

TD n 8 - Correction

Calcul de Hoare

Exercice 1 On rappelle la règle de la conditionnelle :

$$\frac{\bar{f}p \wedge fgS_1 \bar{f}qg \quad \bar{f}p \wedge : fgS_2 \bar{f}qg}{\bar{f}pg \text{ if } f \text{ then } S_1 \text{ else } S_2 \text{ fi } \bar{f}qg} \quad \text{où } p, q, f \in BExpr, S_1, S_2 \in Imp$$

Montrer sa correction.

Correction : Supposons $\bar{f}p \wedge fgS_1 \bar{f}qg$ et $\bar{f}p \wedge : fgS_2 \bar{f}qg$, et montrons :

$$\bar{f}pg \text{ if } f \text{ then } S_1 \text{ else } S_2 \text{ fi } \bar{f}qg$$

Soit donc une affectation σ telle que $\sigma \models p$ et $\sigma' := \llbracket \text{if } f \text{ then } S_1 \text{ else } S_2 \text{ fi} \rrbracket \sigma \neq ?$. Il s'agit de montrer $\sigma' \models q$. D'après la règle de transition associée à la conditionnelle, on a :

$$\sigma' = \begin{cases} \llbracket S_1 \rrbracket \sigma & \text{si } \sigma \models f \\ \llbracket S_2 \rrbracket \sigma & \text{sinon} \end{cases}$$

premier cas : $\sigma \models f$. Puisque $\sigma \models p$ par hypothèse, on a $\sigma \models p \wedge f$. Or $\bar{f}p \wedge fgS_1 \bar{f}qg$ est valide, et $\sigma' = \llbracket S_1 \rrbracket \sigma \neq ?$, donc $\sigma' \models q$.

deuxième cas : même chose en remplaçant S_1 par S_2 et f par $\neg f$.

Exercice 2 Étant donné une condition p et un programme S , écrire une formule de Hoare qui soit valide si et seulement si l'exécution de S boucle à partir de toute affectation vérifiant p , c'est-à-dire $\llbracket S \rrbracket \sigma = ?$ pour tout $\sigma \models p$.

Montrer par le calcul de Hoare qu'il n'y a pas de programme suivant boucle dans tout contexte où la valeur de x est strictement positive :

```
while (x>0) do
  x:=x+1
od
```

Correction : Une formule possible est $\bar{f}pgS \bar{f}Falseg$. Ici, il s'agit de montrer $\bar{f}x > 0gS \bar{f}Falseg$. On utilise la règle de la boucle :

$$\frac{\bar{f}p \wedge (x > 0)gx := x + 1 \bar{f}pg}{\bar{f}pg \text{ while } (x > 0) \text{ do } x := x + 1 \text{ od } \bar{f}p \wedge : (x > 0)g}$$

avec $p = (x > 0)$. Il faut montrer la prémisse : les formules $\bar{f}x + 1 > 0gx := x + 1 \bar{f}x > 0g$ et $p \wedge (x > 0) ! \bar{f}x + 1 > 0$ sont valides donc la prémisse est valide (règle de conséquence).

On conclut en remarquant que $p \wedge : (x > 0) ! \bar{f}False$ est valide (puis règle de conséquence).

Exercice 3 On considère l'instruction :

repeat S **until** f où $f \in BExpr, S \in Imp$

Écrire un programme IMP équivalent à cette instruction, en déduisant la règle d'inférence de sa correction.

Correction : L'instruction est équivalente au programme :

$S; \text{while} : f \text{ do } S \text{ od}$

De l'arbre de preuve :

$$\frac{\frac{\frac{f_q \wedge : fgS \ f_qg}{\overline{q \vdash q} \quad \overline{f_qg \text{ while} : f \text{ do } S \text{ od} \ f_q \wedge : fg} \quad \overline{q \wedge : f \vdash q \wedge f}}{fpgS \ f_qg} \quad \overline{f_qg \text{ while} : f \text{ do } S \text{ od} \ f_q \wedge fg}}{fpgS; \text{while} : f \text{ do } S \text{ od} \ f_q \wedge fg}$$

on déduit la règle :

$$\frac{fpgS \ f_qg \quad f_q \wedge : fgS \ f_qg}{fpg \text{ repeat } S \text{ until } f \ f_q \wedge fg} \quad \text{où } p, q, f \geq BExpr, S \geq Imp$$

La correction des règles utilisées dans l'arbre de preuve implique la correction de cette règle.

Exercice 4 Considérons le programme suivant :

```

y1 := 0;
y2 := 1;
y3 := 1;
while (y3 < x) do
  y1 := y1 + 1;
  y2 := y2 + 2;
  y3 := y3 + y2
od

```

Nous allons montrer avec le calcul d'Hoar qu'il calcule la racine carrée. Plus précisément, qu'il est partiellement correct par rapport à la pré-condition $(x \geq 0)$ et à la post-condition $(y_1 = y_1 \text{ et } (y_1 + 1)^2 > x)$.

Appelons S_0 le sous-programme constitué des 3 premières instructions, et S_1 sous-programme qui forme le corps du boucle **while**.

1. Calculer les valeurs que prennent les variables y_1, y_2 et y_3 au début des premières exécutions du corps du **while**. Conjecturer la valeur de y_2 et y_3 en fonction de y_1 . On note p la conjonction des 2 égalités et de $(y_1 = y_1 \text{ et } x)$.
2. p va être l'invariant du boucle. C'est-à-dire qu'on va montrer :

$$fpg \text{ while } (y_3 < x) \text{ do } S \text{ od} \ f_p \wedge : (y_3 < x)g$$

en utilisant la règle d'inférence du **while**. Écrire la prémisses qu'on doit utiliser, et la montrer par le calcul d'Hoar (indication : partir de la post-condition pour trouver les conditions intermédiaires).

3. Montrer par le calcul d'Hoar que p est vérifié après les 3 premières instructions du programme, sous la condition $(x \geq 0)$. C'est-à-dire, montrer que $f_x \geq 0gS_0 \ fpg$ est valide.
4. Conclure.

Correction :

1. $y_2 = 2 \ y_1 + 1$ et $y_3 = (y_1 + 1)^2$. Donc p est :

$$\underbrace{(y_2 = 2 \ y_1 + 1)}_{f_1} \wedge \underbrace{(y_3 = (y_1 + 1)^2)}_{f_2} \wedge \underbrace{(y_1 = y_1 \text{ et } x)}_{f_3}$$

2. La règle s'écrit ici :

$$\frac{\bar{f}p \wedge (y_3 = x)gS \bar{f}pg}{\bar{f}pg \text{ while } (y_3 = x) \text{ do } S \text{ od } \bar{f}p \wedge : (y_3 = x)g}$$

On part de la fin de S . Par simples applications de la règle de l'affectation, on a :

$$\begin{array}{lll} \bar{f}p'g & y_3 := y_3 + y_2 & \bar{f}pg \\ \bar{f}p''g & y_2 := y_2 + 2 & \bar{f}p'g \\ \bar{f}p'''g & y_1 := y_1 + 1 & \bar{f}p''g \end{array}$$

avec

$$\begin{array}{lll} p' = & f_1 & \wedge (y_3 + y_2 = (y_1 + 1)^2) \wedge f_3 \\ p'' = & (y_2 + 2 = 2 \quad y_1 + 1) & \wedge (y_3 + y_2 + 2 = (y_1 + 1)^2) \wedge f_3 \\ p''' = & (y_2 + 2 = 2 \quad (y_1 + 1) + 1) & \wedge (y_3 + y_2 + 2 = (y_1 + 2)^2) \wedge ((y_1 + 1)^2 = x) \end{array}$$

ce qui, par règle de composition, donne $\bar{f}p'''gS \bar{f}pg$.

Or $p''' \not\models (y_2 = 2 \quad y_1 + 1) \wedge (y_3 = (y_1 + 1)^2) \wedge (y_3 = x)$ c'est-à-dire $f_1 \wedge f_2 \wedge (y_3 = x)$. Mais cette formule est une conséquence de $p \wedge (y_3 = x)$. Donc par règle de conséquence on obtient la prémisse $\bar{f}p \wedge (y_3 = x)gS \bar{f}pg$.

3. En remontant, on obtient facilement :

$$\begin{array}{lll} \bar{f}(1 = 1) \wedge (1 = 1) \wedge (0 = x)g & y_1 := 0 & \bar{f}(1 = 2 \quad y_1 + 1) \wedge (1 = (y_1 + 1)^2) \wedge f_3g \\ \bar{f}(1 = 2 \quad y_1 + 1) \wedge (1 = (y_1 + 1)^2) \wedge f_3g & y_2 := 1 & \bar{f}f_1 \wedge (1 = (y_1 + 1)^2) \wedge f_3g \\ \bar{f}f_1 \wedge (1 = (y_1 + 1)^2) \wedge f_2g & y_3 := 1 & \bar{f}pg \end{array}$$

La première condition étant équivalente à $(x = 0)$, on conclut par règles de conséquence et de composition, que $\bar{f}x = 0gS_0 \bar{f}pg$ est valide.

4. En composant les 2 morceaux, on obtient la validité de :

$$\bar{f}x = 0gP \bar{f}p \wedge y_3 > xg$$

On conclut en utilisant la validité de la formule $p \wedge : (y_3 = x) \vdash (y_1 = y_1 = x) \wedge ((y_1 + 1) = (y_1 + 1) > x)$ et la règle de conséquence. Ouf.