COMANDI

Com : $c \to \underline{skip} \mid \underline{x := e} \mid c ; c \mid$
while b do $c \mid$
if b then c else c

$x := e$

locazione
accesso in scrittura
target / l-value

valore
accessi lettura
source / r-value

$$\frac{\frac{e : \tau}{d : \Delta}}{c}$$

Semantica statica : per i comandi verifica che il comando sia ben formato (contenga solo costrutti che rispettano i vincoli del PL)

$FI : Com \to \wp(Id)$ $DI : Com \to \wp(Id)$

$FI(x := e) = \{x\} \cup Fi(e)$ $DI(x := e) = \emptyset$

tutti gli Id devono trovare significato nell'ambiente di esecuzione (locazione / valore)

Assegnamento ben formato $\Delta \vdash c$

$$\frac{\Delta \vdash e : \tau}{\Delta \vdash x := e} , \Delta(x) = \tau loc$$

Semantica dinamica: L'assegnamento è l'unico comando che genera trasformazioni della memoria

$\Gamma = (C \times Mem) \cup Mem$ configurazioni

$T = Mem$ config. terminali

$\rho \vdash \langle c, \sigma \rangle \to \langle c', \sigma' \rangle$ composizione, while, if

oppure

$\to \sigma'$ (non c'è più nulla da eseguire) skip, assegnamento

$$\frac{\rho \vdash \langle e, \sigma \rangle \to^* \langle k, \sigma \rangle}{\rho \vdash \langle x := e, \sigma \rangle \to \underbrace{\sigma[\ell \mapsto k]}_{\sigma'}}$$

$\rho(x) = \ell$

ℓ è la locazione associata a x da ρ e in ℓ metto k

Esempio

$x := x * y$ → dobbiamo considerare un ambiente e una memoria

$\Delta = [x \mapsto intloc, y \mapsto int]$

$\rho = [x \mapsto \ell_x, y \mapsto 2]$

$\sigma = [\ell_x \mapsto 3]$

Semantica statica:
$$\frac{\Delta \vdash x * y : int \ ?}{\Delta \vdash x := x * y}$$
$\Delta(x) = intloc$ ok

$$\frac{\Delta \vdash x : int \ ? \qquad \Delta \vdash y : int ?}{\Delta \vdash x * y : int}$$

$\Delta(x) = intloc$

$\Delta \vdash x : int$ ok

$\Delta(y) = int$

$\Delta \vdash y : int$ ok

$\Rightarrow \Delta \vdash x := x * y$ è derivabile e quindi $x := x * y$ è ben formato

Semantica dinamica: $\rho \vdash \langle x := x*y, \sigma \rangle \to ?$

$$\frac{\rho \vdash \langle x*y, \sigma \rangle \to^{*} \langle k, \sigma \rangle}{\rho \vdash \langle x := x*y, \sigma \rangle \to ?} \quad (*)$$

$$\frac{\rho \vdash \langle x, \sigma \rangle \to \langle 3, \sigma \rangle}{\rho \vdash \langle x*y, \sigma \rangle \to \langle 3*y, \sigma \rangle} \quad \rho(x) = l_x, \ \sigma(l_x) = 3$$

$$\frac{\rho \vdash \langle y, \sigma \rangle \to \langle 2, \sigma \rangle}{\rho \vdash \langle 3*y, \sigma \rangle \to \langle 3*2, \sigma \rangle} \quad \rho(y) = 2$$

$$\rho \vdash \langle 3*2, \sigma \rangle \to \langle \underset{k}{6}, \sigma \rangle, \quad 3*2 = 6$$

Quindi torniamo all'esecuzione dell'assegnamento $(*)$

$$\frac{\rho \vdash \langle x*y, \sigma \rangle \to^{*} \langle 6, \sigma \rangle}{\rho \vdash \langle x := x*y, \sigma \rangle \to \sigma[l_x \mapsto 6]} \quad \rho(x) = l_x$$

$$[l_x \mapsto 3][l_x \mapsto 6] = [l_x \mapsto 6] = \sigma'$$

STRUTTURE DI CONTROLLO → decido come continuare l'esecuzione, ovvero sul flusso di controllo

- comandi di selezione
- comandi di iterazione

- selettore a due vie (if-then-else)
- selettore a più vie (switch-case ...)

<u>Selettore a due vie</u> → if annidati

```
if (sum==0)
    if (count==0)
        result 0; fi
else result=1
```

?

<u>Selettore a più vie</u> switch - case

```
switch (espressione) {
    case const_expr : ---
    case ...
    [default : ... ]
```

Semantica del if-then-else

Semantica statica:

$$FI(\text{if } e \text{ then } c_1 \text{ else } c_2) = FI(e) \cup FI(C_1) \cup FI(C_2)$$

$$DI(\text{if } e \text{ then } c_1 \text{ else } c_2) = DI(C_1) \cup DI(C_2)$$

$$\frac{\Delta \vdash e : bool \quad \Delta \vdash c_1 \quad \Delta \vdash c_2}{\Delta \vdash \text{if } e \text{ then } c_1 \text{ else } c_2}$$

Semantica dinamica :

$$\frac{\rho \vdash \langle e, \sigma \rangle \longrightarrow^* \langle t, \sigma \rangle}{\rho \vdash \langle \text{if } e \text{ then } c_1 \text{ else } c_2, \sigma \rangle \longrightarrow \langle \text{if } t \text{ then } c_1 \text{ else } c_2, \sigma \rangle}$$

$$\rho \vdash \langle \text{if true then } c_1 \text{ else } c_2, \sigma \rangle \longrightarrow \langle c_1, \sigma \rangle$$

$$\rho \vdash \langle \text{if false then } c_1 \text{ else } c_2, \sigma \rangle \longrightarrow \langle c_2, \sigma \rangle$$

Esempio

if $x = y$ then $x := 5$ else $x := 6$

$\Delta = [x \mapsto intloc, y \mapsto int]$

$\rho = [x \mapsto \ell x, y \mapsto 2]$

$\sigma_1 = [\ell x \mapsto 3]$ $\sigma_2 = [\ell x \mapsto 2]$

$$\frac{\overset{?\,(1)}{\Delta \vdash x = y : bool} \quad \overset{?\,(2)}{\Delta \vdash x := 5} \quad \overset{?\,(3)}{\Delta \vdash x := 6}}{\Delta \vdash \text{if } x = y \text{ then } x := 5 \text{ else } x := 6}$$

(2) $$\frac{\Delta \vdash 5 : int}{\Delta \vdash x := 5} \quad \Delta(x) = intloc \quad ✓$$

(3) $$\frac{\Delta \vdash 6 : int}{\Delta \vdash x := 6} \quad \Delta(x) = intloc \quad ✓$$

(1) $$\frac{\Delta \vdash x : \tau \quad \Delta \vdash y : \tau}{\Delta \vdash x = y : bool} \quad ✓$$

$\Delta \vdash x : int$, $\Delta(x) = intloc$

$\Delta \vdash y : int$, $\Delta(y) = int$

$\Rightarrow$ il comando e' ben formato

Dobbiamo eseguire $\rho \vdash \langle \text{if } x = y \dots, \sigma_1 \rangle \to ?$

(1)

$$\frac{\rho \vdash \langle x = y, \sigma_1 \rangle \to \langle t, \sigma_1 \rangle}{\rho \vdash \langle \text{if } x = y \text{ then} \dots \text{else} \dots, \sigma_1 \rangle \to ?} \quad (*)$$

$$\frac{\rho \vdash \langle x, \sigma_1 \rangle \to \langle 3, \sigma_1 \rangle}{\rho \vdash \langle x = y, \sigma_1 \rangle \to \langle 3 = y, \sigma_1 \rangle} \quad \rho(x) = \ell_x, \ \sigma_1(\ell_x) = 3$$

$$\frac{\rho \vdash \langle y, \sigma_1 \rangle \to \langle 2, \sigma_1 \rangle}{\rho \vdash \langle 3 = y, \sigma_1 \rangle \to \langle 3 = 2, \sigma_1 \rangle} \quad \rho(y) = 2$$

$$\rho \vdash \langle 3 = 2, \sigma_1 \rangle \to \langle \text{false}, \sigma_1 \rangle$$

$$(*) \quad \frac{\rho \vdash \langle x = y, \sigma_1 \rangle \to^* \langle \text{false}, \sigma_1 \rangle}{\rho \vdash \langle \text{if} \dots, \sigma_1 \rangle \to \langle \text{if false then} \dots \text{else}, \sigma_1 \rangle}$$

$$\rho \vdash \langle \text{if false then } x := 5 \text{ else } x := 6, \sigma_1 \rangle \to \langle x := 6, \sigma_1 \rangle$$

$$\rho \vdash \langle x := 6, \sigma_1 \rangle \to \sigma_1[\ell_x \mapsto 6]$$

Provare con $\underline{\sigma_2}$

Semantica del ;
(e di skip)

Statica:

$FI(skip) = DI(skip) = \emptyset$

$FI(C_1; C_2) = FI(C_1) \cup FI(C_2)$

$DI(C_1; C_2) = DI(C_1) \cup DI(C_2)$

$$\frac{\Delta \vdash C_1 \quad \Delta \vdash C_2}{\Delta \vdash C_1; C_2} \qquad \Delta \vdash skip$$

Dinamica

$$\frac{\rho \vdash \langle C_1, \sigma \rangle \longrightarrow \langle C_1', \sigma' \rangle}{\rho \vdash \langle C_1; C_2, \sigma \rangle \longrightarrow \langle C_1'; C_2, \sigma' \rangle}$$

$$\frac{\rho \vdash \langle C_1, \sigma \rangle \longrightarrow \sigma'}{\rho \vdash \langle C_1; C_2, \sigma \rangle \longrightarrow \langle C_2, \sigma' \rangle}$$

$$\rho \vdash \langle skip, \sigma \rangle \longrightarrow \sigma$$

Comando iterativo

iterazione determinata
È noto a priori quante volte eseguire il corpo

$i = 0 \rightarrow n$
for e ...

iterazione non determinata
il controllo dell'esecuzione è mediante guardia logica

for indice:= inizio to $\boxed{\text{fine}}$ by passo do corpo

for (expr1; expr2; expr3) corpo

Semantics

$FI(\text{while } b \text{ do } c) = FI(b) \cup FI(c)$

$DI(\text{while } b \text{ do } c) = DI(c)$

$$\frac{\Delta \vdash b : \underline{bool} \qquad \Delta \vdash c}{\Delta \vdash \text{while } b \text{ do } c}$$

1

$$\frac{\rho \vdash \langle b, \sigma \rangle \to^* \langle t, \sigma \rangle}{\rho \vdash \langle \text{while } b \text{ do } c, \sigma \rangle \to \langle \text{while } t \text{ do } c, \sigma \rangle}$$

$\rho \vdash \langle \text{while } false \text{ do } c, \sigma \rangle \to \sigma$

2 $\rho \vdash \langle \text{while } true \text{ do } c, \sigma \rangle \to \langle c; \text{while } b \text{ do } c, \sigma \rangle$

$1 + 2 \equiv$

$$\frac{\rho \vdash \underline{\langle b, \sigma \rangle \to^* \langle true, \sigma \rangle}}{\rho \vdash \langle \text{while } b \text{ do } c, \sigma \rangle \to \langle c; \text{while } b \text{ do } c, \sigma \rangle}$$