Informatica — 2022-01-24

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

Esercizio 1. *Si enuncino il teorema di Knaster-Tarski e il lemma del minimo punto fisso.*

Esercizio 2. *Le seguenti regole definiscono induttivamente l'insieme T degli alberi binari di interi (regole $[T0], [T1]$), una relazione $R \in \mathcal{P}(T \times T)$ (regole $[R0], [R1]$) e una relazione $Q \in \mathcal{P}(T \times \mathbb{Z})$ (regole $[Q0], [Q1]$). Sotto, a, b, c indicano interi mentre t, s, d indicano alberi in T.*

$$\frac{}{a}[T0](a \in \mathbb{Z}) \qquad \frac{s \quad d}{(s,d)}[T1] \qquad \frac{}{R(a,-a)}[R0] \qquad \frac{R(s_1,s_2) \quad R(d_1,d_2)}{R((s_1,d_1),(s_2,d_2))}[R1]$$

$$\frac{}{Q(a,a)}[Q0] \qquad \frac{Q(s,a) \quad Q(d,b)}{Q((s,d),a \cdot b)}[Q1]$$

1. *[20%] Si fornisca un albero t_1 con 3 interi distinti per cui valga $R(t_1,t_2) \wedge Q(t_1,a)$ per qualche t_2, a e si giustifichi la risposta esibendo due derivazioni.*

2. *[20%] Si enunci il principio di induzione associato alla relazione R.*

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

   $$\forall t_1, t_2 \in T, c \in \mathbb{Z}.\ Q(t_1,c) \wedge R(t_1,t_2) \implies \exists b \in \mathbb{Z}.\ Q(t_2,b) \wedge |c| = |b|$$

   Si riscriva l'enunciato in modo logicamente equivalente nella forma

   $$\forall t_1, t_2 \in T.\ R(t_1,t_2) \implies p(t_1,t_2)$$

   per un qualche predicato p.

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

Soluzione (bozza).
Parte 1
Una possibile soluzione è:

$$\dfrac{\dfrac{\dfrac{}{R(1,-1)} \quad \dfrac{}{R(2,-2)}}{R((1,2),(-1,-2))} \quad \dfrac{}{R(3,-3)}}{R(((1,2),3),((-1,-2),-3))}$$

$$\dfrac{\dfrac{\dfrac{}{Q(1,1)} \quad \dfrac{}{Q(2,2)}}{Q((1,2),2)} \quad \dfrac{}{Q(3,3)}}{Q(((1,2),3),6)}$$

Parte 2
Per avere $\forall t_1, t_2 \in T.\ R(t_1,t_2) \implies p(t_1,t_2)$ è sufficiente che valgano:

1) $\forall a \in \mathbb{Z}.\ p(a,-a)$
2) $\forall s_1, s_2, d_1, d_2 \in T.\ p(s_1,s_2) \wedge p(d_1,d_2) \implies p((s_1,d_1),(s_2,d_2))$

Parte 3
Basta definire $p(t_1,t_2)$ come $\forall c \in \mathbb{Z}.\ Q(t_1,c) \implies \exists b \in \mathbb{Z}.Q(t_2,b) \wedge |c| = |b|$.
Parte 4. Caso [R0].
Dimostriamo $p(a,-a)$, ovvero

$$\forall c \in \mathbb{Z}.\ Q(a,c) \implies \exists b \in \mathbb{Z}.Q(-a,b) \wedge |c| = |b|$$

Assumiamo $IP1 : Q(a, c)$ e dimostriamo $\exists b \in \mathbb{Z}.Q(-a, b) \wedge |c| = |b|$. Invertendo $IP1$, siccome è ottenibile solo da $Q0$, ricaviamo $a = c$.

Per dimostrare la tesi, scegliamo $b = -a$. La prima parte diventa $Q(-a, -a)$ che segue dalla regola $Q0$. La seconda parte diventa $|c| = |-a|$ che è equivalente a $|c| = |a|$, che a sua volta segue da $a = c$.

Caso [R1].

Come ipotesi induttive assumiamo $p(s_1, s_2)$ e $p(d_1, d_2)$, e dimostriamo $p((s_1, d_1), (s_2, d_2))$. In altre parole:

$$\begin{array}{l} IP1 : \ \forall \bar{c} \in \mathbb{Z}.\ Q(s_1, \bar{c}) \implies \exists \bar{b} \in \mathbb{Z}.Q(s_2, \bar{b}) \wedge |\bar{c}| = |\bar{b}| \\ IP2 : \ \forall \hat{c} \in \mathbb{Z}.\ Q(d_1, \hat{c}) \implies \exists \hat{b} \in \mathbb{Z}.Q(d_2, \hat{b}) \wedge |\hat{c}| = |\hat{b}| \\ tesi : \ \forall c \in \mathbb{Z}.\ Q((s_1, d_1), c) \implies \exists b \in \mathbb{Z}.Q((s_2, d_2), b) \wedge |c| = |b| \end{array}$$

Assumiamo quindi $IP3 : Q((s_1, d_1), c)$ e dimostriamo la nuova tesi $\exists b \in \mathbb{Z}.Q((s_2, d_2), b) \wedge |c| = |b|$.

Invertendo $IP3$, siccome è ottenibile solo da $Q1$, si ricava $IP4 : Q(s_1, c_s)$, $IP5 : Q(d_1, c_d)$ per qualche c_s, c_d tali che $c = c_s \cdot c_d$.

Usando $IP1$ possiamo scegliere $\bar{c} = c_s$, e combinando con $IP4$ otteniamo $IP6 : Q(s_2, \bar{b})$ e $IP7 : |c_s| = |\bar{b}|$ per qualche $\bar{b}$.

Usando $IP2$ possiamo scegliere $\hat{c} = c_d$, e combinando con $IP5$ otteniamo $IP8 : Q(d_2, \hat{b})$ e $IP9 : |c_d| = |\hat{b}|$ per qualche $\hat{b}$.

Per dimostrare la tesi, scegliamo quindi $b = \bar{b} \cdot \hat{b}$.

La parte $|c| = |b|$ si ottiene da $|c| = |c_s c_d| = |c_s||c_d| = |\bar{b}||\hat{b}| = |\bar{b}\hat{b}| = |b|$.

La parte $Q((s_2, d_2), b)$, ovvero $Q((s_2, d_2), \bar{b} \cdot \hat{b})$, si ottiene applicando $Q1$ a $IP6, IP8$.

□

Esercizio 3. *Sia $x \in Var$ un nome di variabile fissato. Per ogni $c \in Com$, indichiamo con* $\mathsf{NoX}(c)$ *il fatto che la variabile x non appare in nessun punto all'interno di c.*

1. *[20%] Si definisca tramite regole di inferenza la proprietà* $\mathsf{NoIf} \in \mathcal{P}(Com)$ *che vale su un comando se e solo se nessun comando* if *appare al suo interno.*

2. *[40%] Si definisca tramite regole di inferenza una relazione $R \in \mathcal{P}(Com \times Com)$ che soddisfi entrambe le proprietà seguenti.*

   (a) R è una funzione $Com \to Com$

   (b) $\forall c, c' \in Com.\quad \Big(\mathsf{NoX}(c) \wedge R(c, c')\Big) \implies \Big(c; x := 0 \equiv c'; x := 0 \ \wedge\ \mathsf{NoIf}(c')\Big)$

3. *[40%] Si giustifichi in modo informale la proprietà (b) nel caso in cui c è un* if.

Soluzione (bozza).

Parte 1

$$\frac{}{\mathsf{NoIf}(\mathsf{skip})}$$

$$\frac{}{\mathsf{NoIf}(y := e)}$$

$$\frac{\mathsf{NoIf}(c_1) \quad \mathsf{NoIf}(c_2)}{\mathsf{NoIf}(c_1; c_2)}$$

$$\frac{\mathsf{NoIf}(c)}{\mathsf{NoIf}(\mathsf{while}\ e \neq 0\ \mathsf{do}\ c)}$$

Parte 2

$$\frac{}{R(\mathsf{skip}, \mathsf{skip})}[R - Skip]$$

$$\frac{}{R(y := e, y := e)}[R - Let]$$

$$\frac{R(c_1, c_1') \quad R(c_2, c_2')}{R(c_1; c_2, c_1'; c_2')}[R - Comp]$$

$$\frac{R(c, c')}{R(\mathsf{while}\ e \neq 0\ \mathsf{do}\ c, \mathsf{while}\ e \neq 0\ \mathsf{do}\ c')}[R - While]$$

$$\frac{R(c_1, c_1') \quad R(c_2, c_2')}{\begin{array}{ll} R(& \mathsf{if}\ e \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2, \\ & x := 1; \mathsf{while}\ x \cdot e \neq 0\ \mathsf{do}\ (c_1'; x := 0); \mathsf{while}\ x \neq 0\ \mathsf{do}\ (c_2'; x := 0)) \end{array}}[R - If]$$

Parte 3

[Nota: in questa soluzione forniamo un commento dettagliato, ma per questo esercizio era sufficiente solo descrivere l'idea generale.]

Per la seconda proprietà, intuitivamente la relazione R deve tradurre un comando c che non usa la variabile x in un comando "quasi" equivalente c' che può usare la x ma non deve avere if al suo interno. Qui, "quasi" equivalente vuol dire che alla fine dell'esecuzione di c' il valore di x può differire da quello ottenuto eseguendo c (visto che poi viene eseguito $x := 0$ alla fine), ma sulle altre variabili i due comandi devono avere lo stesso effetto.

Che c' non contiene if è immediato dalla definizione di R. Per vedere che i due comandi c e c' sono "quasi" equivalenti, i casi diversi da $R - If$ sono semplici visto che R non cambia quei costrutti. Per il caso $R - If$, basta vedere se nello stato iniziale e ha valore zero o no.

Se all'inizio e vale non-zero, il comando

$$\mathsf{if}\ e \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

esegue c_1, mentre

$$x := 1; \mathsf{while}\ x \cdot e \neq 0\ \mathsf{do}\ (c_1'; x := 0); \mathsf{while}\ x \neq 0\ \mathsf{do}\ (c_2'; x := 0)$$

imposta x a 1, controlla $x \cdot e$ che è non-zero, esegue quindi c_1' e imposta x a zero, e quindi esce dal primo while e non entra nel secondo ciclo while. Visto che c_1' corrisponde a c_1, il comportamento è quello desiderato.

Se invece all'inizio e vale zero, il comando

$$\mathsf{if}\ e \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

esegue c_2, mentre

$$x := 1; \mathsf{while}\ x \cdot e \neq 0\ \mathsf{do}\ (c_1'; x := 0); \mathsf{while}\ x \neq 0\ \mathsf{do}\ (c_2'; x := 0)$$

imposta x a 1, controlla $x \cdot e$ che è zero, non entra quindi nel primo while, ma visto che $x = 1$ entra nel secondo ciclo while, eseguendo c_2' e impostando x a zero che causa l'uscita dal secondo while. Visto che c_2' corrisponde a c_2, il comportamento è quello desiderato.

Si noti che x non può comparire in e, c_1, c_2 e quindi l'assegnare x a valori 1 e 0 non influenza in nessun modo la valutazione di e nello stato iniziale, né l'esecuzione dei comandi c_1', c_2'.

□

Nome _____ Matricola _____

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

$\{x = N \geq 0\}$

$a := 0;$

---

while $a < x$ do

$y := 2a + 2;$

$y := y - a;$

$a := y - 1$

$\{a = N\}$

Giustificare qui sotto eventuali usi della regola *PrePost*.

Soluzione (bozza).

$\{x = N \geq 0\}$ (1)
$\{INV : 0 \leq 0 \leq x = N\}$
$a := 0;$
$\{INV : 0 \leq a \leq x = N\}$
while $a < x$ do
    $\{INV \wedge a < x\}$ (2)
    $\{0 \leq 2a + 2 - a - 1 \leq x = N\}$
    $y := 2a + 2;$
    $\{0 \leq y - a - 1 \leq x = N\}$
    $y := y - a;$
    $\{0 \leq y - 1 \leq x = N\}$
    $a := y - 1$
$\{INV \wedge \neg(a < x)\}$ (3)
$\{a = N\}$

Per le PrePost:

1) Banale aritmetica.

2) La tesi si semplifica in $0 \leq a + 1 \leq x = N$. La parte $0 \leq a + 1$ segue dell'ipotesi $0 \leq a$. La parte $a + 1 \leq x$ segue dell'ipotesi $a < x$, visto che sono tutti interi. La parte $x = N$ è un'ipotesi.

3) Dalle ipotesi $a \leq x$ e $\neg(a < x)$ otteniamo $a = x$, che usata con l'ipotesi $x = N$ fornisce la tesi.

□