

TD n°6

Sémantique du langage IMP

Exercice 1 Les expressions booléennes suivantes sont des abréviations. Écrivez-les en forme étendue.

1. $(5 > x) \vee (y = 3)$
2. $(7 \leq 3) \rightarrow (x = y)$
3. $(x \leq y) \vee (x \neq z)$

Interprétez ces expressions par rapport à l'affectation $\sigma = [x \mapsto 4, y \mapsto 6, z \mapsto -2]$

Exercice 2 Est-ce que les expressions suivantes sont valides? Si oui justifiez, sinon donnez une affectation qui rend l'expression fausse.

- $(x + 1) * (x + 1) > 0$
- $(x + 1) * (x + 1) \geq 0$
- $(x + y) * x > y \rightarrow x > y$
- $x > 0 \wedge (y \leq 0 \rightarrow x \leq 0) \rightarrow \neg(y \leq 0)$

Exercice 3 Le langage IMP contient l'instruction **if** b **then** p_1 **else** p_2 **fi**. Une instruction similaire existe en Java, mais sans **then** ni **fi**. Pourquoi?

Écrivez en Java un fragment de programme qui correspond au programme IMP suivant :

```
z := 0;
y1 := 0;
while y1 ≠ y do
  z := z + x;
  y1 := y1 + 1;
od
```

Exercice 4 Exécutez à la main les configurations suivantes.

1. $\langle y := x; x := 2; z := x * y, [x \mapsto 4] \rangle$
2. $\langle x := 0; y := 1; \text{if } x > y \text{ then } z := x \text{ else } z := y \text{ fi}, [] \rangle$
3. $\langle \text{while } x > 0 \text{ do } x := x + y; y := y + 1 \text{ od}, [x \mapsto 3, y \mapsto 5] \rangle$

Exercice 5 On imagine une nouvelle instruction "Arrête l'exécution" :

abort

Donnez la règle (ou les règles) de transition de cette instruction

Exercice 6 On imagine une nouvelle instruction "Répète un certain nombre de fois" :

do e times p od

où e est une expression arithmétique et p une liste d'instructions. Donnez la règle (ou les règles) de transition de cette instruction.

Exercice 7 Qu'est-ce qu'il faudrait changer si on voulait ajouter au langage une instruction **affiche** (e) ?