Cours 9: preuve de programmes impératifs

David Delahaye

Polytech' Montpellier
David.Delahaye@lirmm.fr

AIGLE M1

Preuve de programmes impératifs

Principe

- Programmes fonctionnels : spécification sur un résultat ;
- Programmes impératifs :
  - Effets de bords sur un environnement (ensemble de variables) ;
  - Spécification sur l'évolution de l'environnement (les états) ;
  - Logique particulière : logique de Hoare.

Outils

- Beaucoup moins que pour le fonctionnel ;
- Deux outils assez connus :
  - Atelier B (industriel) : http://www.atelierb.eu/ ;
  - Why (académique) : http://why3.lri.fr/.

Le langage

Petit noyau impératif

- Expressions entières et booléennes ;
- Instructions d'affectation, de conditionnelle, et de boucle.

Expressions et instructions

- $e \ ::= \ n \mid x \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 \times e_2 \mid e_1 / e_2$
  $\mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{not}(e) \mid e \ \mathrm{and} \ e \mid e \ \mathrm{or} \ e$
  $\mid e = e \mid e \mathrel{!=} e \mid e < e \mid e \leq e \mid e \geq e \mid e > e$
  où $n \in \mathbb{Z}$ et $x \in \mathbb{V}$ (ensemble de noms de variables) ;
- $i \ ::= \ \mathrm{skip} \mid x := e \mid i; i \mid \mathrm{if} \ e \ \mathrm{then} \ i \ \mathrm{else} \ i \mid \mathrm{while} \ e \ \mathrm{do} \ i.$

Sémantique opérationnelle à grands pas

Sémantique des expressions

- Valeurs : $v_e \ ::= \ n \mid b \mid \mathrm{Err}$, où $n \in \mathbb{Z}$, $b \in \mathbb{B} = \{\top, \bot\}$;
- Contextes d'exécution : $E = (x_1, v_1), (x_2, v_2), \ldots, (x_n, v_n)$;
- Sémantique : relation « $E \vdash e \rightsquigarrow v_e$ » ;
- Règles :

$$\frac{n \in \mathbb{Z}}{E \vdash n \rightsquigarrow n}\ \mathbb{Z} \qquad \frac{(x, v) \in E}{E \vdash x \rightsquigarrow v}\ \mathbb{V}$$

$$\frac{E \vdash e_1 \rightsquigarrow v_1 \qquad E \vdash e_2 \rightsquigarrow v_2}{E \vdash e_1 \ \mathrm{op}\ e_2 \rightsquigarrow v_1 \ \mathrm{op}_{\mathbb{Z}}\ v_2}\ \mathrm{op},\ \text{avec } \mathrm{op} \in \{+, -, \times, /\}$$

$$\frac{}{E \vdash \mathrm{true} \rightsquigarrow \top}\ \text{true} \qquad \frac{}{E \vdash \mathrm{false} \rightsquigarrow \bot}\ \text{false}$$

$$\frac{E \vdash e \rightsquigarrow b}{E \vdash \mathrm{not}(e) \rightsquigarrow \neg b}\ \mathrm{not}$$

Sémantique opérationnelle à grands pas

Sémantique des expressions

- Règles :

$$\frac{E \vdash e_1 \rightsquigarrow b_1 \qquad E \vdash e_2 \rightsquigarrow b_2}{E \vdash e_1 \text{ and } e_2 \rightsquigarrow b_1 \wedge b2} \text{ and}$$

$$\frac{E \vdash e_1 \rightsquigarrow b_1 \qquad E \vdash e_2 \rightsquigarrow b_2}{E \vdash e_1 \text{ or } e_2 \rightsquigarrow b_1 \vee b2} \text{ or}$$

$$\frac{E \vdash e_1 \rightsquigarrow v_1 \qquad E \vdash e_2 \rightsquigarrow v_2}{E \vdash e_1 \text{ op } e_2 \rightsquigarrow v_1 \text{ op}_{\mathbb{Z},\mathbb{B}} \; v_2} \text{ op}, \text{ avec op} \in \{=, !=, <, \leq, \geq, >\}$$

Sémantique opérationnelle à grands pas

Sémantique des instructions

- Valeurs : $v_i \ ::= \ E \mid \mathrm{Err}$;
- Sémantique : relation « $E \vdash e \rightsquigarrow v_i$ » ;
- Règles :

$$\frac{x \in \mathrm{dom}(E) \qquad E \vdash e \rightsquigarrow v}{E \vdash x := e \rightsquigarrow E \leftarrow (x, v)} :=$$

$$\frac{E \vdash i_1 \rightsquigarrow E_1 \qquad E_1 \vdash i_2 \rightsquigarrow E_2}{E \vdash i_1 ; i_2 \rightsquigarrow E_2} ;$$

$$\frac{E \vdash e \rightsquigarrow \top \qquad E \vdash i_1 \rightsquigarrow E'}{E \vdash \mathrm{if}\ e\ \mathrm{then}\ i_1\ \mathrm{else}\ i_2 \rightsquigarrow E'}\ \mathrm{if}_\top$$

$$\frac{E \vdash e \rightsquigarrow \bot \qquad E \vdash i_2 \rightsquigarrow E'}{E \vdash \mathrm{if}\ e\ \mathrm{then}\ i_1\ \mathrm{else}\ i_2 \rightsquigarrow E'}\ \mathrm{if}_\bot$$

Sémantique opérationnelle à grands pas

Sémantique des instructions

- Règles :

$$\frac{E \vdash e \rightsquigarrow \top \qquad E \vdash i \rightsquigarrow E' \qquad E' \vdash \text{while } e \text{ do } i \rightsquigarrow E''}{E \vdash \text{while } e \text{ do } i \rightsquigarrow E''}\ \text{while}_\top$$

$$\frac{E \vdash e \rightsquigarrow \bot}{E \vdash \text{while } e \text{ do } i \rightsquigarrow E}\ \text{while}_\bot$$

Logique de Hoare

Triplet de Hoare

- Triplet noté : $\{P\}\ i\ \{Q\}$, où P et Q sont des assertions logiques, et i une instruction ;
- Assertions logiques : exprimées en logique du premier ordre, où les atomes sont les expressions de notre langage ;
- Un triplet de Hoare $\{P\}\ i\ \{Q\}$ est valide si pour tous états E_1 et E_2 tels que si P est vraie dans E_1 et $E_1 \vdash i \rightsquigarrow E_2$ (i termine), alors Q est vraie dans E_2.

Exemples de triplets de Hoare valides

- $\{x = 1\}\ x := x + 2\ \{x = 3\}$;
- $\{x = y\}\ x := x + y\ \{x = 2 \times y\}$.

Logique de Hoare

Règles

$$\frac{}{\{P\}\ \mathrm{skip}\ \{P\}}\ \textsf{skip} \qquad \frac{}{\{P(e)\}\ x := e\ \{P(x)\}}\ :=$$

$$\frac{\{P\}\ i_1\ \{Q\} \qquad \{Q\}\ i_2\ \{R\}}{\{P\}\ i_1; i_2\ \{R\}}\ ;$$

$$\frac{\{P \wedge e\}\ i_1\ \{Q\} \qquad \{P \wedge \neg e\}\ i_2\ \{Q\}}{\{P\}\ \mathrm{if}\ e\ \mathrm{then}\ i_1\ \mathrm{else}\ i_2\ \{Q\}}\ \textsf{if}$$

$$\frac{\{I \wedge e\}\ i\ \{I\}}{\{I\}\ \mathrm{while}\ e\ \mathrm{do}\ i\ \{I \wedge \neg e\}}\ \textsf{while}$$

$$\frac{\{P'\}\ i\ \{Q'\} \qquad P \Rightarrow P' \qquad Q' \Rightarrow Q}{\{P\}\ i\ \{Q\}}\ \textsf{Aff}$$

Logique de Hoare

Correction totale (avec terminaison)

- La sémantique précédente est partielle : elle suppose que le programme termine ;
- La sémantique peut être totale en imposant que le programme termine (par la pré-condition) ;
- Correction totale :
  Un triplet de Hoare $\{P\}\ i\ \{Q\}$ est valide si pour tous états E_1 et E_2 tels que si P est vraie dans E_1, alors $E_1 \vdash i \rightsquigarrow E_2$ (i termine), et Q est vraie dans E_2 ;
- Nouvelle règle pour le while :

$$\frac{\{I \wedge e \wedge v = n\}\ i\ \{I \wedge v \geq 0 \wedge v < n\}}{\{I\}\ \text{while}\ e\ \text{do}\ i\ \{I \wedge \neg e\}}\ \text{while}$$

  où v est le variant (expression) et n une variable entière n'apparaissant pas dans i.

Exemples

Séquence

$$\dfrac{\dfrac{}{\{0+x\geq 0\}\ a:=0\ \{a+x\geq 0\}}\ {:=} \qquad \pi}{\{0+x\geq 0\}\ a:=0;b:=x\ \{a+b\geq 0\}}\ ;$$

$$\dfrac{\begin{array}{c}\{0+x\geq 0\}\ a:=0;b:=x\ \{a+b\geq 0\}\\ x\geq 0\Rightarrow 0+x\geq 0\\ a+b\geq 0\Rightarrow a\geq -b\end{array}}{\{x\geq 0\}\ a:=0;b:=x\ \{a\geq -b\}}\ \text{Aff}$$

où π est la preuve suivante :

$$\dfrac{}{\{a+x\geq 0\}\ b:=x\ \{a+b\geq 0\}}\ {:=}$$

Exemples

Conditionnelle

$$\dfrac{\dfrac{}{\{y = 0\}\ x := y\ \{x = 0\}}\,{:=} \qquad \pi}{\{\}\ \text{if}\ y = 0\ \text{then}\ x := y\ \text{else}\ x := 0\ \{x = 0\}}\,\mathsf{if}$$

où π est la preuve suivante :

$$\dfrac{\neg(y = 0) \Rightarrow 0 = 0 \qquad \dfrac{}{\{0 = 0\}\ x := 0\ \{x = 0\}}\,{:=}}{\{\neg(y = 0)\}\ x := 0\ \{x = 0\}}\,\mathsf{Aff}$$

Exemples

Boucle while

$$\dfrac{\dfrac{x \geq 0 \wedge x < 10 \Rightarrow x+1 \geq 0 \qquad \pi}{\{x \geq 0 \wedge x < 10\}\ x := x+1\ \{x \geq 0\}}\ \mathsf{Aff}}{\{x \geq 0\}\ \text{while}\ x < 10\ \text{do}\ x := x+1\ \{x \geq 0 \wedge \neg(x < 10)\}}\ \mathsf{while}$$

où π est la preuve suivante :

$$\dfrac{}{\{x+1 \geq 0\}\ x := x+1\ \{x \geq 0\}}\ :=$$

Exercice

Démontrer la validité des triplets de Hoare suivants

- $\{x = 0\}\ x := x + 1; x := x + 1\ \{x = 2\}$;
- $\{x = 1 \wedge y = 2\}\ t := x; x := y; y := t\ \{x = 2 \wedge y = 1\}$;
- $\{x \geq 0\}$ if $x \geq 0$ then $y := 1$ else $y := 2$ $\{y = 1\}$;
- $\{x \geq 0\}$ if $x\ != 0$ then $x := x - 1$ else $x := x + 1$ $\{x \geq 0\}$;
- $\{\}$ while $x\ != 0$ do $x := x - 1$ $\{x = 0\}$.

Exercice

Démontrer que le programme suivant implante la fonction factorielle

$\{\}$
$i := 0;$
$r := 1;$
$\textit{while } i \mathrel{!=} n \text{ do}$
$\quad i := i + 1;$
$\quad r := r \times i;$
$\{r = n!\}$