Informatica — 2019-07-18

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

Esercizio 1. *Si forniscano tutte le regole della semantica big step ($\to_b$) relative ai comandi* skip, if *ed $x := e$. Si commenti brevemente su come vengono usati gli stati σ che compaiono in queste regole.*

Esercizio 2. *Le seguenti regole definiscono induttivamente l'insieme S delle sequenze di numeri naturali (regole $[S0], [S1]$), una relazione $\Sigma \in \mathcal{P}(S \times \mathbb{N})$ (regole $[\Sigma 0], [\Sigma 1]$) e una relazione $R \in \mathcal{P}(S \times S \times S)$ (regole $[R0], [R1]$). Sotto, n, m, k indicano naturali mentre s, w, z indicano sequenze in S.*

$$\frac{}{\epsilon}[S0] \qquad \frac{s}{n : s}(n \in \mathbb{N})[S1] \qquad \frac{}{\Sigma(\epsilon, 0)}[\Sigma 0] \qquad \frac{\Sigma(s, k)}{\Sigma(n : s, n + k)}[\Sigma 1]$$

$$\frac{}{R(\epsilon, s, s)}[R0] \qquad \frac{R(s, w, z)}{R(n : s, w, n : z)}[R1]$$

1. *[20%] Si fornisca un naturale $k \in \mathbb{N}$ per cui valga $\Sigma(1 : 2 : 3 : \epsilon, k)$ e una sequenza z tale che $R(1 : 2 : \epsilon, 1 : 2 : 3 : \epsilon, z)$. Si giustifichi la risposta esibendo opportune derivazioni.*

2. *[20%] Si enunci il principio di induzione associato all'insieme S.*

3. *[10%] Si consideri l'enunciato seguente:*

   $$\forall s, w, z \in S, n, m \in \mathbb{N}.\ \Sigma(s, n) \land \Sigma(w, m) \land R(s, w, z) \implies \Sigma(z, n + m)$$

   Si riscriva l'enunciato in modo logicamente equivalente nella forma

   $$\forall s \in S.\ p(s)$$

   per un qualche predicato p.

4. *[50%] Si concluda la dimostrazione dell'enunciato visto sopra usando il principio di induzione associato a S.*

Soluzione (bozza).
Parte 1.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Sigma(\epsilon, 0)}[\Sigma 0]}{\Sigma(3 : \epsilon, 3)}[\Sigma 1]}{\Sigma(2 : 3 : \epsilon, 5)}[\Sigma 1]}{\Sigma(1 : 2 : 3 : \epsilon, 6)}[\Sigma 1]$$

$$\dfrac{\dfrac{\dfrac{}{R(\epsilon, 1 : 2 : 3 : \epsilon, 1 : 2 : 3 : \epsilon)}[R0]}{R(2 : \epsilon, 1 : 2 : 3 : \epsilon, 2 : 1 : 2 : 3 : \epsilon)}[R1]}{R(1 : 2 : \epsilon, 1 : 2 : 3 : \epsilon, 1 : 2 : 1 : 2 : 3 : \epsilon)}[R1]$$

Parte 2.
Sia $p(s)$ un predicato sulle sequenze. Per dimostrare che vale $\forall s \in S.\ p(s)$ è sufficiente dimostrare che:

1) $p(\epsilon)$
2) $\forall n \in \mathbb{N}, s.\ p(s) \implies p(n : s)$

Parte 3. Basta scegliere

$$p(s) : \quad \forall w, z \in S, n, m \in \mathbb{N}.\ \Sigma(s, n) \land \Sigma(w, m) \land R(s, w, z) \implies \Sigma(z, n + m)$$

Parte 4. Caso $[S0]$**.**
Verifichiamo $p(\epsilon)$, cioè:

$$\forall w, z \in S, n, m \in \mathbb{N}.\ \Sigma(\epsilon, n) \wedge \Sigma(w, m) \wedge R(\epsilon, w, z) \implies \Sigma(z, n+m)$$

Assumiamo quindi

$$\begin{aligned} IP1:&\ \Sigma(\epsilon, n) \\ IP2:&\ \Sigma(w, m) \\ IP3:&\ R(\epsilon, w, z) \end{aligned}$$

e dimostriamo la tesi $\Sigma(z, n+m)$.

Invertendo $IP1$, notiamo che può essere ricavata solo tramite $[\Sigma 0]$ e quindi $n = 0$. Invertendo $IP3$, notiamo che può essere ricavata solo tramite $[R0]$ e quindi $w = z$.

La tesi si riscrive quindi come $\Sigma(w, m)$, che si ottiene da $IP2$.

Caso $[S1]$**.** Assumendo l'ipotesi induttiva $IP1 : p(s)$, dimostriamo la tesi $p(n : s)$, e cioè:

$$\forall w, z \in S, \bar{n}, m \in \mathbb{N}.\ \Sigma(n : s, \bar{n}) \wedge \Sigma(w, m) \wedge R(n : s, w, z) \implies \Sigma(z, \bar{n} + m)$$

Assumiamo quindi:

$$\begin{aligned} IP2:&\ \Sigma(n : s, \bar{n}) \\ IP3:&\ \Sigma(w, m) \\ IP4:&\ R(n : s, w, z) \end{aligned}$$

e dimostriamo la nuova tesi $\Sigma(z, \bar{n} + m)$.

Invertendo $IP2$, notiamo che può essere ricavata solo tramite $[\Sigma 1]$ e quindi otteniamo $IP6 : \Sigma(s, \hat{n})$ con $\bar{n} = n + \hat{n}$.

Invertendo $IP4$, notiamo che può essere ricavata solo tramite $[R1]$ e quindi otteniamo $IP7 : R(s, w, \hat{z})$ con $z = n : \hat{z}$.

Ora, usiamo $IP1$, che si scrive come

$$\forall w', z' \in S, n', m' \in \mathbb{N}.\ \Sigma(s, n') \wedge \Sigma(w', m') \wedge R(s, w', z') \implies \Sigma(z', n' + m')$$

Scegliamo $w' = w, z' = \hat{z}, n' = \hat{n}, m' = m$ e otteniamo:

$$\Sigma(s, \hat{n}) \wedge \Sigma(w, m) \wedge R(s, w, \hat{z}) \implies \Sigma(\hat{z}, \hat{n} + m)$$

L'antecedente segue da $IP6, IP3, IP7$, quindi possiamo assumere la conseguente $\Sigma(\hat{z}, \hat{n} + m)$. Di qui, usando la regola $[\Sigma 1]$ otteniamo $\Sigma(n : \hat{z}, n + \hat{n} + m)$ e cioè $\Sigma(z, \bar{n} + m)$ che è la tesi.

□

Esercizio 3. *Si consideri l'estensione del linguaggio IMP ottenuta aggiungendo il comando* fork x*, la cui semantica è data dalle regole seguenti:*

$$\frac{}{\langle \mathsf{fork}\ x, \sigma\rangle \to_b \sigma[x \mapsto 0]}[Fork-0] \qquad \frac{}{\langle \mathsf{fork}\ x, \sigma\rangle \to_b \sigma[x \mapsto 1]}[Fork-1]$$

1. *[10%] Si dica se il linguaggio così esteso soddisfa il determinismo e/o la totalità, giustificando informalmente la risposta.*

2. *[90%] Si fornisca un comando c di questo linguaggio che soddisfi la proprietà:*

$$\exists \sigma.\ \forall n \in \mathbb{N}.\ \langle c, \sigma\rangle \to_b \sigma[w \mapsto 0][x \mapsto 0][y \mapsto 0][z \mapsto n]$$

Si giustifichi la risposta in modo informale ma dettagliato.

Soluzione (bozza).

Parte 1. La totalità non vale, così come non valeva in IMP. Il comando $\mathsf{while}\ 1 \neq 0\ \mathsf{do\ skip}$ eseguito a partire a un qualunque stato iniziale non termina.

Il determinismo vale in IMP, ma non vale più nel linguaggio esteso. Per esempio, se lo stato iniziale σ associa ad ogni variabile il valore zero, si ha $\langle \mathsf{fork}\ x, \sigma\rangle \to_b \sigma$ e anche $\langle \mathsf{fork}\ x, \sigma\rangle \to_b \sigma[x \mapsto 1]$, dove i due stati finali differiscono sulla x.

Parte 2. Un possibile comando c è:

$$
\begin{array}{l}
\mathsf{fork}\ y; \\
\mathsf{while}\ y \neq 0\ \mathsf{do} \\
\qquad z := z + 1; \\
\qquad \mathsf{fork}\ y
\end{array}
$$

Lo stato iniziale σ è quello che pone tutte le variabili a zero. Dobbiamo quindi dimostrare che il comando può terminare con qualunque valore $n \in \mathbb{N}$ come valore di z nello stato finale, con le altre variabili impostate a zero.

In caso di terminazione, nello stato finale le variabili w, x sono ancora a zero perchè non sono state modificate. La y è a zero perché siamo appena usciti dal while. Quindi ci possiamo concentrare su ottenere $z = n$ per ogni possibile naturale n.

Se la prima fork imposta y a zero ($Fork - 0$), non viene fatta nessuna iterazione del while e il programma termina con $z = 0$. Quindi $n = 0$ può essere ottenuto.

Altrimenti, la prima fork viene fatta secondo la regola $Fork{-}1$ e ciò causa un'iterazione del while, con z che diventa 1. Se la seconda fork imposta y a zero, il while termina, facendo terminare il programma con $z = 1$. Quindi $n = 1$ è ottenibile.

Se anche la seconda fork imposta y a 1, possiamo raggiungere gli n più grandi, continuando a ciclare nel while.

Riassumendo il ragionamento fatto sopra, per ottenere alla fine un qualunque $n \in \mathbb{N}$ come valore di z, è sufficiente che le prime n fork che si eseguono impostino sempre y a 1, e che la fork successiva imposti invece y a 0. In questo modo, il while compie esattamente n iterazioni, e quindi il programma termina con $z = n$.

□

Nome _____ Matricola _____

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

$\{x = A \wedge y = B\}$

---

while $x \neq y$ do

---

if $x > y$ then

$x := x - 1$

else

$y := y - 1$

$\{x = y = \min(A, B)\}$

Giustificare qui sotto eventuali usi della regola $PrePost$.

Soluzione (bozza).

$\{x = A \wedge y = B\}$ (1)
$\{INV : \min(x, y) = \min(A, B)\}$
while $x \neq y$ do
    $\{INV \wedge x \neq y\}$
    if $x > y$ then
        $\{INV \wedge x \neq y \wedge x > y\}$ (2)
        $\{\min(x - 1, y) = \min(A, B)\}$
        $x := x - 1$
    else
        $\{INV \wedge x \neq y \wedge \neg(x > y)\}$ (3)
        $\{\min(x, y - 1) = \min(A, B)\}$
        $y := y - 1$
$\{INV \wedge \neg(x \neq y)\}$ (4)
$\{x = y = \min(A, B)\}$

Per le PrePost:

1) banale sostituzione.

2) Essendo $x > y$, $\min(x, y) = y$ e siccome lavoriamo sugli interi, possiamo diminuire di 1 la x senza alterare il minimo. Quindi, $\min(x - 1, y) = \min(x, y) = y$. Infine, usando INV si ha $\min(x - 1, y) = \min(x, y) = \min(A, B)$, che è la tesi.

3) Essendo $\neg(x > y)$, e anche $x \neq y$, otteniamo $x < y$. Quindi, $\min(x, y) = x$ e siccome lavoriamo sugli interi, possiamo diminuire di 1 la y senza alterare il minimo. Quindi, $\min(x, y - 1) = \min(x, y) = x$. Infine, usando INV si ha $\min(x, y - 1) = \min(x, y) = \min(A, B)$, che è la tesi.

4) Per ipotesi si ha $x = y$, quindi $x = y = \min(x, y)$ che per INV coincide con $\min(A, B)$.

□