Progetto 1.

Si consideri il linguaggio IMP e la sua semantica operazionale definite nelle slide che seguono.

1) Si codifichi in Coq la sintassi e la semantica di IMP.
2) Si dimostri il teorema seguente in Coq.

Proposition

$w \equiv$ while b do c

$w' \equiv$ if b then c ; w else skip $\Rightarrow w \sim w'$

dove la relazione di equivalenza ~ tra comandi è definita alla fine delle slide.

3) Sia **w** il programma **while 0< x do (y:=2*y;x:=x-1).** Si dimostri in Coq che ∀σ ∃ σ* <w, σ[2/x][3/y]> → σ* **OPPURE** dimostrare che dato lo stato σ tale che per ogni locazione i σ(i) =0, allora ∃ σ* <w, σ[2/x][3/y]> → σ*

Syntactic Sets

- number N
- truth values B
- arithmetic expressions $Aexp$
- boolean expression $Bexp$
- commands Com
- locations Loc

formation rules

- ***Aexp***

$$a ::= \mathtt{n} \mid X \mid a_0 \mathtt{+}\ a_1 \mid a_0 \mathtt{-}\ a_1 \mid a_0 \times a_1$$

- ***Bexp***

$$b ::= \mathtt{true} \mid \mathtt{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1$$

- ***Com***

$$c ::= \mathtt{skip} \mid c_1 \mathtt{\ ;\ } c_2 \mid \mathtt{if}\ b\ \mathtt{then}\ c_0\ \mathtt{else}\ c_1 \mid X \mathtt{\ :=\ } a \mid \mathtt{while}\ b\ \mathtt{do}\ c$$

meta-variables

n, m : range over $\boldsymbol{N}$

X, Y: range over ***Loc***

a: to represent an arbitrary element of ***Aexp***

b: to represent an arbitrary element of ***Bexp***

c : to represent an arbitrary element of ***Com***

Operational Semantics

- **evaluation** of arithmetic and boolean expression
- **execution** of commands
  - **semantic sets**

Z : set of "*machine number*"

Σ : set of "*states*"

$$\Sigma : \{\sigma | \sigma : \mathbf{Loc} \rightarrow \mathbf{Z}\}$$

C: set of "*configurations*"

$$\mathbf{C} = \{\langle p, \sigma \rangle | p \in \mathbf{Aexp}, \mathbf{Bexp}, \mathbf{Com}, \sigma \in \Sigma\}$$

evaluation of **Aexp** elements

evalutation relation: $\langle a, \sigma\rangle \to n \quad (n \in \mathbf{Z})$

Numbers:

$$\langle \mathtt{n}, \sigma\rangle \to n$$

locations:

$$\langle X, \sigma\rangle \to \sigma(X)$$

sums:

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \mathtt{+} a_1, \sigma\rangle \to n_0 + n_1}$$

substractions:

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \mathtt{\,-\,} a_1, \sigma\rangle \to n_0 - n_1}$$

products:

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \times a_1, \sigma\rangle \to n_0 \times n_1}$$

evaluation of **Bexp** elements

evalutation relation: $\langle b, \sigma\rangle \to t \quad t \in \{\mathbf{true}, \mathbf{false}\}$

$$\langle \mathtt{true}, \sigma\rangle \to \mathit{true}$$

$$\langle \mathtt{false}, \sigma\rangle \to \mathit{false}$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \mathtt{\,=\,} a_1, \sigma\rangle \to \mathit{true}} \text{ when } n_0 = n_1$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \mathtt{\,=\,} a_1, \sigma\rangle \to \mathit{false}} \text{ when } n_0 \neq n_1$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \leq a_1, \sigma\rangle \to \mathit{true}} \text{ when } n_0 \leq n_1$$

$$\frac{\langle a_0, \sigma\rangle \to n_0 \;\; \langle a_1, \sigma\rangle \to n_1}{\langle a_0 \leq a_1, \sigma\rangle \to \mathit{false}} \text{ when } n_0 \not\leq n_1$$

$$\frac{\langle b, \sigma\rangle \to t}{\langle \neg b, \sigma\rangle \to \neg t}$$

$$\frac{\langle b_0, \sigma\rangle \to t_0 \;\; \langle b_1, \sigma\rangle \to t_1}{\langle b_0 \wedge b_1, \sigma\rangle \to t_0 \wedge t_1}$$

$$\frac{\langle b_0, \sigma\rangle \to t_0 \;\; \langle b_1, \sigma\rangle \to t_1}{\langle b_0 \vee b_1, \sigma\rangle \to t_0 \vee t_1}$$

$b_0 \sim b_1$ iff $(\forall t \in \{\mathit{false}, \mathit{true}\}\ \forall \sigma \in \sum . \langle b_0, \sigma\rangle \to t \Leftrightarrow \langle b_1, \sigma\rangle \to t)$

execution of commands

command execution relation → $\langle c, \sigma \rangle \to \sigma'$

we assume the existence of an *initial state* σ_0 such that

$$(\forall X \in \textbf{\textit{Loc}}.\ \sigma_0(X) = 0)$$

$\langle c, \sigma \rangle \to \sigma'$ ⇒ the execution of c in σ terminates in σ′

Notation. Let σ be a state. Let $m \in \boldsymbol{Z}$. Let $X \in \mathit{Loc}$. We write $\sigma[m/X]$ for "the state obtained from σ by replacing the contents of X by m", i.e.

$$\sigma[m/X](Y) = \begin{cases} m & \text{if } Y = X \\ \sigma(Y) & \text{otherwise} \end{cases}$$

Example: consider σ such that $\sigma(X) = 2$, $\sigma(Y) = 4$. Let σ' be $\sigma[5/X]$. We have that $\sigma'(X) = 5$, $\sigma'(Y) = 4$

Atomic commands:

$$\langle \texttt{skip},\ \sigma \rangle \to \sigma$$

$$\frac{\langle a,\ \sigma \rangle \to m}{\langle X\ \texttt{:=}\ a,\ \sigma \rangle \to \sigma[m/X]}$$

Sequencing:

$$\frac{\langle c_0, \sigma \rangle \to \sigma''\ \ \langle c_1, \sigma'' \rangle \to \sigma'}{\langle c_0\ \texttt{;}\ c_1, \sigma \rangle \to \sigma'}$$

Conditionals:

$$\frac{\langle b,\sigma\rangle \to true \;\; \langle c_0,\sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma\rangle \to false \;\; \langle c_1,\sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1,\sigma\rangle \to \sigma'}$$

While-loops:

$$\frac{\langle b,\sigma\rangle \to false}{\langle \texttt{while}\ b\ \texttt{do}\ c,\sigma\rangle \to \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \;\; \langle c,\sigma\rangle \to \sigma'' \;\; \langle \texttt{while}\ b\ \texttt{do}\ c,\sigma''\rangle \to \sigma'}{\langle \texttt{while}\ b\ \texttt{do}\ c,\sigma\rangle \to \sigma'}$$

equivalence of commands

$c_0 \sim c_1$ iff $(\forall \sigma, \sigma' \in \sum . \langle c_0,\sigma\rangle \to \sigma' \Leftrightarrow \langle c_1,\sigma\rangle \to \sigma')$