

TD 0 : Logique de Hoare

Rappelons les règles du système déductif associé à la logique de Hoare :

Règle (a). (Axiome)	$\{ p(t) \} \quad x = t \quad \{ p(x) \}$
Règle (b). (Composition)	$\{ p \} S1 \{ q \} ; \{ q \} S2 \{ r \}$ $\{ p \} \quad S1 ; S2 \quad \{ r \}$
Règle (c1). (Conditionnelle 1)	$\{ (p \wedge b) \} S \{ q \} ; \{ (p \wedge \neg b) \} \{ q \}$ $\{ p \} \quad \text{si } b \text{ alors } S \text{ fsi } \{ q \}$
Règle (c2). (Conditionnelle 2)	$\{ (p \wedge b) \} S1 \{ q \} ; \{ (p \wedge \neg b) \} S2 \{ q \}$ $\{ p \} \quad \text{si } b \text{ alors } S1 \text{ sinon } S2 \text{ fsi } \{ q \}$
Règle (d). (Conséquence)	$\{ (p \Rightarrow q) \} ; \{ q \} S \{ r \} ; \{ (r \Rightarrow s) \}$ $\{ p \} \quad S \quad \{ s \}$
Règle (e). (Invariant de boucle)	$\{ (p \wedge b) \} S \{ p \}$ $\{ p \} \quad \text{tant que } b \text{ faire } S \text{ fin-tq } \{ (p \wedge \neg b) \}$

1. Prouvez les formules suivantes :

$$\begin{aligned} & \{ (x = 3a) \} \\ & \quad a \leftarrow a + 2 ; \\ & \{ (x = 3a - 6) \} \end{aligned}$$

$$\begin{aligned} & \{ x = 2a \} \\ & \quad a \leftarrow a - 1 ; \\ & \{ x = 2.a + 2 \} \end{aligned}$$

$$\begin{aligned} & \{ (a \geq 0) \} \\ & \quad x \leftarrow 1 ; \\ & \{ (a = x.a) \} \end{aligned}$$

$$\begin{aligned} & \{ b \geq 0 \} \\ & \quad c \leftarrow b ; \\ & \quad p \leftarrow 0 ; \\ & \quad \text{tant que } (c > 0) \text{ faire} \\ & \quad \quad p \leftarrow p + b ; \\ & \quad \quad c \leftarrow c - 1 ; \\ & \quad \text{fin tant que} \\ & \{ p = b.b \} \end{aligned}$$

$$\begin{aligned} & \{ (z = a.b.y + y(a+1)) \} \\ & \quad b \leftarrow b + 1 ; \\ & \quad x \leftarrow a.b + 1 ; \\ & \{ (z = x.y) \} \end{aligned}$$

2. Prouvez la formule suivante :

```
{ entier(a) ∧ entier(b) ∧ (a ≥ 0) ∧ (b > 0) }
  q ← 0 ; r ← a ;
  tant que ( r ≥ b ) faire
    q ← q + 1 ; r ← r - b ;
  fin tant que
{ ( a = q.b + r ) ∧ (r ≥ 0) ∧ (r < b) }
```

3. Prouvez les formules suivantes :

```
{ entier(a) ∧ (a ≥ 0) }
  x ← a div 2 ;
  si (a mod 2 = 0) alors
    y ← 0 ;
  sinon
    y ← 1 ;
{ (a = 2x + y) ∧ (1 ≥ y ≥ 0) }

{ entier(b) ∧ (b ≥ 1) }
  c ← b ;
  f ← 1 ;
  tant que (c > 1) faire
    f ← f.c ;
    c ← c - 1 ;
  fin tant que
{ (f = b !) } factorielle

{ (entier(a) ∧ entier(b)) }
  x ← a ;
  y ← b ;
  si (x > y) alors
    p ← x - y ;
  sinon
    p ← y - x ;
{ p = |a-b| } (valeur absolue)
```

4. Considérons le programme suivant (a et b sont des entiers) :

```
Prog4 (a, b) : entier
Debut
  res ← 0 ;
  Si (a < 0) x ← -a ; sinon x ← a ;
  Si (b < 0) y ← -b ; sinon y ← b ;
  tant que (y > 0) faire
    res ← res + x ;
    y ← y-1 ;
  fin tant que
  retourne res ;
fin
```

- a. Que calcule ce programme ? Prouvez sa correction partielle.
- b. Prouvez sa terminaison.

5. Considérons le programme suivant (a et b sont des entiers) :

```
Prog5 (a, b) : entier
debut
  res ← 1 ;
  si (a < 0) x ← -a ; sinon x ← a ;
  si (b < 0) y ← -b ; sinon y ← b ;
  tant que (y > 0) faire
    res ← Prog4(res,x) ;
    y ← y-1 ;
  fin tant que
  retourne res ;
fin
```

- a. Que calcule ce programme ? Prouvez sa correction partielle.
- b. Prouvez sa terminaison.

6. Considérons le programme suivant. Ce programme calcule la partie entière de la racine carrée de la valeur absolue de a.

```
Prog6(a) : entier
Debut
  si (a < 0) x ← -a ; sinon x ← a ;
  y1 ← 0 ; y2 ← 1 ; y3 ← 1 ;
  tant que (y3 <= a) faire
    y1 ← y1+1 ;
    y2 ← y2+2 ;
    y3 ← y3+y2 ;
  fin tant que
  retourne y1 ;
fin
```

Rappel: $(x+1)^2 = x^2 + 2x + 1$

- a. Prouvez sa correction partielle. A savoir que le programme termine lorsque $(y1^2 \leq |a|$ et $(y1+1)^2 > |a|$)
- b. Prouvez sa terminaison.

7. Considérons le programme suivant. Ce programme trie un tableau Tab de Max entiers.

```

Prog7 (tab, Max) : Tableau d'entiers
debut
  for (i ← 0 ; i < Max-1, i++)
    for (j ← 0 ; j < Max-1-i, j++)
      si Tab[j] > Tab[j+1] alors
        Permutation (Tab, j, j+1);

  retourne Tab ;
fin

```

- a. Réécrivez le programme en changeant les boucles for en boucle while équivalente.
- b. Prouvez la formule suivante concernant la procédure permutation suivante.

```

Permutation (Tab, i, j)
debut
  { (Tab[j] = y) ∧ (Tab[i] = x) }
  temp ← Tab[i] ;
  Tab[i] ← Tab[j] ;
  Tab[j] ← temp ;
  { (Tab[j] = x) ∧ (Tab[i] = y) }

fin

```

- c. Prouvez la correction partielle du nouveau code.

8. Considérons le programme suivant. Ce programme calcule le PGCD(a, b). a et b sont des entiers. PGCD(a,b) est le plus grand diviseur commun à la valeur absolue de a et de b, par exemple PGCD(20,50) = 10. Prouvez sa correction partielle et sa terminaison.

Formules valides :

$$\{ (((x \geq 0) \wedge (y \geq 0)) \wedge (x > y)) \wedge (PGCD(x, y) = XXX) \Rightarrow (PGCD(x-y, y) = XXX) \}$$

$$\{ (((x \geq 0) \wedge (y \geq 0)) \Rightarrow (PGCD(x, x) = x)) \}$$

```

Prog8 (a, b) : entier
debut
  si (a < 0) x ← -a ; sinon x ← a ;
  si (b < 0) y ← -b ; sinon y ← b ;
  tant que (x ≠ y) faire
    si x > y alors
      x ← x - y ;
    sinon y ← y - x ;
  fin tant que
  retourne x ;
fin

```