Preuves de programmes impératifs

Spécification par les triplets de Hoare

- Un programme impératif modifie l'état de la mémoire à chaque étape de son exécution
  - Semblable à une fonction mathématique dont le paramètre est l'état initial de la mémoire et le résultat est l'état final de la mémoire
  - Chaque étape de calcul est une fonction de l'état précédent de la mémoire vers l'état suivant de la mémoire
- Spécification d'un programme impératif
  - Post-condition ψ : Propriétés attendues sur l'état de la mémoire à la fin de l'exécution
  - Pré-condition φ : En fonction des propriétés de l'état de la mémoire au début de l'exécution
  - Notation : Triplet de Hoare $\{\varphi\}\, P\, \{\psi\}$

Exemple de spécification formelle

Élévation au carré efficace

Syntaxe similaire à Ada mais plus compacte.

$$
\begin{array}{l}
\{0 \leq N\} \\
x := 0; \\
y := 0; \\
\mathit{while}\ x \neq N\ \mathit{do} \\
\qquad y := y + 2 * x + 1; \\
\qquad x := x + 1 \\
\mathit{od} \\
\{y = N^2\}
\end{array}
$$

Preuves de programmes impératifs

Preuve de correction

- Chaque étape intermédiaire est annotée par une propriété de l'état de la mémoire
- Chaque instruction *I* est :
  - précédée d'une pré-condition φ (propriété de l'état de la mémoire avant l'exécution de l'instruction)
  - suivie d'une post-condition ψ (propriété de l'état de la mémoire après l'exécution de l'instruction)
- Chaque instruction annotée doit satisfaire les règles de la logique de Hoare $\{\varphi\}$ *I* $\{\psi\}$
  - Correction partielle : Si la pré-condition φ est satisfaite avant l'exécution de l'instruction *I* et l'exécution se termine alors la post-condition ψ est satisfaite après l'exécution
  - Correction totale : Si la pré-condition φ est satisfaite avant l'exécution de l'instruction *I* alors l'exécution se termine et la post-condition ψ est satisfaite après l'exécution
- Les propriétés de l'état de la mémoire sont représentées par de la logique équationnelle (logique du premier ordre et spécifications algébriques)

Exemple de preuve de correction partielle

Élévation au carré efficace

Utilisation d'un invariant $y = x^2$

$$
\begin{array}{l}
\{0 \leq N\} \\
\{0 = 0^2\} \\
x := 0; \\
\{0 = x^2\} \\
y := 0; \\
\{y = x^2\} \\
\textit{while } x \neq N \textit{ invariant } y = x^2 \textit{ do} \\
\qquad \{y = x^2 \wedge x \neq N\} \\
\qquad \{y + 2 \times x + 1 = (x+1)^2\} \\
\qquad y := y + 2 * x + 1; \\
\qquad \{y = (x+1)^2\} \\
\qquad x := x + 1 \\
\qquad \{y = x^2\} \\
\textit{od} \\
\{y = x^2 \wedge \neg(x \neq N)\} \\
\{y = N^2\}
\end{array}
$$

Logique de Floyd/Hoare

Règles de déduction

$$\frac{}{\{\psi\}\,\mathsf{skip}\,\{\psi\}}\ \text{skip} \qquad \frac{}{\{[E/x]\,\psi\}\,x := E\,\{\psi\}}\ \text{assign}$$

$$\frac{\{\varphi\}\,P\,\{\chi\} \qquad \{\chi\}\,Q\,\{\psi\}}{\{\varphi\}\,P\ ;\ Q\,\{\psi\}}\ \text{sequence}$$

$$\frac{\{\varphi \wedge C\}\,P\,\{\psi\} \qquad \{\varphi \wedge \neg C\}\,Q\,\{\psi\}}{\{\varphi\}\,\mathsf{if}\ C\ \mathsf{then}\ P\ \mathsf{else}\ Q\ \mathsf{fi}\,\{\psi\}}\ \text{conditional}$$

$$\frac{\{\varphi \wedge C\}\,P\,\{\varphi\}}{\{\varphi\}\,\mathsf{while}\ C\ \mathsf{invariant}\ \varphi\ \mathsf{do}\ P\ \mathsf{od}\,\{\varphi \wedge \neg C\}}\ \text{partial loop}$$

$$\frac{\{\varphi \wedge C \wedge E \in \mathbb{N} \wedge V = E\}\,P\,\{\varphi \wedge E \in \mathbb{N} \wedge V > E\}}{\{\varphi \wedge E \in \mathbb{N}\}\,\mathsf{while}\ C\ \mathsf{invariant}\ \varphi\ \mathsf{variant}\ E\ \mathsf{do}\ P\ \mathsf{od}\,\{\varphi \wedge \neg C\}}\ \text{total loop}$$

$$\frac{\varphi \to \chi \qquad \{\chi\}\,P\,\{\psi\}}{\{\varphi\}\,P\,\{\psi\}}\ \text{weaken} \qquad \frac{\{\varphi\}\,P\,\{\chi\} \qquad \chi \to \psi}{\{\varphi\}\,P\,\{\psi\}}\ \text{strengthen}$$

Exemple de preuve de correction totale

Élévation au carré efficace

Utilisation d'un variant : $N - x$

$$
\begin{array}{l}
\{0 \leq N\} \\
\{\cdots \wedge (N-0) \in \mathbb{N}\} \\
x := 0; \\
\{\cdots \wedge (N-x) \in \mathbb{N}\} \\
y := 0; \\
\{\cdots \wedge (N-x) \in \mathbb{N}\} \\
\textit{while } x \neq N \textit{ invariant } y = x^2 \textit{ variant } N - x \textit{ do} \\
\qquad \{\cdots \wedge x \neq N \wedge (N-x) \in \mathbb{N} \wedge V = N - x\} \\
\qquad \{\cdots \wedge (N-(x+1)) \in \mathbb{N} \wedge N-(x+1) < V\} \\
\qquad y := y + 2 * x + 1; \\
\qquad \{\cdots \wedge (N-(x+1)) \in \mathbb{N} \wedge N-(x+1) < V\} \\
\qquad x := x + 1 \\
\qquad \{\cdots \wedge (N-x) \in \mathbb{N} \wedge N - x < V\} \\
\textit{od} \\
\{\cdots\} \\
\{y = N^2\}
\end{array}
$$

Logique de Floyd/Hoare

Règles de déduction

- Inaction : `skip`
  - Le contenu de la mémoire n'est pas modifié durant l'exécution
  - La pré-condition et la post-condition sont identiques

$$\frac{}{\{\psi\}\,\textsf{skip}\,\{\psi\}}\ \textsf{skip}$$

- Renforcement de la pré-condition (hypothèse plus forte)

$$\frac{\varphi \to \chi \qquad \{\chi\}\,P\,\{\psi\}}{\{\varphi\}\,P\,\{\psi\}}\ \textsf{weaken}$$

- Affaiblissement de la post-condition (conclusion plus faible)

$$\frac{\{\varphi\}\,P\,\{\chi\} \qquad \chi \to \psi}{\{\varphi\}\,P\,\{\psi\}}\ \textsf{strengthen}$$

Logique de Floyd/Hoare

Affectation et Plus faible pré-condition

Affectation : `x := E`

- Seul le contenu de `x` est modifié
  - Mémoire avant : $\langle x_1 \mapsto v_1, \ldots, x_n \mapsto v_n \rangle$
  - Mémoire après : $\langle x_1 \mapsto v'_1, \ldots, x_n \mapsto v'_n \rangle$
  - Si $x = x_i$ alors :
    - $v'_i = [v_1/x_1, \ldots, v_n/x_n]\, E$
    - et $\forall j \in [1, \ldots, n].\, ((j \neq i) \rightarrow v'_j = v_j)$
  - Donc : $[v_1/x_1, \ldots, v_n/x_n]\, [E/x]\, \psi = [v'_1/x_1, \ldots, v'_n/x_n]\, \psi$

$$\frac{}{\{[E/x]\, \psi\}\; x := E\; \{\psi\}}\ \text{assign}$$

- Il s'agit de calculer la plus faible pré-condition nécessaire pour établir la post-condition ψ
- Travaux de Dijkstra
- Base des outils de mécanisation comme Why3

Logique de Floyd/Hoare

Séquence et Conditionnelle

- Séquence : `P ; Q`
  - L'état de la mémoire entre l'exécution de `P` et de `Q` est caractérisé par la propriété χ

    $$\frac{\{\varphi\}\, P\, \{\chi\} \qquad \{\chi\}\, Q\, \{\psi\}}{\{\varphi\}\, P\, ;\ Q\, \{\psi\}}\ \text{sequence}$$

  - Calcul de la plus faible pré-condition

- Conditionnelle : `if C then P else Q fi`
  - L'état de la mémoire après l'exécution des branches `P` et de `Q` doit satisfaire la propriété ψ
  - L'état de la mémoire avant l'exécution des branches `P` et de `Q` est distingué par la valeur de la condition `C`

    $$\frac{\{\varphi \wedge C\}\, P\, \{\psi\} \qquad \{\varphi \wedge \neg C\}\, Q\, \{\psi\}}{\{\varphi\}\ \text{if } C \text{ then } P \text{ else } Q \text{ fi}\ \{\psi\}}\ \text{conditional}$$

  - Calcul de la plus faible pré-condition

Logique de Floyd/Hoare

Boucle, Correction partielle

- Boucle : `while C invariant` φ `variant E do P od`
- Dépliage boucle :
  - Notons $R = \mathit{while}\ C\ \mathit{invariant}\ \varphi\ \mathit{do}\ P\ \mathit{od}$
  - Alors R satisfait $R = \mathit{if}\ C\ \mathit{then}\ P\ ;\ \ R\ \mathit{else\ skip\ fi}$
  - Plus faible pré-condition est une disjonction infinie des séquences $P^{(i)}$ issues des dépliages de R
  - Besoin d'une propriété finie plus forte : l'invariant de boucle φ
- Correction partielle :

$$\frac{\{\varphi \wedge C\}\ P\ \{\varphi\}}{\{\varphi\}\ \text{while } C \text{ invariant } \varphi \text{ do } P \text{ od}\ \{\varphi \wedge \neg C\}}\ \text{partial loop}$$

- Principe construction :

$$\dfrac{\dfrac{\dfrac{\{\varphi \wedge C\}\ P\ \{\varphi\} \qquad \{\varphi\}\ R\ \{\varphi \wedge \neg C\}}{\{\varphi \wedge C\}\ P\ ;\ R\ \{\varphi \wedge \neg C\}}\ \text{sequence} \qquad \dfrac{}{\{\varphi \wedge \neg C\}\ \text{skip}\ \{\varphi \wedge \neg C\}}\ \text{skip}}{\{\varphi\}\ \text{if } C \text{ then } P\ ;\ \ R \text{ else } \mathit{skip} \text{ fi}\ \{\varphi \wedge \neg C\}}\ \text{conditional}}{\{\varphi\}\ \text{while } C \text{ invariant } \varphi \text{ do } P \text{ od}\ \{\varphi \wedge \neg C\}}\ \text{unfold loop}$$

Logique de Floyd/Hoare

Boucle, Correction totale

- Boucle : `while C invariant` φ `variant E do P od`
- Correction totale : Combiner la correction partielle et une preuve de terminaison
- Induction bien fondée sur l'état de la mémoire
- L'état de la mémoire doit décroitre strictement à chaque exécution du corps de la boucle
  - Variant : $E \in \mathbb{N}$
  - $[v'_1/x_1, \ldots, v'_n/x_n]\, E < [v_1/x_1, \ldots, v_n/x_n]\, E$
- Correction totale :

$$\frac{\{\varphi \wedge C \wedge E \in \mathbb{N} \wedge V = E\}\, P\, \{\varphi \wedge E \in \mathbb{N} \wedge V > E\}}{\{\varphi \wedge E \in \mathbb{N}\}\, \text{while } C \text{ invariant } \varphi \text{ variant } E \text{ do } P \text{ od}\, \{\varphi \wedge \neg C\}} \text{ total loop}$$

- Principe de construction identique à la correction partielle (variable V contient valeur du variant avant exécution du corps de la boucle)
- Il peut être nécessaire d'enrichir l'invariant avec $C \to E \in \mathbb{N}$

Logique de Floyd/Hoare

Méthode de preuve

1. Recherche des candidats invariants et variants pour chaque boucle
2. Annotations des boucles
3. Remontée des plus faibles pré-conditions le long des séquences, affectations et branches de conditionnelle
4. Construction des obligations de preuve issues des affaiblissements et renforcements dans les boucles et les branches de conditionnelle
5. En cas d'échec dans la preuve des obligations, renforcement des invariants avec les propriétés manquantes pour réaliser ces preuves
6. Reprise à l'étape 2

Exemple d'application de la méthode

Élévation au carré efficace

$$\{0 \leq N\}$$
$$x := 0;$$
$$y := 0;$$
$$\textit{while } x \neq N \textit{ do}$$
$$\quad y := y + 2 * x + 1;$$
$$\quad x := x + 1$$
$$\textit{od}$$
$$\{y = N^2\}$$

Étapes de boucle	x	y	$y = x^2$	$N - x$
0	0	0	$\top$	N
1	1	1	$\top$	$N - 1$
2	2	4	$\top$	$N - 2$
3	3	9	$\top$	$N - 3$
4	4	16	$\top$	$N - 4$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

- Premier invariant candidat : $\varphi_0 : y = x^2$
- Premier variant candidat : $N - x$

Exemple d'application de la méthode

Élévation au carré efficace

$\{0 \leq N\}$
$\{0 = 0^2 \wedge (N - 0) \in \mathbb{N}\}$
$x := 0;$
$\{0 = x^2 \wedge (N - x) \in \mathbb{N}\}$
$y := 0;$
$\{y = x^2 \wedge (N - x) \in \mathbb{N}\}$
while $x \neq N$ *invariant* $y = x^2$ *variant* $N - x$ *do*
$\qquad \{y = x^2 \wedge x \neq N \wedge (N - x) \in \mathbb{N} \wedge V = N - x\}$
$\qquad \{y + 2 \times x + 1 = (x + 1)^2 \wedge (N - (x + 1)) \in \mathbb{N} \wedge N - (x + 1) < V\}$
$\qquad y := y + 2 * x + 1;$
$\qquad \{y = (x + 1)^2 \wedge (N - (x + 1)) \in \mathbb{N} \wedge N - (x + 1) < V\}$
$\qquad x := x + 1$
$\qquad \{y = x^2 \wedge (N - x) \in \mathbb{N} \wedge N - x < V\}$
od
$\{y = x^2 \wedge \neg(x \neq N)\}$
$\{y = N^2\}$

Exemple d'application de la méthode

Élévation au carré efficace

$$0 \leq N \rightarrow 0 = 0^2 \wedge (N - 0) \in \mathbb{N}$$

$$\left\{ \begin{array}{l} y = x^2 \\ \wedge\, x \neq N \\ \wedge\, (N - x) \in \mathbb{N} \\ \wedge\, V = N - x \end{array} \right. \rightarrow \left\{ \begin{array}{l} y + 2 \times x + 1 = (x+1)^2 \\ \wedge\, (N - (x+1)) \in \mathbb{N} \\ \wedge\, N - (x+1) < V \end{array} \right.$$

$$(y = x^2 \wedge \neg(x \neq N)) \rightarrow y = N^2$$

Logique de Floyd/Hoare
Bilan

- La logique de Floyd/Hoare est complète, consistante et correcte vis à vis de la sémantique des programmes
- Le calcul des plus faibles pré-conditions est complet, consistant et décidable
- Les caractéristiques de la logique de Hoare dépendent donc des caractéristiques de la logique utilisée pour exprimer les obligations de preuve
- Il existe de nombreux outils qui intègrent la logique de Floyd/Hoare dans les langages de programmation
  - SPARK Ada par AdaCore et Altran Praxis
  - CAVEAT et Frama-C par CEA
  - Why3 par LRI et INRIA
  - Boogie par MicroSoft Research
  - Spec# par MicroSoft Research
  - $F^\star$ par MicroSoft Research