Université Grenoble Alpes **UFR IM^{2}AG**
DU ISN-C2.3 **Année 2019-20**

Langages et compilation : sémantique statique

EXERCICES (2)

Exercice 1

On étend la syntaxe du langage pour autoriser l'initialisation des variables lors de leur déclaration :

D ::= var x : t := e | ...

Exemple de programme :

```
var x1 : Entier := 3 ;
var x2 : Booleen := x1 > 4 ;
if (x2) then x1 := ...
```

Complétez les règles de sémantique statique des déclarations pour prendre en compte cette nouvelle construction.

Exercice 2.

On considère le programme suivant :

```
begin
    var x : Entier ;
    var y : Entier ;
    x : = 3 ;   // (1)
    begin
        var x : Booleen ;
        var z : Entier ;
        x := z + y < 2 ;  // (2)
    end  ;
    y := x ;  // (3)
end
```

Donnez les fonction "environnement" aux points de contrôle **(1)**, **(2)**, et **(3)**.
Ce programme est-il correct du point de vue des règles de typage ?

Exercice 3.
Montrez que, d'après les règles de sémantique statique vues en cours :

1. le programme suivant est **correct** :

```
var x1 : Entier ;
var x2 : Booleen ;
proc p1 (x1 : Booleen)
     x2 := x1 ;
call p1 (x2) ;
```

2. le programme suivant est **incorrect** :

```
proc p1
     call p2 ;
proc p2
     call p1 ;
call p1 ;
```

Modifiez ces règles pour autoriser la définition de telles procédures mutuellement récursives.

Indication : il faut pour cela analyser deux fois chaque séquence de déclaration, une première fois pour construire l'environnement local qui lui est associé, et une seconde fois pour vérifier que cette séquence de déclaration est correcte lorsque l'on prend en compte cet environnement local.