;
14
15 begin
16   x := f(new array of integer [42]);
17   f(g(x), new array of integer[24])
18 end.
```

1. Représenter la table des symboles. Parmi les informations stockées on indiquera la sorte (variable globale, variable locale, fonction, procédure), le type, la portée (globale ou locale à la fonction x) et la ligne de déclaration.
2. Quelle(s) erreur(s) l’analyse sémantique va-t-elle produire ? On fera un parcours systématique de l’AST.