Logique de Hoare

Marie Pelleau
marie.pelleau@univ-cotedazur.fr

Basé sur le cours de Frédéric Peschanski

Le langage

Version très simplifiée d'un langage de programmation

- Programme = Corps d'une méthode
- pas d'invocation
- types booléens, entiers et tableaux
- expressions arithmétiques et logiques de base
- instructions : affectations, séquencement, alternatives et boucles while

Triplets de Hoare

Triplet de Hoare

$\{P\}$ `prog` $\{Q\}$
où

- P est la précondition
- `prog` est un extrait de programme
- Q est la postcondition

Interprétation

En supposant P vraie avant exécution, et si on exécute `prog`, alors Q est vraie après exécution

Axiome d'affectation

Axiome d'affecation

$$\frac{}{\{Q[\mathrm{expr}/V]\}\ \mathtt{V} = \mathtt{expr}\ \{Q\}}\ (\mathit{aff})$$

Remarque

$Q[\mathrm{expr}/V] \stackrel{def}{=} Q$ en substituant toute occurence libre de V dans Q par expr (ou expr "écrase" V dans Q)

Axiome d'affectation

Axiome d'affecation

$$\frac{}{\{Q[\mathrm{expr}/V]\}\ \mathtt{V = expr}\ \{Q\}}\ (aff)$$

Exercice 1

On cherche la précondition la plus faible P telle que

$$\{P\}\mathrm{x} = \mathrm{y} + 1\{x = 3\}$$

$$\frac{}{\{x = 3[\mathrm{y} + 1/x]\}\ \mathrm{x} = \mathrm{y} + 1\ \{x = 3\}}\ (aff)$$

$$\frac{}{\{y + 1 = 3\}\ \mathrm{x} = \mathrm{y} + 1\ \{x = 3\}}\ (aff)$$

$$\frac{}{\{y = 2\}\ \mathrm{x} = \mathrm{y} + 1\ \{x = 3\}}\ (aff)$$

Axiome d'affectation

Axiome d'affecation

$$\frac{}{\{Q[\mathrm{expr}/V]\}\ \mathtt{V} = \mathtt{expr}\ \{Q\}}\ (aff)$$

Exercice 2

Trouver P et Q "intéressantes" telles que

$$\{P\}\mathtt{x} = -\mathtt{y}\{Q\}$$

Exercice 3

Trouver `prog` tel que

$$\{y = a\}\mathtt{prog}\{y = a \wedge x = 2 * a\}$$

Règle de séquencement

Règle de séquencement

$$\frac{\{P\}\ \mathtt{C}_1\ \{Q_1\} \quad \{Q_1\}\ \mathtt{C}_2\ \{Q_2\} \quad \dots \quad \{Q_{n-1}\}\ \mathtt{C}_n\ \{Q\}}{\{P\}\ \mathtt{C}_1; \dots; \mathtt{C}_n\ \{Q\}}\ (seq)$$

Exercice 4

Prouver que

$$\{\mathtt{true}\}\mathtt{z} = \mathtt{x}; \mathtt{z} = \mathtt{z} + \mathtt{y}; \mathtt{u} = \mathtt{z}\{u = x + y\}$$

$$\frac{\dfrac{}{\{x = x\}\mathtt{z} = \mathtt{x}\{z = x\}}\ (aff) \quad \dfrac{}{\{z = x\}\mathtt{z} = \mathtt{z} + \mathtt{y}\{z = x + y\}}\ (aff) \quad \dfrac{}{\{z = x + y\}\mathtt{u} = \mathtt{z}\{u = x + y\}}\ (aff)}{\{x = x\}\mathtt{z} = \mathtt{x}; \mathtt{z} = \mathtt{z} + \mathtt{y}; \mathtt{u} = \mathtt{z}\{u = x + y\}}\ (seq)$$

$$x = x \iff \mathtt{true}$$

Modus ponens

Règles du modus-ponens

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{C}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp)$$

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{C}\ \{Q\}}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp\text{-}pre)$$

$$\frac{\{P\}\ \mathtt{C}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp\text{-}post)$$

Remarque

En pratique, on utilisera presque exclusivement la règle (*mp-pre*)

Modus ponens

Règles du modus-ponens

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{c}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{c}\ \{Q\}}\ (mp)$$

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{c}\ \{Q\}}{\{P\}\ \mathtt{c}\ \{Q\}}\ (mp\text{-}pre)$$

$$\frac{\{P\}\ \mathtt{c}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{c}\ \{Q\}}\ (mp\text{-}post)$$

Exercice 5

Prouver

$$\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}$$

Modus ponens

Exercice 5 (*mp-pre*)

Prouver

$$\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}$$

$$\dfrac{x = 3 \implies x \geq 0 \quad \dfrac{}{\{x \geq 0\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}}\,(\mathit{aff})}{\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}}\,(\mathit{mp\text{-}pre})$$

Exercice 5 (*mp-post*)

Prouver

$$\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}$$

$$\dfrac{\dfrac{}{\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{x = 3 \wedge y = 4\}}\,(\mathit{aff}) \quad x = 3 \wedge y = 4 \implies y > 0}{\{x = 3\}\mathtt{y} = \mathtt{x} + 1\{y > 0\}}\,(\mathit{mp\text{-}post})$$

Règle des alternatives

Règle des alternatives

$$\frac{\{P_1\}\ \mathtt{C}_1\ \{Q\} \quad \{P_2\}\ \mathtt{C}_2\ \{Q\}}{\{(B \implies P_1) \wedge (\neg B \implies P_2)\}\ \mathtt{if(B)\ C_1\ else\ C_2}\ \{Q\}}\ (alt)$$

Technique de preuve

1. Chercher P_1 telle que $\{P_1\}\mathtt{C}_1\{Q\}$
2. Chercher P_2 telle que $\{P_2\}\mathtt{C}_2\{Q\}$
3. La précondition recherchée est $P \stackrel{def}{=} (B \implies P_1) \wedge (\neg B \implies P_2)$

Règle des alternatives alternative

$$\frac{\{B \wedge P\}\ \mathtt{C}_1\ \{Q\} \quad \{\neg B \wedge P\}\ \mathtt{C}_2\ \{Q\}}{\{P\}\ \mathtt{if(B)\ C_1\ else\ C_2}\ \{Q\}}\ (alt)$$

Règle des alternatives

Règle des alternatives

$$\frac{\{P_1\}\ \mathtt{C}_1\ \{Q\} \quad \{P_2\}\ \mathtt{C}_2\ \{Q\}}{\{(B \implies P_1) \wedge (\neg B \implies P_2)\}\ \mathtt{if(B)\ C_1\ else\ C_2}\ \{Q\}}\ (alt)$$

Exercice 6

Trouver P telle que

$$\{P\}\ \mathtt{if(x<y)\ x=y\ else\ x=2}\ \{x = 2\}$$

$$\frac{\{y = 2\}\ \mathtt{x=y}\ \{x = 2\} \quad \{\mathtt{true}\}\ \mathtt{x=2}\ \{x = 2\}}{\{(x < y \implies y = 2) \wedge (x \geq y \implies \mathtt{true})\}\ \mathtt{if(x<y)\ x=y\ else\ x=2}\ \{x = 2\}}\ (alt)$$

$$\frac{\{y = 2\}\ \mathtt{x=y}\ \{x = 2\} \quad \{\mathtt{true}\}\ \mathtt{x=2}\ \{x = 2\}}{\{(x < y \implies y = 2)\}\ \mathtt{if(x<y)\ x=y\ else\ x=2}\ \{x = 2\}}\ (alt)$$

$$\frac{\{y = 2\}\ \mathtt{x=y}\ \{x = 2\} \quad \{\mathtt{true}\}\ \mathtt{x=2}\ \{x = 2\}}{\{(x \geq y \vee y = 2)\}\ \mathtt{if(x<y)\ x=y\ else\ x=2}\ \{x = 2\}}\ (alt)$$

Règle des alternatives

Règle des alternatives

$$\frac{\{P_1\}\ \mathtt{C}_1\ \{Q\} \quad \{P_2\}\ \mathtt{C}_2\ \{Q\}}{\{(B \implies P_1) \wedge (\neg B \implies P_2)\}\ \mathtt{if(B)\ C_1\ else\ C_2}\ \{Q\}}\ (alt)$$

Exercice 7

Prouver

$$\{\mathtt{true}\}\ \mathtt{a=x+1;\ if((a-1)==0)\ y=1\ else\ y=a}\ \{y = x + 1\}$$

$$\dfrac{\dfrac{}{\{P\}\ \mathtt{a=x+1}\ \{P_3\}}\ (aff) \quad \dfrac{\{P_1\}\ \mathtt{y=1}\ \{y = x + 1\} \quad \{P_2\}\ \mathtt{y=a}\ \{y = x + 1\}}{\{P_3 \equiv ((a = 1 \implies P_1) \wedge (a \neq 1 \implies P_2))\}\ \mathtt{if((a-1)==0)\ y=1\ else\ y=a}\ \{y = x + 1\}}\ (alt)}{\{P\}\ \mathtt{a=x+1;\ if((a-1)==0)\ y=1\ else\ y=a}\ \{y = x + 1\}}\ (seq)$$

$P_2 \stackrel{def}{=} a = x + 1$

$P_1 \stackrel{def}{=} x = 0$

$P_3 \stackrel{def}{=} (a = 1 \implies x = 0) \wedge (a \neq 1 \implies a = x + 1)$

$P \stackrel{def}{=} (x + 1 = 1 \implies x = 0) \wedge (x + 1 \neq 1 \implies x + 1 = x + 1)$

$\iff (x = 0 \implies x = 0) \wedge (x \neq 0 \implies x = x) \iff \mathtt{true}$

Exercices

Exercice 8

Chercher la plus faible précondition P qui satisfasse

1. $\{P\}$ `i = i + 1` $\{i > 0\}$
2. $\{P\}$ `z = x` $\{z \geq x \wedge z \geq y\}$
3. $\{P\}$ `i = (lo + hi)/2` $\{lo \leq i \wedge i \leq hi\}$
4. $\{P\}$ `x = x + 1 ;y = y + 1` $\{x = y\}$
5. $\{P\}$ `x = 2*x + 1; y = y - 1` $\{y = 3x + 1\}$
6. $\{P\}$ `if (x > 0) z = x else z = -x` $\{z = |x|\}$
7. $\{P\}$ `x = 4; if (x > y) z = x else z = y` $\{z = 3\}$

Exercices

Exercice 9

Prouver

1. $\{\texttt{true}\}\ \texttt{y = x; y = x + x + y}\ \{y = 3x\}$
2. $\{xy + pq = n\}\ \texttt{x = x - p; q = q + y}\ \{xy + pq = n\}$
3. $\{x > 2\}\ \texttt{a = 1; y = x; y = y - a}\ \{y > 0 \wedge x > y\}$
4. $\{i \neq 0\}\ \texttt{if (i == 0) j = 0 else j = 1}\ \{j = 1\}$
5. $\{x \geq 0 \wedge y \geq 0\}\ \texttt{if(x>y) x = x-y else y = y-x}\ \{x \geq 0 \wedge y \geq 0\}$

Exercices

Exercice 10

1. Trouver une règle d'inférence et une technique pour trouver la plus faible précondition d'une alternative à une seule branche (sans branche `else`)
2. Montrer $\{x = 2\}$ `if (x > 1) x = x + 1` $\{x = 3\}$

Blocs lexicaux

Syntaxe

{var v_1;...; var v_n; <corps>}

Règle des blocs lexicaux

$$\frac{\{P\}\ \mathtt{C}\ \{Q\} \quad v_1, \ldots, v_n \notin P \cup Q}{\{P\}\ \{\mathtt{var}\ v_1;\ldots;\ \mathtt{var}\ v_n;\ \mathtt{C}\}\ \{Q\}}\ (let)$$

Attention

Renommage nécessaire des variables en collision

```
{var x; x = 1; {var x; x = 2} z = x}
⇝{var x; x = 1; {var y; y = 2} z = x}
```

Blocs lexicaux

Règle des blocs lexicaux

$$\frac{\{P\}\ \mathtt{C}\ \{Q\} \quad v_1, \ldots, v_n \notin P \cup Q}{\{P\}\ \{\mathtt{var\ v_1;\ldots;\ var\ v_n;\ C}\}\ \{Q\}}\ (let)$$

Exercice 11

Montrer

$$\{x = a \wedge y = b\}\ \{\mathtt{var\ r;\ r = x;\ x = y;\ y = r}\}\ \{x = b \wedge y = a\}$$

$$\dfrac{\dfrac{\{P\}\ \mathtt{-;\ -;\ -}\ \{P3\} \quad r \notin P \cup P3}{\{P\}\ \{\mathtt{var\ r;\ -;\ -;\ -}\}\ \{P3\}}\ (let) \quad \dfrac{}{\{P3\}\ \mathtt{r = x}\ \{P2\}}\ (aff) \quad \dfrac{}{\{P2\}\ \mathtt{x = y}\ \{P1\}}\ (aff) \quad \dfrac{}{\{P1\}\ \mathtt{y = r}\ \{x = b \wedge y = a\}}\ (aff)}{\{P\}\ \{\mathtt{var\ r;\ r = x;\ x = y;\ y = r}\}\ \{x = b \wedge y = a\}}\ (seq)$$

$P1 \stackrel{def}{=} x = b \wedge r = a$

$P2 \stackrel{def}{=} y = b \wedge r = a$

$P3 \stackrel{def}{=} y = b \wedge x = a$

$P \stackrel{def}{=} y = b \wedge x = a$

Affectation dans un tableau

Syntaxe

$\texttt{tab}[e_1] = e_2$

Axiome d'affectation dans un tableau

$$\frac{}{\{Q[tab(e_1 \leftarrow e_2)/tab]\}\ \texttt{tab}[e_1] = e_2\ \{Q\}}\ (tab)$$

avec pour une propriété P dépendant de $tab(e_1 \leftarrow e_2)[e_3]$:

$P\{tab(e_1 \leftarrow e_2)[e_3]\}) \stackrel{def}{=} (e_1 = e_3 \implies P\{e_2\}) \wedge (e_1 \neq e_3 \implies P\{tab[e3]\})$

Simplifications directes

- si on sait que $e_1 = e_3$ alors $tab(e_1 \leftarrow e_2)[e_3]$ se simplifie en e_2
- si on sait que $e_1 \neq e_3$ alors $tab(e_1 \leftarrow e_2)[e_3]$ se simplifie en $tab[e_3]$

Affectation dans un tableau

Axiome d'affectation dans un tableau

$$\frac{}{\{Q[tab(e_1 \leftarrow e_2)/tab]\}\ \texttt{tab[}e_1\texttt{]}\ =e_2\ \{Q\}}\ (tab)$$

Simplifications directes

- si on sait que $e_1 = e_3$ alors $tab(e_1 \leftarrow e_2)[e_3]$ se simplifie en e_2
- si on sait que $e_1 \neq e_3$ alors $tab(e_1 \leftarrow e_2)[e_3]$ se simplifie en $tab[e_3]$

Exercice 12

chercher P la plus faible précondition telle que

$$\{P\}\ \texttt{\{A[x]=1\}}\ \{A[y] = 0 \wedge A[x] = 1\}$$

$$\frac{}{\{A(x \leftarrow 1)[y] = 0 \wedge A(x \leftarrow 1)[x] = 1\}\ \texttt{\{A[x]=1\}}\ \{A[y] = 0 \wedge A[x] = 1\}}\ (tab)$$

$$\frac{}{\{A(x \leftarrow 1)[y] = 0 \wedge 1 = 1\}\ \texttt{\{A[x]=1\}}\ \{A[y] = 0 \wedge A[x] = 1\}}\ (tab)$$

$$\frac{}{\{(x = y \implies 1 = 0) \wedge (x \neq y \implies A[y] = 0)\}\ \texttt{\{A[x]=1\}}\ \{A[y] = 0 \wedge A[x] = 1\}}\ (tab)$$

$$\frac{}{\{x \neq y \wedge A[y] = 0\}\ \texttt{\{A[x]=1\}}\ \{A[y] = 0 \wedge A[x] = 1\}}\ (tab)$$

Boucles

- Préconditions et postconditions de méthodes
- Boucles

  Invariant expression logique vraie avant et après chaque tour de boucle

  Variant expression décroissante strictement entre deux tours de boucles sur un ordre bien fondé

Exemple

```
i = 0;
max = MIN_INT;
while (i < tab.length) {
  if (max <= tab[i]) {
    max = tab[i];
  }
  i = i + 1;
}
```

Précondition `true`

Postcondition la variable `max` contient le plus grand élément de tab

Invariant caractériser de façon cohérente et complète la sémantique de la boucle $\Rightarrow$ invariant "utile"

`true` "marche" mais n'est pas utile

Invariants de boucle : techniques

Question comment trouver un **bon** invariant de boucle ?

Réponse réfléchir, imaginer, ruminer, cogiter, méditer ...

Techniques

- illumination
- simulation
  - Sélectionner les variables "importantes"
  - Simuler le fonctionnement de la boucle
  - Étudier l'évolution des valeurs de variables
  - Déduire un invariant
- analyse symbolique des préconditions/postconditions
- utilisation d'un outil (Daikon, etc.)

Invariant de boucle : simulation

```
i = 0;
max = MIN_INT;
while (i < tab.length) {
  if (max <= tab[i]) {
    max = tab[i];
  }
  i = i + 1;
}
```

tour	i	max
1	0	tab[0]
2	1	Max(tab[0],tab[1])
3	2	Max(tab[0],tab[1],tab[2])
4	3	Max(tab[0]...tab[3])
		etc.

Invariant candidat max = Max(tab[0]...tab[i])

Variants de boucle : techniques

Question comment trouver un **bon** variant de boucle ?

Réponse réfléchir, imaginer, ruminer, cogiter, méditer (mais moins)

```
i = 0;
max = MIN_INT;
while (i < tab.length) {
  if (max <= tab[i]) {
    max = tab[i];
  }
  i = i + 1;
}
```

Candidat `-i` ou “mieux” `tab.length - i`

Boucles while

Règle de la boucle while

$$\frac{\{P \wedge S\}\ \texttt{C}\ \{P\}}{\{P\}\ \texttt{while (S) C}\ \{Q\}}\ (\mathit{while})$$

Deux interprétations possibles

Correction partielle si la boucle se termine et l'hypothèse alors la conclusion

Correction totale on prouve que la boucle se termine et si l'hypothèse alors la conclusion

Boucles while : autre version

Règle de la boucle while

$$\frac{P \wedge S \implies P' \quad \{P'\}\ \mathtt{C}\ \{P\} \quad P \wedge \neg S \implies Q}{\{P\}\ \mathtt{while\ (S)\ C}\ \{Q\}}\ (\mathit{while})$$

Technique (correction partielle)

1. trouver l'invariant de boucle P

   en capturant des informations liées au variant (critère de terminaison)
2. prouver $P \wedge \neg S \implies Q$
3. déduire P' la plus faible précondition telle que $\{P'\}\ \mathtt{C}\ \{P\}$
4. prouver $P \wedge S \implies P'$

   $\Rightarrow P$ est la précondition recherchée

Correction partielle de boucle

Exercice 13

Montrer

$\{x \geq 0\}$

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y*z;
}
```

$\{y = x!\}$

Exercices

Exercice 14

La spécification suivante est-elle correcte ? Faire la démonstration

$\{tab[i] = X \wedge tab[j] = Y\}$

```
{ var r;
  r = tab[i];
  tab[i] = tab[j];
  tab[j] = r;
}
```

$\{tab[i] = Y \wedge tab[j] = X\}$

Exercices

Exercice 15

Soit le programme `Prog1` suivant

```
var a;
a = x;
y = 0;
while (a != 0) {
   y = y + 1;
   a = a - 1;
}
```

1. Trouver un variant et un invariant de boucle
2. Donner un argument concernant la terminaison du programme
3. Prouver $\{x \geq 0\}$ `Prog1` $\{x = y\}$

Exercices

Exercice 16

Prouver la spécification suivante

$\{n \geq 0\}$

```
{ var m; var y;
  m = n; p = 1; y = x;
  if (x != 0)
    while (m != 0) {
      if (odd(m)) p = p*y;
      else p = p;
      m = m / 2;
      y = y * y;
    }
  else p = 0
}
```

$\{p = xn\}$

Axiome d'affecation

$$\frac{}{\{Q[\mathrm{expr}/V]\}\ \mathtt{V} = \mathtt{expr}\ \{Q\}}\ (aff)$$

Règle de séquencement

$$\frac{\{P\}\ \mathtt{C}_1\ \{Q_1\} \quad \{Q_1\}\ \mathtt{C}_2\ \{Q_2\} \quad \ldots \quad \{Q_{n-1}\}\ \mathtt{C}_n\ \{Q\}}{\{P\}\ \mathtt{C}_1; \ldots; \mathtt{C}_n\ \{Q\}}\ (seq)$$

Règles du modus-ponens

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{C}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp)$$

$$\frac{P \implies P' \quad \{P'\}\ \mathtt{C}\ \{Q\}}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp\text{-}pre)$$

$$\frac{\{P\}\ \mathtt{C}\ \{Q'\} \quad Q' \implies Q}{\{P\}\ \mathtt{C}\ \{Q\}}\ (mp\text{-}post)$$

Règle des alternatives

$$\frac{\{P_1\}\ \mathtt{C}_1\ \{Q\} \quad \{P_2\}\ \mathtt{C}_2\ \{Q\}}{\{(B \implies P_1) \wedge (\neg B \implies P_2)\}\ \mathtt{if(B)\ C_1\ else\ C_2}\ \{Q\}}\ (alt)$$

Règle des blocs lexicaux

$$\frac{\{P\}\ \mathtt{C}\ \{Q\} \quad v_1, \ldots, v_n \notin P \cup Q}{\{P\}\ \mathtt{\{var\ v_1;\ldots;\ var\ v_n;\ C\}}\ \{Q\}}\ (let)$$

Axiome d'affectation dans un tableau

$$\frac{}{\{Q[tab(e_1 \leftarrow e_2)/tab]\}\ \mathtt{tab[e_1]=e_2}\ \{Q\}}\ (tab)$$

Règle de la boucle while

$$\frac{P \wedge S \implies P' \quad \{P'\}\ \mathtt{C}\ \{P\} \quad P \wedge \neg S \implies Q}{\{P\}\ \mathtt{while\ (S)\ C}\ \{Q\}}\ (while)$$