Informatica — 2023-02-13

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

Esercizio 1. *Si enuncino tutti i risultati relativi al determinismo e alla totalità della semantica delle espressioni ($\to_e$) e dei comandi ($\to_b$) di IMP.*

Esercizio 2. *Le seguenti regole definiscono induttivamente l'insieme T degli alberi di naturali (regole $[T0], [T1]$) e una relazione $R \in \mathcal{P}(T \times T \times \mathbb{N})$ (regole $[R0], [R1]$). Sotto, n, k indicano naturali, mentre s, d, t, u indicano alberi in T.*

$$\frac{}{n}(n \in \mathbb{N})[T0] \qquad \frac{s \quad d}{(s,d)}[T1] \qquad \frac{}{R(n, n+1, 1)}[R0] \qquad \frac{R(s,s',k) \quad R(d,d',k')}{R((s,d),\ (s',d'),\ k+k')}[R1]$$

1. *[20%] Si trovino due alberi t_0, t_1 tale per cui valga $R(t_0, t_1, 3)$. Si giustifichi la risposta esibendo una derivazione.*
2. *[20%] Si enunci il principio di induzione associato alla relazione R.*
3. *[10%] Si consideri l'enunciato seguente:*

   $$\forall t_0, t_1, t_2 \in T, n_0, n_1 \in \mathbb{N}.\ R(t_0, t_1, n_0) \wedge R(t_1, t_2, n_1) \implies n_0 = n_1$$

   Si riscriva l'enunciato in modo logicamente equivalente nella forma

   $$\forall t, u \in T, n \in \mathbb{N}.\ R(t, u, n) \implies p(t, u, n)$$

   per un qualche predicato p.
4. *[50%] Si concluda la dimostrazione dell'enunciato visto sopra usando il principio di induzione <u>associato a R</u>.*

Soluzione (bozza).
Parte 1.
Una tra le possibili derivazioni è:

$$\cfrac{\cfrac{}{R(10,11,1)} \qquad \cfrac{\cfrac{}{R(20,21,1)} \quad \cfrac{}{R(30,31,1)}}{R((20,30),(21,31),2)}}{R((10,(20,30)),(11,(21,31)),3)}$$

Parte 2. Affinché per ogni t, u, n tali che $R(t, u, n)$ valga $p(t, u, n)$ basta che:

$$\begin{array}{ll} R0) & \forall n.\ p(n, n+1, 1) \\ R1) & \forall s, s', k, d, d', k'.\ p(s, s', k) \wedge p(d, d', k') \implies p((s,d), (s',d'), k+k') \end{array}$$

Parte 3.
Basta definire

$$p(t, u, n) : \forall t_2 \in T, n_1 \in \mathbb{N}.\ R(u, t_2, n_1) \implies n = n_1$$

Parte 4. Caso $[R0]$.
Dobbiamo dimostrare $p(n, n+1, 1)$ e cioè

$$\forall t_2 \in T, n_1 \in \mathbb{N}.\ R(n+1, t_2, n_1) \implies 1 = n_1$$

Assumiamo quindi $IP1 : R(n+1, t_2, n_1)$ e dimostriamo la nuova tesi $1 = n_1$.

Invertendo $IP1$ notiamo che può essere generata solo da $[R0]$ e quindi $t_2 = n+1$ e $n_1 = 1$, da cui la tesi.

Caso $[R1]$.

Assumiamo come ipotesi induttive $IP1 : p(s, s', k)$ e $IP2 : p(d, d', k')$ e dimostriamo la tesi $p((s, d), (s', d'), k + k')$.

Riscrivendo tutto, otteniamo:

$$\begin{array}{rl} IP1: & \forall \bar{t}_2 \in T, \bar{n}_1 \in \mathbb{N}.\ R(s', \bar{t}_2, \bar{n}_1) \implies k = \bar{n}_1 \\ IP2: & \forall \hat{t}_2 \in T, \hat{n}_1 \in \mathbb{N}.\ R(d', \hat{t}_2, \hat{n}_1) \implies k' = \hat{n}_1 \\ tesi: & \forall \tilde{t}_2 \in T, \tilde{n}_1 \in \mathbb{N}.\ R((s', d'), \tilde{t}_2, \tilde{n}_1) \implies k + k' = \tilde{n}_1 \end{array}$$

Assumiamo quindi $IP3 : R((s', d'), \tilde{t}_2, \tilde{n}_1)$ e dimostriamo la nuova tesi $k + k' = \tilde{n}_1$.

Invertendo $IP3$ notiamo che può essere generata solo da $[R1]$ e quindi esistono $v_0, v_1 \in T, n'_0, n'_1 \in \mathbb{N}$ tali che

$$\begin{array}{l} \tilde{t}_2 = (v_0, v_1) \\ \tilde{n}_1 = n'_0 + n'_1 \\ IP4: \ R(s', v_0, n'_0) \\ IP5: \ R(d', v_1, n'_1) \end{array}$$

Usiamo $IP1$ scegliendo $\bar{t}_2 = v_0, \bar{n}_1 = n'_0$ assieme a $IP4$, e ricaviamo $k = n'_0$.

Usiamo $IP2$ scegliendo $\hat{t}_2 = v_1, \hat{n}_1 = n'_1$ assieme a $IP5$, e ricaviamo $k' = n'_1$.

Usando le equazioni menzionate sopra, la tesi segue da $k + k' = n'_0 + n'_1 = \tilde{n}_1$.

$\square$

Esercizio 3. *In un'estensione $\mathcal{S}$ del linguaggio IMP con i vettori e tre funzioni intere* $\mathsf{f}, \mathsf{g}, \mathsf{h}$*, si consideri il seguente comando, dove v e w sono vettori interi di lunghezza* 100*.*

$$c_{\mathcal{S}} = \Big[\ i := 0;\ \mathsf{while}\ i < 100\ \mathsf{do}\ \Big(\ w[i] := \mathsf{h}(\mathsf{g}(\mathsf{f}(v[i])));\ i := i + 1\ \Big)\ \Big]$$

Si consideri ora un'altra estensione $\mathcal{P}$ del linguaggio IMP con i vettori, ma senza le funzioni $\mathsf{f}, \mathsf{g}, \mathsf{h}$*. Al loro posto, troviamo invece un comando nuovo della forma*

$\mathsf{doFGH}(x, y, z, a, b, c)$ *con x, y, z, a, b, c tutte <u>variabili distinte</u>*

Questo comando di $\mathcal{P}$ ha lo stesso effetto del comando di $\mathcal{S}$ $x := \mathsf{f}(a); y := \mathsf{g}(b); z := \mathsf{h}(c)$.

1. *[70 %] Si trovi un comando $c_{\mathcal{P}}$ di $\mathcal{P}$ che abbia lo stesso effetto sui vettori v e w del comando $c_{\mathcal{S}}$ di $\mathcal{S}$. L'effetto dei comandi sulle altre variabili può essere diverso. È consentito l'uso di variabili che non compaiono in $c_{\mathcal{S}}$, così come l'uso di tutti i costrutti di $\mathcal{P}$ (e quindi di IMP) incluso le espressioni booleane come guardie. È vietato accedere ai vettori con indici al di fuori dell'intervallo* $[0, 100)$*. Si giustifichi informalmente la risposta.*

2. *[30 %] Nello svolgere il punto precedente, si faccia anche in modo che $c_{\mathcal{P}}$ non esegua il comando* doFGH *più di 150 volte. Si giustifichi informalmente la risposta.*

Soluzione (bozza).

Una soluzione semplice è quella di simulare individualmente ogni singolo uso di funzione $\mathsf{f}, \mathsf{g}, \mathsf{h}$ con una doFGH dedicata, prendendo solo il risultato che ci interessa ignorando gli altri due. Basta quindi prendere il seguente $c_{\mathcal{P}}$:

```
i := 0;
while i < 100 do
    a := v[i];
    doFGH(x, y, z, a, b, c);      // calcola f
    b := x;
    doFGH(x, y, z, a, b, c);      // calcola g
    c := y;
    doFGH(x, y, z, a, b, c);      // calcola h
    w[i] := z;
    i := i + 1
```

Questo comando è corretto, ma chiama la doFGH 300 volte. Un modo alternativo più efficiente invece è il seguente $c_{\mathcal{P}}$:

$$
\begin{array}{l}
i := 0; \\
\textsf{while } i < 102 \textsf{ do} \\
\quad \textsf{if } i < 100 \textsf{ then } a := v[i] \textsf{ else skip}; \\
\quad b := x; \\
\quad c := y; \\
\quad \textsf{doFGH}(x, y, z, a, b, c); \\
\quad \textsf{if } i \geq 2 \textsf{ then } w[i-2] := z \textsf{ else skip}; \\
\quad i := i + 1
\end{array}
$$

Sotto, indichiamo i valori delle variabili in ogni iterazione subito dopo l'esecuzione di doFGH(x, y, z, a, b, c). I valori che sono colorati, come quelli che dipendono dai valori iniziali val_b o val_c, sono valori irrilevanti ai fini del vettore risultato w.

i	x	y	z	a	b	c
0	$\mathsf{f}(v[0])$	$\mathsf{g}(val_b)$	$\mathsf{h}(val_c)$	$v[0]$	val_b	val_c
1	$\mathsf{f}(v[1])$	$\mathsf{g}(\mathsf{f}(v[0]))$	$\mathsf{h}(\mathsf{g}(val_b))$	$v[1]$	$\mathsf{f}(v[0])$	$\mathsf{g}(val_b)$
2	$\mathsf{f}(v[2])$	$\mathsf{g}(\mathsf{f}(v[1]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[0])))$	$v[2]$	$\mathsf{f}(v[1])$	$\mathsf{g}(\mathsf{f}(v[0]))$
3	$\mathsf{f}(v[3])$	$\mathsf{g}(\mathsf{f}(v[2]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[1])))$	$v[3]$	$\mathsf{f}(v[2])$	$\mathsf{g}(\mathsf{f}(v[1]))$
...						
98	$\mathsf{f}(v[98])$	$\mathsf{g}(\mathsf{f}(v[97]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[96])))$	$v[98]$	$\mathsf{f}(v[97])$	$\mathsf{g}(\mathsf{f}(v[96]))$
99	$\mathsf{f}(v[99])$	$\mathsf{g}(\mathsf{f}(v[98]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[97])))$	$v[99]$	$\mathsf{f}(v[98])$	$\mathsf{g}(\mathsf{f}(v[97]))$
100	$\mathsf{f}(v[99])$	$\mathsf{g}(\mathsf{f}(v[99]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[98])))$	$v[99]$	$\mathsf{f}(v[99])$	$\mathsf{g}(\mathsf{f}(v[98]))$
101	$\mathsf{f}(v[99])$	$\mathsf{g}(\mathsf{f}(v[99]))$	$\mathsf{h}(\mathsf{g}(\mathsf{f}(v[99])))$	$v[99]$	$\mathsf{f}(v[99])$	$\mathsf{g}(\mathsf{f}(v[99]))$

Si noti come nel cuore del ciclo si calcola f su $v[i]$, g su $\mathsf{f}(v[i-1])$, e h su $\mathsf{g}(\mathsf{f}(v[i-2])$. Questo "sfasamento" rende possibile sfruttare appieno tutte e tre le computazioni (f, g, h).

A causa dello sfasamento, nelle primissime iterazioni (il "prologo") ci sono alcuni valori colorati che non devono essere utilizzati: questi vengono scartati dal condizionale if $i \geq 2$. A parte questi valori scartati, i valori di z sono esattamente quelli desiderati e vengono quindi messi dentro w.

Sempre a causa dello sfasamento, abbiamo anche qualche valore irrilevante colorato nelle ultime iterazioni ("epilogo"). Qui usiamo if $i < 100$ per evitare di accedere all'inesistente valore $v[100]$ e ai successivi.

Il ciclo esegue doFGH 102 volte, quindi non più di 150 volte come richiesto.

□

Nome ____________________________________ Matricola ______________

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

$\{n = N >= 100\}$

____

$x := 0;$

____

$y := 1;$

____

while $x < n$ do

____

____

if $x < 42$ then

____

____

$x := x + 2;$

____

$y := y * x * (x - 1)$

else

____

____

$x := x + 1;$

____

$y := y * x$

____

$\{y = N!\}$

Si giustifichino qui sotto gli eventuali usi della regola $PrePost$.

____

____

____

____

____

Soluzione (bozza).

$\{n = N >= 100\}$ (1)
$\{1 = 0! \wedge 0 \leq n \wedge n = N \geq 100\}$
$x := 0;$
$\{1 = x! \wedge x \leq n \wedge n = N \geq 100\}$
$y := 1;$
$\{INV : y = x! \wedge x \leq n \wedge n = N \geq 100\}$
while $x < n$ do
  $\{INV \wedge x < n\}$
  if $x < 42$ then
    $\{INV \wedge x < n \wedge x < 42\}$ (2)
    $\{y(x + 2)(x + 2 - 1) = (x + 2)! \wedge x + 2 \leq n \wedge n = N \geq 100\}$
    $x := x + 2;$
    $\{yx(x - 1) = x! \wedge x \leq n \wedge n = N \geq 100\}$
    $y := y * x * (x - 1)$
  else
    $\{INV \wedge x < n \wedge \neg(x < 42)\}$ (3)
    $\{y(x + 1) = (x + 1)! \wedge x + 1 \leq n \wedge n = N \geq 100\}$
    $x := x + 1;$
    $\{yx = x! \wedge x \leq n \wedge n = N \geq 100\}$
    $y := y * x$
$\{INV \wedge \neg(x < n)\}$ (4)
$\{y = N!\}$

Per le PrePost:

1) banale aritmetica.

2) Dall'ipotesi $y = x!$ si ricava la prima tesi $y(x + 2)(x + 1) = (x + 2)!$ direttamente. Per la seconda tesi si ha $x + 2 < 42 + 2 < 100 \leq n$. La terza tesi è parte di INV.

(Nota pedante: siccome il fattoriale è di solito definito solo sui naturali, dovremmo anche dimostrare che $(x + 2)!$ è definito, ma visto che $x!$ compare nelle ipotesi possiamo assumere che $x!$ è definito e quindi $x \in \mathbb{N}$. In pratica l'ipotesi $y = x!$ sottointende $x \in \mathbb{N}$.)

3) Dall'ipotesi $y = x!$ si ricava la prima tesi $y(x + 1) = (x + 1)!$ direttamente. Per la seconda tesi da $x < n$ siccome siamo sugli interi ricaviamo $x + 1 \leq n$. La terza tesi è parte di INV.

4) Da $x \leq n$ e $\neg(x < n)$ si ha $x = n$. Da questo, $n = N$ e $y = x!$ si ha la tesi $y = N!$. $\square$