Informatica — 2014-07-15

Nota: Scrivete su **tutti** i fogli nome e matricola.

Esercizio 1. *Si dimostri che una funzione $f : \mathcal{P}(U) \to \mathcal{P}(U)$ continua secondo Scott è anche monotona.*

Esercizio 2. *Si consideri l'estensione di IMP ottenuta aggiungendo il seguente comando* try*:*

$$\mathsf{try}\ c\ \mathsf{for}\ e \neq 0$$

dove c è un comando ed e è un'espressione intera. La semantica informale di tale costrutto è la seguente. Inizialmente, si esegue il comando c. Poi, si verifica se la condizione $e \neq 0$ è vera. Se lo è, non si fa nulla. Altrimenti, tutte le variabili del programma vengono reimpostate al valore che esse avevano precedentemente all'esecuzione di c.

1. *Si formalizzi la semantica big-step del comando* try *aggiungendo una o più regole di inferenza alla semantica di IMP.*

2. *Si consideri l'ulteriore estensione di IMP ottenuta aggiungendo le espressioni intere $f(e)$ e $g(e)$, dove e è un'espressione intera, e f, g sono funzioni non meglio precisate. Per ogni comando sottostante, se ne definisca uno equivalente senza usare il* try*. Si giustifichi* informalmente *la risposta.*

$$\begin{aligned} c_1 = &\quad \mathsf{try}\ x := y;\ y := y + z + x\ \mathsf{for}\ y \neq 0 \\ c_2 = &\quad \mathsf{try}\ \mathsf{while}\ f(x) = 0\ \mathsf{do}\ x := g(x)\ \mathsf{for}\ f(x) \neq 0 \\ c_3 = &\quad \mathsf{try}\ \mathsf{while}\ f(x) = 0\ \mathsf{do}\ x := g(x)\ \mathsf{for}\ g(x) \neq 0;\ y := 0 \end{aligned}$$

Soluzione (bozza). Punto 1. Basta aggiungere le regole

$$\frac{\langle c,\ \sigma\rangle \to_b \sigma' \qquad \langle e,\ \sigma'\rangle \to_e v \neq 0}{\langle \mathsf{try}\ c\ \mathsf{for}\ e \neq 0,\ \sigma\rangle \to_b \sigma'}[Try - True]$$

$$\frac{\langle c,\ \sigma\rangle \to_b \sigma' \qquad \langle e,\ \sigma'\rangle \to_e 0}{\langle \mathsf{try}\ c\ \mathsf{for}\ e \neq 0,\ \sigma\rangle \to_b \sigma}[Try - False]$$

Punto 2. Per c_1 basta osservare che alla fine il valore di y sarà quello che nello stato iniziale era $y + z + y$. Quindi lo stesso effetto si può ottenere con il comando

$$c_1 \equiv \quad \mathsf{if}\ y + z + y \neq 0\ \mathsf{then}\ x := y;\ y := y + z + x\ \mathsf{else}\ \mathsf{skip}$$

Per c_2 basta osservare che quando il while termina sicuramente si ha $f(x) \neq 0$, quindi il try è ridondante in questo caso.

$$c_2 \equiv \quad \mathsf{while}\ f(x) = 0\ \mathsf{do}\ x := g(x)$$

Per c_3, osserviamo che il valore finale di y è 0. Inoltre, solo la variabile x viene usata nel try. Questo ci consente di scrivere un comando equivalente che usa y come variabile "d'appoggio". Per prima cosa memorizziamo in y il valore di x all'inizio del try. Fatto ciò, eseguiamo il corpo del try (cioè il ciclo while). Dopo, controlliamo se $g(x) \neq 0$ e, se non vale, reimpostiamo x al valore precedente. Infine azzeriamo y.

$$\begin{aligned} c_3 \equiv \quad & y := x;\ \Big(\textsf{while } f(x) = 0 \textsf{ do } x := g(x)\Big); \\ & \Big(\textsf{if } g(x) \neq 0 \textsf{ then skip else } x := y\Big);\ y := 0 \end{aligned}$$

□

Esercizio 3. *Si consideri il comando di IMP* $w = (\textsf{while } x-32 \neq 0 \textsf{ do } x := 2\cdot x)$ *e si descriva* informalmente *per quali stati iniziali* σ *si ha che il comando* w termina*, ovvero che* $\langle w, \sigma\rangle \to_b \sigma'$ *per qualche* σ'*.*
Dopo, si dimostri formalmente *che per ogni stato iniziale* σ *tale che* $\sigma(x) < 0$ *il comando* w *non termina, secondo la traccia seguente. Per prima cosa, si dimostri che tale fatto segue da* $(\to_b) \subseteq p$ *dove* p *è la proprietà*

$$p(c, \sigma, \sigma') = \text{“}\sigma(x) < 0 \implies \left(\begin{array}{c} c \neq w \ \wedge \\ c \neq (x := 2 \cdot x; w) \ \wedge \\ c = (x := 2 \cdot x) \implies \sigma'(x) < 0 \end{array}\right) \text{”}$$

Quindi, si dimostri $(\to_b) \subseteq p$ *sfruttando il principio di induzione e la definizione della semantica di IMP. Per brevità, si richiedono solo i casi relativi alle regole [Comp] e [While-True].*

Soluzione (bozza). Si noti che l'esercizio è una variante della dimostrazione di non totalità della semantica big-step di IMP che è presente nelle note.

Si ha che il comando termina quando lo stato iniziale σ soddisfa $\sigma(x) \in \{2^0, 2^1, \ldots, 2^5\}$. In tutti gli altri casi è impossibile raggiungere $32 = 2^5$ moltiplicando per 2.

Se $\sigma(x) < 0$, verifichiamo che la non terminazione di w è una conseguenza di $(\to_b) \subseteq p$. Se infatti, per assurdo, w terminasse, avremmo $\langle w, \sigma\rangle \to_b \sigma'$ per qualche σ'. Per l'inclusione, avremmo quindi $p(w, \sigma, \sigma')$ da cui per definizione di p si ha $w \neq w$ — assurdo.

Verifichiamo ora che p sia induttiva, nei casi [Comp] e [While-True]. Ricordiamo la [Comp]:

$$\frac{\langle c_1\ ,\ \sigma\rangle \to_b \sigma' \qquad \langle c_2\ ,\ \sigma'\rangle \to_b \sigma''}{\langle c_1; c_2\ ,\ \sigma\rangle \to_b \sigma''}[Comp]$$

Come ipotesi induttive, abbiamo

$$p(c_1, \sigma, \sigma') \wedge p(c_2, \sigma', \sigma'')$$

da cui dobbiamo ricavare $p(c_1; c_2, \sigma, \sigma'')$. Per farlo assumiamo $\sigma(x) < 0$ e verifichiamo le tre conseguenze.

- $c_1; c_2 \neq w$: ovvio.

- $c_1; c_2 \neq x := 2 \cdot x; w$: supponiamo per assurdo che siano uguali e che quindi $c_1 = x := 2 \cdot x$ e $c_2 = w$. Sfruttando $p(c_1, \sigma, \sigma')$, siccome $\sigma(x) < 0$, otteniamo $c_1 = (x := 2 \cdot x) \implies \sigma'(x) < 0$. La premessa è vera, quindi $\sigma'(x) < 0$. Sfruttando ora $p(c_2, \sigma', \sigma'')$, siccome $\sigma'(x) < 0$, si ha che $c_2 \neq w$ — assurdo.

- $c_1; c_2 = x := 2 \cdot x \implies \sigma''(x) < 0$: l'ipotesi è banalmente falsa, quindi l'implicazione è vera.

Per la [While-True]:

$$\frac{\langle e, \sigma\rangle \to_e v \neq 0 \qquad \langle c; \mathsf{while}\ e \neq 0\ \mathsf{do}\ c\ ,\ \sigma\rangle \to_b \sigma'}{\langle \mathsf{while}\ e \neq 0\ \mathsf{do}\ c\ ,\ \sigma\rangle \to_b \sigma'}[While - true]$$

Sia $w' = (\mathsf{while}\ e \neq 0\ \mathsf{do}\ c)$. Come ipotesi induttiva abbiamo $p(c; w', \sigma, \sigma')$, e dobbiamo ricavare $p(w', \sigma, \sigma')$. Per farlo, assumiamo $\sigma(x) < 0$ e verifichiamo le tre parti:

- $w' \neq w$: per assurdo, sia $w' = w$. In tal caso $c = (x := 2 \cdot x)$, per cui $c; w' = x := 2 \cdot x; w$. Dall'ipotesi induttiva, visto che $\sigma(x) < 0$ si ha $c; w' \neq x := 2 \cdot x; w$ — assurdo.

- $w' \neq x := 2 \cdot x; w$: ovvio.

- $w' = x := 2 \cdot x \implies \sigma'(x) < 0$: l'ipotesi è banalmente falsa, quindi l'implicazione è vera.

□

Nome _____ Matricola _____

Esercizio 4. *Si dimostri formalmente la validità della tripla di Hoare seguente riempiendo le linee sottostanti con opportune asserzioni.*

$\{x = 0 \wedge y = Y\}$

_____

while $y \neq 0$ do

_____

_____

if $y < 0$ then

_____

_____

$y := -1 - y$

else

_____

_____

$y := 1 - y$

_____

$x := x + 1$

_____

$\{x = |Y|\}$

Giustificare qui sotto eventuali usi della regola *PrePost*.

_____

_____

_____

_____

_____

_____

_____

Soluzione (bozza).

$\{x = 0 \wedge y = Y\}$
$\{INV: \ x + |y| = |Y|\}$ (1)
while $y \neq 0$ do
  $\{INV \wedge y \neq 0\}$
  if $y < 0$ then
    $\{INV \wedge y \neq 0 \wedge y < 0\}$
    $\{x + 1 + | - 1 - y| = |Y|\}$ (2)
    $y := -1 - y$
  else
    $\{INV \wedge y \neq 0 \wedge y \geq 0\}$
    $\{x + 1 + |1 - y| = |Y|\}$ (3)
    $y := 1 - y$
  $\{x + 1 + |y| = |Y|\}$
  $x := x + 1$
$\{INV \wedge y = 0\}$
$\{x = |Y|\}$ (4)

Le PrePost sono giustificate come segue.

(1): ovvia.

(2): siccome $y < 0$ e y è intero, si ha $y \leq -1$ quindi $-y - 1 \geq 0$ da cui $| - 1 - y| = -1 - y = -1 + |y|$. Quindi,

$$x + 1 + | - 1 - y| = x + 1 - 1 + |y| = x + |y| = |Y|$$

dove l'ultima uguaglianza deriva dall'invariante.

(3): siccome $y \geq 0$, $y \neq 0$ si ha che $y > 0$. Visto che y è intero, si ha $y \geq 1$ per cui $1 - y \leq 0$. Quindi,

$$x + 1 + |1 - y| = x + 1 - 1 + y = x + y = x + |y| = |Y|$$

dove l'ultima uguaglianza deriva dall'invariante.

(4): per l'invariante e $y = 0$, si ha che $|Y| = x + |y| = x + |0| = x$.

□