

Informatica — 2026-01-26

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

Esercizio 1. Dato un insieme di regole $\mathcal{R}$ su U , si definisca l'associato operatore delle conseguenze immediate $\hat{\mathcal{R}} : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$. Si dimostri che è monotono.

Esercizio 2. Le seguenti regole definiscono induttivamente l'insieme S delle sequenze di naturali (regole $[S0], [S1]$), una relazione $R \in \mathcal{P}(S \times \mathbb{N} \times \mathbb{N})$ (regole $[R0], [R1]$), e un predicato $Q \in \mathcal{P}(S)$ (regole $[Q0], [Q1]$). Sotto, si assume $s \in S$ e $n, a, b \in \mathbb{N}$.

$$\begin{array}{c} \frac{}{\epsilon} [S0] \quad \frac{s}{n : s} (n \in \mathbb{N}) [S1] \quad \frac{}{R(\epsilon, a, a)} [R0] \quad \frac{R(s, n \cdot a, b)}{R(n : s, a, b)} [R1] \\ \frac{}{Q(4 : s)} [Q0] \quad \frac{Q(s)}{Q(n : s)} [Q1] \end{array}$$

1. [20%] Si trovi b per cui valga $R(1 : 3 : 5 : \epsilon, 10, b)$. 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:

$$\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \wedge Q(s) \implies b \text{ pari}$$

Si riscriva l'enunciato in modo logicamente equivalente nella forma

$$\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \implies p(s, a, b)$$

per un qualche predicato p . Questo sarà quindi equivalente a

$$\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \implies R(s, a, b) \wedge p(s, a, b) \quad (1)$$

4. [50%] Si assuma “IP1: $\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \implies b$ multiplo di a ”, e si dimostri l'enunciato (1) usando il principio di induzione associato a R .

Soluzione (bozza).

Parte 1.

$$\frac{\frac{\frac{\frac{}{R(\epsilon, 5 \cdot 3 \cdot 1 \cdot 10, 150)} [R0]}{R(5 : \epsilon, 3 \cdot 1 \cdot 10, 150)} [R1]}{R(3 : 5 : \epsilon, 1 \cdot 10, 150)} [R1]}{R(1 : 3 : 5 : \epsilon, 10, 150)} [R1]$$

Parte 2.

Affinché valga $\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \implies p(s, a, b)$ basta che

$$\begin{array}{l} R0) \forall a. p(\epsilon, a, a) \\ R1) \forall s, n, a, b. p(s, n \cdot a, b) \implies p(n : s, a, b) \end{array}$$

Parte 3.

Basta prendere $p(s, a, b) : Q(s) \implies b$ pari.

Parte 4.

Assumiamo come richiesto IP1 : $\forall s \in S, a, b \in \mathbb{N}. R(s, a, b) \implies b$ multiplo di a e procediamo per induzione.

Caso $[R0]$.

Dobbiamo dimostrare $R(s, a, a) \wedge p(\epsilon, a, a)$. La prima parte segue da $R0$. Rimane da dimostrare $p(\epsilon, a, a) : Q(\epsilon) \implies a$ pari.

Assumiamo $IP2 : Q(\epsilon)$ e procediamo a dimostrare la nuova tesi a pari.

Invertendo $IP2$ osserviamo che nessuna regola può produrre tale ipotesi. La tesi quindi segue vacuamente.

Caso [R1].

Assumiamo l'ipotesi induttiva $R(s, na, b) \wedge p(s, na, b)$ (quindi $IP2 : R(s, na, b)$ e $IP3 : p(s, na, b)$) e dimostriamo $R(n : s, a, b) \wedge p(n : s, a, b)$. La prima parte della tesi deriva da $R1$ con $IP2$.

Rimane da dimostrare $p(n : s, a, b)$. In altre parole:

$$\begin{aligned} IP2 : R(s, na, b) \\ IP3 : Q(s) \implies b \text{ pari} \\ \text{tesi} : Q(n : s) \implies b \text{ pari} \end{aligned}$$

Assumiamo quindi $IP4 : Q(n : s)$ e procediamo a dimostrare la nuova tesi b pari.

Invertendo $IP4$, notiamo che può essere ricavata in due modi:

- Caso [Q0]. Qui otteniamo $n = 4$, per cui $IP2$ si riscrive come $R(s, 4a, b)$. Da questo e $IP1$, otteniamo che b è multiplo di $4a$, e quindi è pari.
- Caso [Q1]. Qui otteniamo $IP5 : Q(s)$. Usando $IP5$ assieme a $IP3$, otteniamo la tesi (b pari).

□

Esercizio 3. Indichiamo con IMP il consueto linguaggio imperativo, includendo i consueti operatori aritmetico-logici ($\wedge, \neg, \leq, \dots$). Si consideri un'estensione $IMP^{\text{find}1}$ del linguaggio e il comando `find x for  $\phi$` dove ϕ è un'espressione booleana. La semantica informale di `find` è quella di cercare un numero naturale x che renda vera l'espressione ϕ . Tale numero viene quindi assegnato alla variabile x . Se ci sono più soluzioni di ϕ , non viene meglio specificato quale valore di x viene effettivamente scelto. Se invece non ci sono soluzioni, il comando `find` non termina.

1. [40%] Si definisca una possibile implementazione del comando `find` dentro IMP , eventualmente modificando anche altre variabili “di appoggio” non usate dentro ϕ . Si giustifichi la risposta.
2. [30%] Si consideri ora un'ulteriore estensione $IMP^{\text{find}2}$ di $IMP^{\text{find}1}$, aggiungendo il comando `find2 x, y for  $\phi$` che cerca invece una coppia di valori naturali che soddisfi ϕ . Si provi ad adattare, nel modo più immediato, l'implementazione precedente al comando `find2`. Si commenti sulla sua (in-)correttezza.
3. [30%] Si assuma che, nelle espressioni di IMP , ci siano tre funzioni $f(-, -)$, $g(-)$, $h(-)$ a valori naturali tali che $g(f(a, b)) = a$ e $h(f(a, b)) = b$ per ogni $a, b \in \mathbb{N}$. Si sfruttino tali funzioni per ottenere un'implementazione di `find2` dentro IMP (con eventuali variabili “di appoggio”). Si giustifichi la risposta.

Soluzione (bozza).

Parte 1.

Una possibile implementazione è:

```
t := 0;
x := 0;
while t = 0 do
  if  $\phi$  then
    t := 1
  else
    x := x + 1
```

Questo codice prova $x = 0, 1, 2, \dots$ testando ogni volta se vale ϕ . Quando ϕ diventa vera, la variabile di appoggio t viene impostata a 1 causando l'uscita dal ciclo.

Un'altra implementazione potrebbe essere:

```
x := 0;
while  $\neg\phi$  do
  x := x + 1
```

Parte 2.

Un adattamento “ingenuo” potrebbe essere

```
t := 0;
x := 0;
while t = 0 do
  y := 0;
  while t = 0 do
    if  $\phi$  then
      t := 1
    else
      y := y + 1
  if  $\phi$  then
    t := 1
  else
    x := x + 1
```

Questo però non funziona per alcune proprietà ϕ , come per esempio $x = y = 1$. Il programma sopra prova le coppie nell'ordine $(x = 0, y = 0), (x = 0, y = 1), (x = 0, y = 2), \dots$ e se non ci sono soluzioni non arriva mai a provare $(x = 1, y = 0), \dots$ perché il **while** interno non termina.

Parte 3.

Un'implementazione corretta di **find2** è:

```
t := 0;
z := 0;
while t = 0 do
  x := g(z);
  y := h(z);
  if  $\phi$  then
    t := 1
  else
    z := z + 1
```

Infatti, se il programma sopra termina, t deve essere diventato non nullo, e questo avviene solo se ϕ è diventato vero, e quindi la coppia x, y è una sua soluzione.

Viceversa, se ϕ ha almeno una soluzione, per esempio $(x = a, y = b)$, il ciclo sopra deve terminare trovando una delle soluzioni. Se per assurdo continuasse all'infinito provando $z = 0, 1, 2, \dots$ prima o poi si avrebbe $z = f(a, b)$, ma questo causa l'assegnamento $x = g(z) = a$ e $y = h(z) = b$ rendendo vera la ϕ e quindi facendo terminare il ciclo.

□

Nome _____ Matricola _____

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

$$\{n = N \geq 0\}$$
$$x := 0;$$
$$y := 0;$$

```
while  $y < n$  do
```

$$y := y + 1;$$
$$x := x - 7 + 3 * y$$
$$\{2x = 3N^2 - 11N\}$$

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

[illegible]

Soluzione (bozza).

```
{n = N ≥ 0} (1)
{0 ≤ n = N ∧ 2 · 0 = 3 · 02 - 11 · 0}
x := 0;
{0 ≤ n = N ∧ 2x = 3 · 02 - 11 · 0}
y := 0;
{INV : y ≤ n = N ∧ 2x = 3y2 - 11y}
while y < n do
  {INV ∧ y < n} (2)
  {y + 1 ≤ n = N ∧ 2(x - 7 + 3(y + 1)) = 3(y + 1)2 - 11(y + 1)}
  y := y + 1;
  {y ≤ n = N ∧ 2(x - 7 + 3y) = 3y2 - 11y}
  x := x - 7 + 3 * y
{INV ∧ ¬(y < n)} (3)
{2x = 3N2 - 11N}
```

Per le PrePost:

1) Banale aritmetica.

2) La parte di tesi $y + 1 \leq n$ segue da $y < n$, visto che sono valori interi. La parte $n = N$ è un'ipotesi. Per l'ultima equazione in tesi, si semplifica come

$$\begin{aligned} 2x - 14 + 6y + 6 &= 3y^2 + 6y + 3 - 11y - 11 \\ 2x - 8 &= 3y^2 - 11y - 8 \\ 2x &= 3y^2 - 11y \end{aligned}$$

che è in INV .

3) Da $y \leq n = N$, e $\neg(y < n)$ si ha $y = n = N$, per cui da INV si ottiene la tesi $2x = 3n^2 - 11N$.

□