

TD n 7 - Correction

Logique de Hoare

Exercice 1 Dire, dans chacun des cas suivants, si l'affectation satisfait la formule de Hoare.

- $[x \neq 0] \not\models fx = 0gy := x + 1 \ fy = 1g$
- $[x \neq 1] \not\models fx = 0gy := x + 1 \ fy = 1g$
- $[x \neq 0] \not\models fx = 0gy := x + 1 \ fy = 0g$
- $[x \neq 1] \not\models fx = 0gy := x + 1 \ fy = 0g$
- $[y \neq 0] \not\models fx = 1gy := x + 1 \ fy = 42g$
- $[x \neq 0; y \neq 1] \not\models fx < 2gy := x; x := 2; z := x - y \ fz = 2g$
- $[x \neq 5; y \neq 4] \not\models fy > 0g \text{ while } 1 > 0 \text{ do } x := x + y; y := y + 1 \text{ od } fx < 0g$

Correction :

- Oui.
- Oui, car la prémisse est fausse.
- Non, car la prémisse est vraie et la conclusion est fausse.
- Oui, car la prémisse est fausse.
- Oui, car la prémisse est fausse.
- Oui, il faut exécuter le programme pour le voir.
- Oui, car le programme ne se termine pas.

Exercice 2 Dire, dans chacun des cas suivants, si la formule de Hoare est valide, satisfaisable ou non satisfaisable.

- $fx < 2gy := x; x := 2; z := x - y \ fz = 2g$
- $fx > 0gx := 0; y := 1; \text{if } x > y \text{ then } z := y \text{ else } z := x \text{ fi } fz > 0g$
- $f(x > 0) \wedge (y > 0)g \text{ while } 1 > 0 \text{ do } x := x + y; y := y + 1 \text{ od } fx < 0g$
- $f(y > 0) \wedge (z = y)gx := 0; \text{while } z > 0 \text{ do } x := x + z; z := z - 1 \text{ od } f2 - x = (y - (y + 1))g$

Correction :

- Valide, car la valeur de z est le double de la valeur initiale de x .
- Pas valide : il suffit de prendre une affectation qui satisfait la précondition, la valeur finale de z est 0. La formule est satisfaisable, elle est satisfaite par toute affectation qui ne satisfait *pas* la précondition.
- Valide, car pour toute affectation initiale le programme ne se termine pas (il boucle).
- Valide, c'est la formule de Gauss.

Exercice 3 Est-ce que la formule de Hoare :

$$f\text{True}gx := e \ fx = eg$$

est valide pour toute expression e ?

Correction : D'après la Définition 26, la formule de Hoare est valide si et seulement si pour toute affectation on a $\{e\} = \{e'\}$ avec $e' = [x := e]$.

Cela est toujours le cas si x n'apparaît pas dans e (car e et e' ne diffèrent que sur x).

Mais si x apparaît dans e , cela n'est pas en général le cas (ex : $e = x + 1$), mais peut tout de même l'être pour des cas bien particuliers et peu intéressants (ex : $e = x$ ou $e = 3 - x - 2 - x$).

La formule n'est donc pas valide pour toute expression e .

Exercice 4 Dir , dans chacun d s cas suivants, pour qu ls programm s P la formul d Hoar st valid

1. $\neg \text{True} \wedge P \wedge \neg \text{True}$
2. $\neg \text{False} \wedge P \wedge \neg \text{False}$
3. $\neg \text{False} \wedge P \wedge \neg \text{True}$
4. $\neg \text{True} \wedge P \wedge \neg \text{False}$

Correction : Toute formule de Hoare dont la pré-condition est **False** ou dont la post-condition est **True** est valide quel que soit le programme.

1. Tout programme, car il y a **True** comme post-condition
2. Tout programme, car il y a **False** comme pré-condition
3. Tout programme, car il y a **False** comme pré-condition
4. Les programmes qui ne se terminent jamais (divergent).

Exercice 5 Trouvez une expression booléenne f qui rende valid s l s formul s suivant s. Cherchez la formule f la plus "simple" (dans quel sens ?)

1. $\neg f \wedge g \wedge x := x + 2 \wedge f \wedge x = 5 \wedge g$
2. $\neg f \wedge g \wedge x := y \wedge f \wedge x = y \wedge g$
3. $\neg f \wedge g \wedge y := 1; \text{ while } y > 0 \text{ do } y := y + x \text{ od } \neg f \wedge y = 2 \wedge g$
4. $f \wedge x > 2 \wedge g \wedge x := x + 2 \wedge \neg f \wedge g$
5. $f \wedge x > 0 \wedge x < 1 \wedge g \text{ if } x > 0 \text{ then } x := x + 1 \text{ else } x = -x \text{ fi } \neg f \wedge g$

Correction :

1. $x = 3$
2. $y = y$
3. $x = 0$: Si le programme termine on a que $y = 0$, et la postcondition n'est pas satisfaite. Il faut donc assurer que le programme ne termine pas.
4. $x > 4$
5. $x > 1$