



## TD 1 : Typage de programmes impératifs

### 1 Arbres syntaxiques

Pour les questions suivantes, référez-vous à la grammaire p. 20  
Donnez les arbres syntaxiques des expressions et instructions suivantes :

1.  $(x < 3) \ \&\& \ (y > 7)$
2.  $b == (y + z == 2)$
3. **while**  $(x < 5)$  **do** {  
     $x = x + 1$ ;  
     $y = x - y$   
}
4. **while**  $(x < 5) \ || \ b$  **do** {  
     $x = 2 * x - 1$   
};  
 $y = x * x$ ;

### 2 Typage d'expressions et instructions

Donnez les dérivations de typage pour les expressions de l'exercice 1 pour l'environnement  $[(int \ x); (int \ y), (bool \ b); (int \ z)]$ .

### 3 Règles de typage

Dans le langage C, il existe un opérateur conditionnel ternaire  $(e_1 ? e_2 : e_3)$ , où les  $e_1, e_2, e_3$  sont des expressions. Si la valeur de  $e_1$  est "vrai", cet opérateur renvoie la valeur de  $e_2$ , autrement la valeur de  $e_3$ . Donc, l'opérateur conditionnel se comporte presque comme un **if .. then .. else**, sauf que c'est une expression et non pas une instruction. Donnez des règles de typage pour le conditionnel et comparez avec les règles pour **if .. then .. else**.

### 4 Typage de programmes

Dans cet exercice, nous comparons le comportement des compilateurs de C et de Java sur des fonctions mutuellement récursives.

Sur le site Web, vous trouvez

- le fichier **pairimpair.c**, écrit en C, qui détermine si un nombre est pair par des appels récursifs entre une fonction **pair** et une fonction **impair**.
- le fichier **Main.java** qui contient (presque) le même programme en Java.

1. Qu'est-ce qui se passe si vous compilez le programme **pairimpair.c** tant que le profil de la fonction **impair** est commenté? <sup>1</sup>

```
//bool impair (int);
```

---

1. Le résultat peut dépendre du compilateur, utilisez gcc

2. Pourquoi est-ce que le problème disparaît si vous dé-commentez cette ligne ?
3. Quel est le comportement du compilateur Java (compiler avec `javac Main.java`, ensuite exécuter avec `java Main`).
4. Faites maintenant la vérification de type de la fonction `pair`, comme vue en cours.