Esercizi per il Corso di Informatica (LT Matematica)

Roberto Zunino

1 Esercizi sulla continuità

Nota: dal'AA 2016/17 questo argomento è stato rimosso dal corso.

Esercizio 1 *Sia U un insieme universo, e $f : \mathcal{P}(U) \to \mathcal{P}(U)$ una funzione continua secondo Scott.*

1) Dimostrare che la funzione

$$\begin{array}{l} h : \mathcal{P}(U) \to \mathcal{P}(U) \\ h(X) = X \cup f(X) \end{array}$$

è continua.

2) Dimostrare che il minimo punto fisso di h esiste e che coincide con quello di f.

Suggerimento. *Esprimete i minimi punti fissi usando il Teorema di Kleene, e ragionate insiemisticamente.*

Soluzione. *(Soluzione completa) Per il punto 1), sia $\{X_i\}_{i\in\mathbb{N}}$ una successione crescente di insiemi. Si ha:*

$$\begin{array}{ll} h(\bigcup_i X_i) & \\ = (\bigcup_i X_i) \cup f(\bigcup_i X_i) & [\mathit{def}\ h] \\ = (\bigcup_i X_i) \cup \bigcup_i f(X_i) & [\mathit{cont}\ f] \\ = \bigcup_i (X_i \cup f(X_i)) & [\mathit{insiemistica}] \end{array}$$

Per il punto 2) l'esistenza del punto fisso $fix(h)$ deriva dalla continuità di h e dal Teorema di Kleene.

Per confrontare $fix(h)$ e $fix(f)$ dimostriamo che per ogni $i \in \mathbb{N}$ si ha $h^i(\emptyset) = f^i(\emptyset)$, procedendo per induzione sul naturale i. Il caso $i = 0$ è banale. Per quello induttivo:

$$\begin{array}{ll} h^{i+1}(\emptyset) & \\ = h^i(\emptyset) \cup f(h^i(\emptyset)) & [\mathit{def}\ h] \\ = f^i(\emptyset) \cup f(f^i(\emptyset)) & [\mathit{ipotesi\ induttiva}] \\ = f^i(\emptyset) \cup f^{i+1}(\emptyset) & \\ = f^{i+1}(\emptyset) & [\mathit{cont}\ f]\ (*) \end{array}$$

dove il passaggio (*) deriva dal fatto che $f^i(\emptyset) \subseteq f^{i+1}(\emptyset)$ per le f continue (l'abbiamo dimostrato nella dimostrazione del Teorema di Kleene – è sufficiente che f sia monotona).

Dopodiché, concludiamo per il Teorema di Kleene:

$$fix(h) = \bigcup_i h^i(\emptyset) = \bigcup_i f^i(\emptyset) = fix(f)$$

Esercizio 2 *Sia U un insieme universo, A un sottoinsieme di U e $f : \mathcal{P}(U) \to \mathcal{P}(U)$ una funzione continua secondo Scott.*

1) Dimostrare che la funzione

$$\begin{array}{l} h : \mathcal{P}(U) \to \mathcal{P}(U) \\ h(X) = A \cup f(X) \end{array}$$

è continua.

2) Indicando con fix il minimo punto fisso, dimostrate che $fix(h)$ esiste e vale

$$fix(h) \supseteq fix(f)$$

3) Definite A e f in modo che $fix(h) \neq fix(f)$.

Suggerimento. *Esprimete i minimi punti fissi usando il Teorema di Kleene, e ragionate insiemisticamente. Ricordatevi che le funzioni continue secondo Scott sono anche monotone.*

Soluzione. *(Bozza) La parte 1) è immediata da*

$$\bigcup_{i\in\mathbb{N}} (A \cup Y_i) = A \cup \left(\bigcup_{i\in\mathbb{N}} Y_i \right)$$

e dalla continuità di f.

Per la parte 2), basta fare vedere che $h^i(\emptyset) \supseteq f^i(\emptyset)$. Nel dimostrarlo, si noti che per la monotonia di f si ha

$$f(h^{i+1}(X)) = f(A \cup f(h^i(X))) \supseteq f(f(h^i(X)))$$

Per la parte 3), basta prendere $f(X) = X$ e A non vuoto. (Esempi meno banali sono costruibili prendendo $f = \hat{\mathcal{R}}$ per un opportuno insieme di regole di inferenza $\mathcal{R}$.)

Esercizio 3 *Sia U un insieme universo, e $f, g : \mathcal{P}(U) \to \mathcal{P}(U)$ due funzione continue secondo Scott.*

1) Dimostrare che la funzione

$$\begin{array}{l} h : \mathcal{P}(U) \to \mathcal{P}(U) \\ h(X) = f(X) \cup g(X) \end{array}$$

è continua.

2) Indicando con fix il minimo punto fisso, dimostrate che $fix(h)$ esiste e vale

$$fix(h) \supseteq fix(f) \cup fix(g)$$

3) [Più impegnativo] Stabilite se la seguente funzione è continua o meno:

$$\begin{array}{l} h' : \mathcal{P}(U) \to \mathcal{P}(U) \\ h'(X) = f(X) \cap g(X) \end{array}$$

Suggerimento. *Esprimete i minimi punti fissi usando il Teorema di Kleene, e ragionate insiemisticamente. Per il punto 3), ricordate che la continuità implica la monotonia.*

Esercizio 4 (Impegnativo) *Sia $h : \mathcal{P}(U) \times \mathcal{P}(U) \to \mathcal{P}(U)$ una funzione binaria tale che, per ogni A, B insiemi le funzioni*

$$\begin{array}{l} f : \mathcal{P}(U) \to \mathcal{P}(U) \\ f_A(Y) = h(A, Y) \\ g : \mathcal{P}(U) \to \mathcal{P}(U) \\ g_B(X) = h(X, B) \end{array}$$

sono continue secondo Scott. Dimostrare che

$$\begin{array}{l} d : \mathcal{P}(U) \to \mathcal{P}(U) \\ d(Z) = h(Z, Z) \end{array}$$

è continua secondo Scott.

Soluzione. *Sia $\{Z_i\}_{i \in \mathbb{N}}$ una successione crescente di sottoinsiemi di U. Indichiamo con L l'unione $\bigcup_{i\in\mathbb{N}} Z_i$. Abbiamo:*

$$\begin{array}{l} d(\bigcup_{i\in\mathbb{N}} Z_i) \\ = h(\bigcup_{i\in\mathbb{N}} Z_i, \bigcup_{i\in\mathbb{N}} Z_i) \\ = h(\bigcup_{i\in\mathbb{N}} Z_i, L) \\ = g_L(\bigcup_{i\in\mathbb{N}} Z_i) \\ = \bigcup_{i\in\mathbb{N}} g_L(Z_i) \\ = \bigcup_{i\in\mathbb{N}} h(Z_i, L) \\ = \bigcup_{i\in\mathbb{N}} h(Z_i, \bigcup_{j\in\mathbb{N}} Z_j) \\ = \bigcup_{i\in\mathbb{N}} f_{Z_i}(\bigcup_{j\in\mathbb{N}} Z_j) \\ = \bigcup_{i\in\mathbb{N}} \bigcup_{j\in\mathbb{N}} f_{Z_i}(Z_j) \\ = \bigcup_{i,j\in\mathbb{N}} h(Z_i, Z_j) \end{array}$$

Per dimostrare la continuità di d basta quindi verificare che

$$\bigcup_{i,j\in\mathbb{N}} h(Z_i, Z_j) = \bigcup_{n\in\mathbb{N}} h(Z_n, Z_n) \qquad (*)$$

visto che il secondo membro è proprio $\bigcup_{i\in\mathbb{N}} d(Z_i)$. Per farlo dimostriamo la doppia inclusione $\subseteq, \supseteq$. L'inclusione $\supseteq$ è banale: se x appartiene a $h(Z_n, Z_n)$ per qualche n, ovviamente appartiene a $h(Z_i, Z_j)$ per qualche i, j (basta prendere

$i = j = n$). Per l'inclusione $\subseteq$, supponiamo che x appartenga a $h(Z_i, Z_j)$ per qualche i, j. Supponiamo inoltre $i \leq j$ (l'altro caso è simmetrico). Abbiamo quindi

$$\begin{aligned} & x \in h(Z_i, Z_j) \\ & = g_{Z_j}(Z_i) \\ & \subseteq g_{Z_j}(Z_i) \\ & = h(Z_j, Z_j) \end{aligned}$$

a causa della monotonia di g_{Z_j}, che è implicata dalla continuità. Quindi, prendendo $n = j$, ho che $x \in h(Z_n, Z_n)$ per qualche n.

Nota: nel caso simmetrico $i \geq j$ avrei preso $n = i$ e poi sfruttato la monotonia di f_{Z_i}.

Esercizio 5 *Sia U un insieme universo, e $f : \mathcal{P}(U) \to \mathcal{P}(U)$ una funzione continua secondo Scott.*

1) Dimostrare che la funzione

$$\begin{aligned} & g : \mathcal{P}(U) \to \mathcal{P}(U) \\ & g(X) = f(f(X)) \end{aligned}$$

è continua.

2) Dimostrare che il minimo punto fisso di g coincide con quello di f.

Suggerimento. *Per il punto 1), notate che f è monotona, siccome è continua. Quindi se $\{X_i\}_{i \in \mathbb{N}}$ è una successione crescente di insiemi, allora lo è anche $\{f(X_i)\}_{i \in \mathbb{N}}$.*

Per il punto 2), ragionate insiemisticamente dopo avere usato il teorema di Kleene. Per dimostrare che due insiemi sono uguali basta dimostrare la doppia inclusione $\subseteq, \supseteq$.

2 Esercizi sull'induzione

Esercizio 6 *Sia S l'insieme delle sequenze finite di naturali definito induttivamente da*

$$\frac{}{\epsilon}\ [S0] \qquad \frac{s}{n : s}(n \in \mathbb{N})\ [S1]$$

Si definisca induttivamente la relazione $\mathsf{len} \in \mathcal{P}(S \times \mathbb{N})$ che associa ad ogni sequenza la sua lunghezza.

Soluzione. *Una possibile soluzione è:*

$$\frac{}{\epsilon \ \mathsf{len}\ 0} \qquad \frac{s \ \mathsf{len}\ a}{n : s \ \mathsf{len}\ a + 1}$$

Esercizio 7 *Nelle stesse ipotesi del precedente esercizio, si definisca induttivamente la relazione $\mathsf{app} \in \mathcal{P}(S \times S \times S)$ tale per cui $\mathsf{app}(x, y, z)$ vale quando la sequenza z è ottenuta concatenando le sequenze x e y. Per esempio, $\mathsf{app}(1 : 2 : \epsilon, 5 : 1 : 3 : \epsilon, z)$ vale solo quando $z = 1 : 2 : 5 : 1 : 3 : \epsilon$.*

Suggerimento. *Procedete induttivamente sul primo argomento.*

Soluzione. *Una possibile soluzione:*

$$\frac{}{\mathsf{app}(\epsilon, y, y)} \qquad \frac{\mathsf{app}(x, y, z)}{\mathsf{app}(n : x, y, n : z)}$$

Esercizio 8 *Nelle stesse ipotesi del precedente esercizio, dimostrate che la concatenazione "somma le lunghezze", ovvero che:*

$$\forall x, y, z \in S, n, m \in \mathbb{N}.\ \mathsf{app}(x, y, z) \wedge x \ \mathsf{len}\ n \wedge y \ \mathsf{len}\ m \implies z \ \mathsf{len}\ (n + m)$$

Procedete per induzione su uno tra $x \in S, n \in \mathbb{N}, \mathsf{app}(x, y, z), x \ \mathsf{len}\ n$. Potete provare a fare dimostrazioni diverse usando principi di induzione diversi.

Suggerimento. *Per esempio, procedendo per induzione su $x \in S$ uno può considerare la proprietà p sulle sequenze*

$$p(x) : \qquad \forall y, z \in S, n, m \in \mathbb{N}.\ \mathsf{app}(x, y, z) \wedge x \ \mathsf{len}\ n \wedge y \ \mathsf{len}\ m \implies z \ \mathsf{len}\ (n+m)$$

Interpretando p come insieme, uno quindi verifica che $\hat{\mathcal{R}}_S(p) \subseteq p$ dove $\mathcal{R}_S = \{S0, S1\}$, ovvero che p è preservato da tutte le regole di S. Fatto questo, si ricava $S \subseteq p$, cioè che p vale su tutte le sequenze, e quindi la tesi.

Invece, procedendo per induzione su $\mathsf{app}(x, y, z)$, si prende

$$p(x, y, z) : \qquad \forall n, m \in \mathbb{N}.\ x \ \mathsf{len}\ n \wedge y \ \mathsf{len}\ m \implies z \ \mathsf{len}\ (n + m)$$

Qui p è una relazione ternaria / insieme di triple, come lo è app. Quindi, si verifica che $\hat{\mathcal{R}}_{\mathsf{app}}(p) \subseteq p$, ovvero che p è preservato da tutte le regole di app. Fatto questo, si ricava $\mathsf{app} \subseteq p$, cioè che p vale su tutte le triple si sequenze (x, y, z) su cui vale app. Da questo segue la tesi.

Esercizio 9 *(Lungo) Nelle stesse ipotesi del precedente esercizio, dimostrate che len e app sono funzioni.*

Esercizio 10 *Descrivete in modo informale ma preciso la relazione seguente. (S è definito come sopra.)*

$$\mathsf{map} \in \mathcal{P}((\mathbb{N} \to \mathbb{N}) \times S \times S)$$

$$\frac{}{\mathsf{map}(f,\ \epsilon, \epsilon)}[M0] \qquad \frac{\mathsf{map}(f,\ s,\ z)}{\mathsf{map}(f,\ n : s,\ f(n) : z)}[M1]$$

Esercizio 11 *Sia $R \in \mathcal{P}(\mathbb{Z} \times \mathbb{Z})$ la relazione definita induttivamente da*

$$\frac{}{10\ R\ 4} \qquad \frac{x\ R\ y \qquad y\ R\ 0}{3x\ R\ 2y} \qquad \frac{x\ R\ x^2}{x\ R\ 2}$$

Dimostrare che

$$\forall x, y \in \mathbb{Z}.\ x\ R\ y \implies x^2 > y$$

Suggerimento. *Usate il principio di induzione associato ad R.*

Esercizio 12 *Si considerino le definizioni di S (insieme delle sequenze di naturali) e* $\mathsf{map} \in \mathcal{P}((\mathbb{N} \to \mathbb{N}) \times S \times S)$ *viste sopra. Si definisca anche* $Q \in \mathcal{P}((\mathbb{N} \to \mathbb{N}) \times S \times S)$ *induttivamente come segue:*

$$\frac{}{Q(f, \epsilon, \epsilon)}\ [Q0] \qquad \frac{Q(f, s, z)}{Q(f, n : s, f(f(n)) : z)}\ [Q1]$$

Si dimostri la seguente proprietà:

$$R : \qquad \forall f, s_1, s_2, s_3.\ \mathsf{map}(f, s_1, s_2) \wedge \mathsf{map}(f, s_2, s_3) \implies Q(f, s_1, s_3)$$

Provate due strade diverse: 1) procedendo per induzione su $s_1 \in S$, 2) procedendo per induzione su $\mathsf{map}(f, s_1, s_2)$. *In entrambi i casi, partite definendo la proprietà T che usate per l'induzione, e verificate che quello che si ottiene per induzione effettivamente implica R.*

Suggerimento. *Per la strada 1), una proprietà possibile è*

$$T(s) : \qquad \forall f, s_2, s_3.\ \mathsf{map}(f, s, s_2) \wedge \mathsf{map}(f, s_2, s_3) \implies Q(f, s_1, s_3)$$

In altri termini, T è una proprietà unaria, ovvero un insieme. Se le ipotesi del principio di induzione sono vere ($\hat{\mathcal{R}}_S(T) \subseteq T$ dove $\mathcal{R}_S = \{S0, S1\}$), dall'induzione si ricava che $S \subseteq T$, cioè che T è vero su tutte le sequenze, da cui R.

Per la strada 2), una proprietà possibile è

$$T(f, s, z) : \qquad \forall s_3.\ \mathsf{map}(f, z, s_3) \implies Q(f, s, s_3)$$

In altri termini, T è una relazione / proprietà ternaria, dello stesso tipo di map. *Se le ipotesi del principio di induzione sono vere ($\hat{\mathcal{R}}_{\mathsf{map}}(T) \subseteq T$ dove $\mathcal{R}_{\mathsf{map}} = \{M0, M1\}$), dall'induzione si ricava che $\mathsf{map} \subseteq T$, cioè che T è vero su tutte le triple (f, s, z) che soddisfano* map. *Da questo si ricava R.*

Esercizio 13 *Riusando le stesse definizioni dell'esercizio di sopra, si dimostri la seguente proprietà:*

$$R : \qquad \forall f, g, s_1, s_2, s_3.\ \mathsf{map}(f, s_1, s_2) \wedge \mathsf{map}(g, s_2, s_3) \implies \mathsf{map}(g \circ f, s_1, s_3)$$

dove $g \circ f$ denota la composizione ($(g \circ f)(x) = g(f(x))$). Provate due strade diverse: 1) procedendo per induzione su $s_1 \in S$, 2) procedendo per induzione su $\mathsf{map}(f, s_1, s_2)$. *In entrambi i casi, partite definendo la proprietà T che usate per l'induzione, e verificate che quello che si ottiene per induzione effettivamente implica R.*

Suggerimento. *Per la strada 1), una proprietà possibile è*

$$T(s) : \qquad \forall f, g, s_2, s_3.\ \mathsf{map}(f, s, s_2) \wedge \mathsf{map}(g, s_2, s_3) \implies \mathsf{map}(g \circ f, s, s_3)$$

Qui, T è una proprietà unaria, ovvero un insieme.

Per la strada 2), una proprietà possibile è

$$T(f, s, z) : \qquad \forall g, s_3.\ \mathsf{map}(g, z, s_3) \implies \mathsf{map}(g \circ f, s, s_3)$$

Qui, T è una proprietà ternaria, dello stesso tipo di map.

Esercizio 14 (Impegnativo) *Il principio di induzione afferma che, quando $f \in \mathcal{P}(U) \to \mathcal{P}(U)$ è monotona,*

$$f(Y) \subseteq Y \implies \mathsf{fix}(f) \subseteq Y$$

Di tale principio esiste anche la seguente variante "forte":

$$f(\mathsf{fix}(f) \cap Y) \subseteq Y \implies \mathsf{fix}(f) \subseteq Y$$

Questa variante, usa un'ipotesi induttiva più forte, che non afferma più soltanto l'appartenenza a Y, ma anche a $\mathsf{fix}(f)$. Per esempio, procedendo per induzione "debole" sui naturali e dimostrando il caso induttivo, si deve fare vedere che $p(n) \implies p(n+1)$ dove n è un valore qualunque (un arbitrario elemento dell'universo U sul quale abbiamo dato le regole di inferenza) e non per forza un naturale (!). Il principio "forte" consente invece di assumere anche $n \in \mathbb{N}$ tra le ipotesi induttive, e quindi semplificando la dimostrazione del caso induttivo.

Si dimostri il principio forte, sfruttando quello debole.

Soluzione. *Assumiamo $IP1: \ f(\mathsf{fix}(f) \cap Y) \subseteq Y$ e dimostriamo la tesi $\mathsf{fix}(f) \subseteq Y$.*

Per la monotonia di f e la definizione di fix*, abbiamo che*

$$IP2: \qquad f(\mathsf{fix}(f) \cap Y) \subseteq f(\mathsf{fix}(f)) \subseteq \mathsf{fix}(f)$$

Da $IP1, IP2$ otteniamo quindi

$$f(\mathsf{fix}(f) \cap Y) \subseteq \mathsf{fix}(f) \cap Y$$

A quanto sopra si può applicare il principio debole di induzione (scegliendo $\mathsf{fix}(f) \cap Y$ come insieme Y) dal quale otteniamo

$$\mathsf{fix}(f) \subseteq \mathsf{fix}(f) \cap Y$$

da cui la tesi $\mathsf{fix}(f) \subseteq Y$.

3 Esercizi sulla semantica

Esercizio 15 *Si estenda IMP con il comando*

$$\mathsf{do}\ c\ \mathsf{while}\ e \neq 0$$

dove c è un comando ed e è un'espressione intera. La semantica intuitiva del comando aggiunto è la seguente. All'inizio si esegue c; poi si valuta e. Se e vale zero, si esce dal ciclo. Altrimenti si ripete il tutto: si riesegue c, si rivaluta e, etc.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Soluzione. *Basta aggiungere le regole:*

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

$$\frac{\langle c,\sigma\rangle \to_b \sigma' \qquad \langle e,\sigma'\rangle \to_e 0}{\langle \mathsf{do}\ c\ \mathsf{while}\ e\neq 0,\sigma\rangle \to_b \sigma'}[doWhile-false]$$

Esercizio 16 *Si estenda IMP con il comando*

$$\mathsf{save}\ x\ \mathsf{in}\ c$$

(dove x è una variabile e c un comando) con la seguente semantica intuitiva. Si annota il valore di x, poi si esegue c, e dopo si ripristina il vecchio valore di x precedentemente annotato.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Soluzione.

$$\frac{\langle c,\sigma\rangle \to_b \sigma'}{\langle \mathsf{save}\ x\ \mathsf{in}\ c,\sigma\rangle \to_b \sigma'[x\mapsto \sigma(x)]}[Save]$$

Esercizio 17 (Impegnativo) *Si estenda IMP con il comando*

$$\mathsf{while}\ e\neq 0\ \mathsf{alternate}\ c_1\ \mathsf{with}\ c_2$$

(dove e è un'espressione intera e c_1, c_2 sono comandi) con la seguente semantica intuitiva. Si verifica la condizione $e \neq 0$. Se non vale si esce dal ciclo. Se vale si esegue c_1 e si continua come segue. Si verifica la condizione $e \neq 0$. Se non vale si esce dal ciclo. Se vale si esegue c_2 e si ripete il tutto da capo, alternando c_1 e c_2 ad ogni iterazione.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Soluzione.

$$\frac{\langle e,\sigma\rangle \to_e v\neq 0 \qquad \langle c_1;\mathsf{while}\ e\neq 0\ \mathsf{alternate}\ c_2\ \mathsf{with}\ c_1,\sigma\rangle \to_b \sigma'}{\langle \mathsf{while}\ e\neq 0\ \mathsf{alternate}\ c_1\ \mathsf{with}\ c_2,\sigma\rangle \to_b \sigma'}[Altern{-}true]$$

$$\frac{\langle e,\sigma\rangle \to_e 0}{\langle \mathsf{while}\ e\neq 0\ \mathsf{alternate}\ c_1\ \mathsf{with}\ c_2,\sigma\rangle \to_b \sigma}[Altern-false]$$

Esercizio 18 *Dopo avere letto l'esercizio precedente, dimostrate in modo informale che il comando*

$$\mathsf{while}\ e\neq 0\ \mathsf{alternate}\ c_1\ \mathsf{with}\ c_2$$

è equivalente a

```
while e ≠ 0 do
    c1;
    if e ≠ 0 then c2 else skip
```

Esercizio 19 (Impegnativo) *Si rimuovano dalla definizione della semantica big step di IMP le due regole relative al comando* if*, e si aggiunga alle regole rimaste la regola seguente:*

$$\frac{\langle e,\sigma\rangle \to_e v \qquad \langle c_1,\sigma\rangle \to_b \sigma_1 \qquad \langle c_2,\sigma\rangle \to_b \sigma_2}{\langle \mathsf{if}\ e\neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2,\sigma\rangle \to_b \mathbf{cond}(v,\sigma_1,\sigma_2)}[If]$$

dove l'operatore **cond** *è definito come segue*

$$\mathbf{cond} : \mathbb{Z} \times Store \times Store \to Store$$
$$\mathbf{cond}(v, \sigma_1, \sigma_2) = \begin{cases} \sigma_1 & se\ v \neq 0 \\ \sigma_2 & se\ v = 0 \end{cases}$$

Dire se dopo questa modifica alle regole, la semantica rimane equivalente alla semantica originale di IMP. Se si, giustificarlo in modo informale ma preciso. Se no, trovare su quale comando la nuova semantica differisce e spiegarne il motivo.

Suggerimento. *La nuova semantica non è equivalente a quella originale.*

Soluzione. *Prendete* $w = \mathsf{while}\ 1 \neq 0\ \mathsf{do\ skip}$. *Per nessun* σ, σ' *ho che* $\langle w, \sigma \rangle \to_b \sigma'$ *e questo vale sia con la vecchia che con la nuova semantica, visto che le regole per il while sono invariate.*

Considerate quindi il comando c *definito come segue:*

$$\mathsf{if}\ 1 \neq 0\ \mathsf{then\ skip\ else}\ w$$

Con la vecchia semantica ho che $\langle c, \sigma \rangle \to_b \sigma$ *per ogni* σ. *Nella nuova semantica invece ho che* $\langle c, \sigma \rangle \to_b \sigma'$ *per NESSUN* σ, σ'. *Infatti la nuova regola per l'*if *mi richiede di trovare due stati finali* σ_1, σ_2 *per ambo i rami, ma il ramo* w *non ammette stato finale. Quindi la nuova semantica è diversa da quella originale.*

Esercizio 20 *Si estenda IMP con il comando*

$$\mathsf{swap}(x, y)$$

(dove x, y *sono variabili) con la seguente semantica intuitiva. I valori delle due variabili vengono scambiati.*

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Esercizio 21 *Si estenda IMP con il comando*

$$\mathsf{repeat}\ e\ \mathsf{times}\ c$$

(dove e *è un'espressione intera e* c *è un comando) con la seguente semantica intuitiva. All'inizio viene valutata* e*: sia* z *il suo valore. Se* $z \leq 0$, *il comando* repeat *non ha nessun effetto. Altrimenti, il comando* repeat *esegue il comando* c *esattamente* z *volte.*

Per esempio,

$$\begin{array}{l} x := 0; \\ y := 5; \\ \mathsf{repeat}\ y\ \mathsf{times} \\ \qquad x := x + 1; \\ \qquad y := y + 1 \end{array}$$

termina in uno stato in cui x *vale* 5 *e* y *vale* 10.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Esercizio 22 *Si estenda IMP con il comando*

$$\text{loop } c_1 \text{ exit if } e = 0 \text{ otherwise } c_2$$

(dove e è un'espressione intera e c_1, c_2 sono comandi) con la seguente semantica intuitiva. All'inizio viene eseguito c_1. Poi viene valutata e: se il risultato è nullo, il comando loop *termina qui. Altrimenti, viene eseguito c_2 e poi si ripete tutto il* loop *da capo.*

Per esempio,

$$\begin{array}{l} x := 0; \\ y := 5; \\ \text{loop} \\ \quad y := y - 1 \\ \quad \text{exit if } y = 0 \text{ otherwise} \\ \quad x := x + 1 \end{array}$$

termina in uno stato in cui x vale 4 e y vale 0.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Soluzione.

$$\frac{\langle c_1 \ , \ \sigma \rangle \to_b \sigma' \qquad \langle e \ , \ \sigma' \rangle \to_e 0}{\langle \text{loop } c_1 \text{ exit if } e = 0 \text{ otherwise } c_2 \ , \ \sigma \rangle \to_b \sigma'} [Loop - True]$$

$$\frac{\langle c_1 \ , \ \sigma \rangle \to_b \sigma' \qquad \langle e \ , \ \sigma' \rangle \to_e v \neq 0 \qquad \langle c_2; \text{loop } c_1 \text{ exit if } e = 0 \text{ otherwise } c_2 \ , \ \sigma' \rangle \to_b \sigma''}{\langle \text{loop } c_1 \text{ exit if } e = 0 \text{ otherwise } c_2 \ , \ \sigma \rangle \to_b \sigma''} [Loop - False]$$

Esercizio 23 *Si estenda IMP con il comando*

$$\text{take } x \text{ in } c$$

(dove x è una variabile e c è un comando) con la seguente semantica intuitiva. All'inizio viene eseguito c. Poi, il valore di tutte le variabili viene ripristinato al valore precedente all'esecuzione di c, tranne per x che invece preserva il suo valore successivo all'esecuzione di c.

Aggiungere una o più regole alla semantica big step per formalizzare questo comportamento.

Soluzione.

$$\frac{\langle c \ , \ \sigma \rangle \to_b \sigma'}{\langle \text{take } x \text{ in } c \ , \ \sigma \rangle \to_b \sigma[x \mapsto \sigma'(x)]}$$

Esercizio 24 *Sia x una variabile fissata di IMP. Si consideri il seguente predicato sui comandi di IMP, definito ricorsivamente.*

$$\frac{}{p(x := 0)} [P0] \qquad \frac{}{p(x := 2 \cdot e)} [P1]$$

$$\frac{p(c_2)}{p(c_1; c_2)} [P2] \qquad \frac{p(c_1) \qquad p(c_2)}{p(\text{if } e \neq 0 \text{ then } c_1 \text{ else } c_2)} [P3]$$

(Sopra, x è fissata, mentre e, c_1, c_2 sono arbitrari)

Dimostrare, sfruttando il principio di induzione associato a p, che

$$p(c) \wedge \langle c, \sigma \rangle \to_b \sigma' \implies \sigma'(x) \textit{ pari}$$

4 Esercizi sulle equivalenze

Esercizio 25 *Dimostrate che il comando*

$$\mathsf{while}\ e \neq 0\ \mathsf{do}\ c$$

è equivalente a

$$\begin{array}{l} \mathsf{while}\ e \neq 0\ \mathsf{do} \\ \qquad \mathsf{if}\ e \neq 0\ \mathsf{then}\ c\ \mathsf{else\ skip} \end{array}$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi di sopra.

Esercizio 26 *Dimostrate che il comando*

$$\mathsf{if}\ e_1 \cdot e_2 \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

è equivalente a

$$\begin{array}{l} \mathsf{if}\ e_1 \neq 0\ \mathsf{then} \\ \qquad \mathsf{if}\ e_2 \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2 \\ \mathsf{else} \\ \qquad c_2 \end{array}$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi e/o espressioni di sopra.

Esercizio 27 *Dimostrate che il comando*

$$\mathsf{if}\ x \neq 0\ \mathsf{then}\ x := 1\ \mathsf{else}\ x := x + 1$$

è equivalente a

$$x := 1$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi e/o espressioni di sopra.

Esercizio 28 *Dimostrate che il comando*

$$x := x + 1; \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

è equivalente a

$$\mathsf{if}\ x + 1 \neq 0\ \mathsf{then}\ x := x + 1; c_1\ \mathsf{else}\ x := x + 1; c_2$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi e/o espressioni di sopra.

Esercizio 29 *Dimostrate che il comando*

$$x := 5\ ;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

è equivalente a

$$x := 5\ ;\ c_1$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi e/o espressioni di sopra.

Esercizio 30 *Dimostrate che il comando*

$$x := 5\ ;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

è equivalente a

$$x := 5\ ;\ c_1$$

Nella dimostrazione, fate riferimento alle relative derivazioni per i comandi e/o espressioni di sopra.

Esercizio 31 *Considerate il comando di IMP (esteso con $\wedge$)*

$$\mathsf{if}\ \phi_1 \wedge \phi_2\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

dove ϕ_1, ϕ_2 sono espressioni booleane.

Riscrivete il comando sopra in modo equivalente senza usare l'operatore $\wedge$. (Potete assumere che dentro ϕ_1, ϕ_2 non ci siano altri $\wedge$.)

Infine, ripetete l'esercizio sopra nella sua naturale variante per

$$\mathsf{if}\ \phi_1 \vee \phi_2\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

e per

$$\mathsf{if}\ \neg\phi_1\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2$$

5 Esercizi sulla correttezza dei programmi

Esercizio 32 *Si dimostri la seguente tripla di Hoare*

```
{n ≥ 0}
s := 0;
x := 0;
while x ≤ n do
      if f(x) ≠ 0 then
            s := s + f(x)
      else
            skip;
      x := x + 1
{s = Σ_{X=0}^{n} f(X)}
```

Come si può semplificare il programma di sopra in modo equivalente?

Esercizio 33 *Si dimostri la seguente tripla di Hoare*

$\{n \geq 0\}$
$s := 0;$
$x := 0;$
while $x \leq n$ do
    if $f(x) \geq 0$ then
        $s := s + f(x)$
    else
        $s := s - f(x);$
    $x := x + 1$
$\{s = \sum_{X=0}^{n} \mid f(X) \mid\}$

Soluzione. *(Soluzione completa) Ho la seguente derivazione per la tripla di Hoare:*

$\{n \geq 0\}$
$\{n \geq 0 \wedge 0 \leq 0 \leq n+1 \wedge 0 = \sum_{X=0}^{0-1} \mid f(X) \mid\}$ (1)
$s := 0;$
$\{n \geq 0 \wedge 0 \leq 0 \leq n+1 \wedge s = \sum_{X=0}^{0-1} \mid f(X) \mid\}$
$x := 0;$
$\{INV: \ n \geq 0 \wedge 0 \leq x \leq n+1 \wedge s = \sum_{X=0}^{x-1} \mid f(X) \mid\}$
while $x \leq n$ do
    $\{INV \wedge x \leq n\}$
    if $f(x) \geq 0$ then
        $\{INV \wedge x \leq n \wedge f(x) \geq 0\}$
        $\{n \geq 0 \wedge 0 \leq x \leq n \wedge s + f(x) = \sum_{X=0}^{x} \mid f(X) \mid\}$ (2)
        $s := s + f(x)$
    else
        $\{INV \wedge x \leq n \wedge \neg(f(x) \geq 0)\}$
        $\{n \geq 0 \wedge 0 \leq x \leq n \wedge s - f(x) = \sum_{X=0}^{x} \mid f(X) \mid\}$ (3)
        $s := s - f(x);$
    $\{n \geq 0 \wedge 0 \leq x \leq n \wedge s = \sum_{X=0}^{x} \mid f(X) \mid\}$
    $\{n \geq 0 \wedge 0 \leq x+1 \leq n+1 \wedge s = \sum_{X=0}^{(x+1)-1} \mid f(X) \mid\}$ (4)
    $x := x + 1$
    $\{INV\}$
$\{INV \wedge \neg(x \leq n)\}$
$\{s = \sum_{X=0}^{n} \mid f(X) \mid\}$ (5)

Le applicazioni della regola PrePost sono giustificate come segue. (1) Si sta sommando su un intervallo vuoto, quindi la somma vale zero. (2) L'invariante mi dice che s somma fino a $x-1$, e siccome $f(x) \geq 0$ ho che $f(x) = \mid f(x) \mid$ e quindi per ottenere la somma fino a x mi basta aggiungere $f(x)$. (3) L'invariante mi dice che s somma fino a $x-1$, e siccome $f(x) < 0$ ho che $-f(x) = \mid f(x) \mid$ e quindi per ottenere la somma fino a x mi basta sottrarre $f(x)$. (4) Semplice aritmetica. (5) Ho che $x \leq n+1$ e che $\neg(x \leq n)$ quindi $x = n+1$.

Esercizio 34 *Sotto, indichiamo con* $fib(n)$ *l'ennesimo numero di Fibonacci:* $fib(0) = 0, fib(1) = 1, fib(n+2) = fib(n) + fib(n+1)$. *Si dimostri la seguente tripla di Hoare*

$\{n = N \geq 0\}$
$a := 0;$
$b := 1;$
while $n > 0$ do
$\quad b := a + b;$
$\quad a := b - a;$
$\quad n := n - 1$
$\{a = fib(N)\}$

Soluzione. *(Soluzione completa)*

$\{n = N \geq 0\}$
$\{n \geq 0 \wedge 0 = fib(N - n) \wedge 1 = fib(N - n + 1)\}(1)$
$a := 0;$
$\{n \geq 0 \wedge a = fib(N - n) \wedge 1 = fib(N - n + 1)\}$
$b := 1;$
$\{INV: \ n \geq 0 \wedge a = fib(N - n) \wedge b = fib(N - n + 1)\}$
while $n > 0$ do
$\quad \{INV \wedge n > 0\}$
$\quad \{n - 1 \geq 0 \wedge a + b - a = fib(N - n + 1) \wedge a + b = fib(N - n + 2)\}(2)$
$\quad b := a + b;$
$\quad \{n - 1 \geq 0 \wedge b - a = fib(N - n + 1) \wedge b = fib(N - n + 2)\}$
$\quad a := b - a;$
$\quad \{n - 1 \geq 0 \wedge a = fib(N - n + 1) \wedge b = fib(N - n + 2)\}$
$\quad n := n - 1;$
$\quad \{INV\}$
$\{INV \wedge \neg(n > 0)\}(3)$
$\{a = fib(N)\}$

Le applicazioni della regola PrePost sono giustificate come segue.

1) Semplice aritmetica e definizione di fib*:* $fib(N - N) = 0$ *e* $fib(N - N + 1) = 1$.

2) Da $n > 0$ *ho che* $n - 1 \geq 0$. *Inoltre, dall'invariante* $b = fib(N - n + 1)$ *è immediato che* $a + b - a = fib(N - n + 1)$. *Infine dall'invariante* $a = fib(N - n) \wedge b = fib(N - n + 1)$ *si ha* $a + b = fib(N - n) + fib(N - n + 1) = fib(N - n + 2)$.

3) Dall'invariante $n \geq 0$ *e* $\neg(n > 0)$ *ho che* $n = 0$, *quindi da* $a = fib(N - n)$ *concludo* $a = fib(N)$.

Esercizio 35 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{n = N \geq 0\} \\
x := 0; \\
y := 1; \\
\textsf{while } x < n \textsf{ do} \\
\qquad y := 0 - y; \\
\qquad x := x + 1 \\
\{y = (-1)^N\}
\end{array}
$$

Esercizio 36 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{n = N \geq 0\} \\
x := 0; \\
y := 1; \\
\textsf{while } x < n \textsf{ do} \\
\qquad \textsf{if } y = 1 \textsf{ then} \\
\qquad\qquad y := -1 \\
\qquad \textsf{else} \\
\qquad\qquad y := 1; \\
\qquad x := x + 1 \\
\{y = (-1)^N\}
\end{array}
$$

Esercizio 37 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{n = N \geq 0 \wedge a = A\} \\
\textsf{while } n > 0 \textsf{ do} \\
\qquad a := a * a; \\
\qquad n := n - 1 \\
\{a = A^{(2^N)}\}
\end{array}
$$

Esercizio 38 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{a \leq b\} \\
\textsf{while } a < b \textsf{ do} \\
\qquad \textsf{if } b - a > 5 \textsf{ then} \\
\qquad\qquad x := 5 \\
\qquad \textsf{else} \\
\qquad\qquad x := 1; \\
\qquad a := a + x \\
\{a = b\}
\end{array}
$$

Esercizio 39 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{x = X \geq 0 \wedge y = q = 0\} \\
\textsf{while } x > 0 \textsf{ do} \\
\quad x := x - 1; \\
\quad y := y + 1; \\
\quad \textsf{if } y = 5 \textsf{ then} \\
\quad\quad y := 0; \\
\quad\quad q := q + 1 \\
\quad \textsf{else} \\
\quad\quad \textsf{skip} \\
\{q = \lfloor X/5 \rfloor\}
\end{array}
$$

Esercizio 40 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{0 \leq x = X \leq y = Y\} \\
z := y - x; \\
\textsf{while } z > 0 \textsf{ do} \\
\quad x := x + 1; \\
\quad y := y - 1; \\
\quad z := z - 1; \\
\{x = Y \wedge y = X\}
\end{array}
$$

Esercizio 41 *Si dimostri la validità della tripla di Hoare seguente:*

$$
\begin{array}{l}
\{n = N \geq 0\} \\
s := 0; \\
x := 1; \\
\textsf{while } n > 0 \textsf{ do} \\
\quad s := s + x; \\
\quad x := 2 * x; \\
\quad n := n - 1; \\
\{s = 2^N - 1\}
\end{array}
$$

6 Esercizi sulle triple di Hoare

Esercizio 42 *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P\}\ c\ \{Q\}}{\{P\}\ \textsf{while } \phi \textsf{ do } c\ \{Q\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto. In altri termini, stabilire se continua a valere

$$\vdash \{P\}\ c\ \{Q\} \implies \models \{P\}\ c\ \{Q\}$$

per ogni comando c.

Suggerimento. *Il sistema deduttivo diventa non corretto.*

Esercizio 43 *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P\}\ c\ \{P\}}{\{P\}\ \mathsf{while}\ \phi\ \mathsf{do}\ c\ \{P\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto.

Suggerimento. *Il sistema deduttivo rimane corretto.*

Soluzione. *Per dimostrare la correttezza del sistema deduttivo "con la regola in più" basta fare vedere che in questo sistema si deducono solo triple di Hoare che si possono dedurre anche nel sistema originale. Siccome in sistema originale è corretto, questo implica che anche il nuovo sistema lo è.*

Per dimostrarlo, basta notare che se esiste una derivazione nel nuovo sistema che usa la regola di sopra, la si può trasformare in una derivazione che usa solo le regole originali nel seguente modo. Basta riscrivere una derivazione

$$\cfrac{\cfrac{\cfrac{\vdots}{\{P\}\ c\ \{P\}}}{\{P\}\ \mathsf{while}\ \phi\ \mathsf{do}\ c\ \{P\}}}{\vdots}$$

così:

$$\cfrac{P \implies P \qquad \cfrac{\cfrac{P \wedge \phi \implies P \qquad \cfrac{\vdots}{\{P\}\ c\ \{P\}} \qquad P \implies P}{\{P \wedge \phi\}\ c\ \{P\}}[PrePost]}{\{P\}\ \mathsf{while}\ \phi\ \mathsf{do}\ c\ \{P \wedge \neg\phi\}}[While] \qquad P \wedge \neg\phi \implies P}{\cfrac{\{P\}\ \mathsf{while}\ \phi\ \mathsf{do}\ c\ \{P\}}{\vdots}}[PrePost]$$

Esercizio 44 *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P\}\ c\ \{Q\}}{\{P\}\ \mathsf{if}\ \phi\ \mathsf{then}\ c\ \mathsf{else}\ c\ \{Q\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto.

Suggerimento. *Il sistema deduttivo rimane corretto.*

Esercizio 45 *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P\}\ x := e_2\ \{Q\}}{\{P\}\ x := e_1;\ x := e_2\ \{Q\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto.

Suggerimento. *Il sistema deduttivo diventa non corretto.*

Esercizio 46 (Impegnativo) *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P\{5/x\} \land x = 0\}\ c_2\ \{Q\}}{\{P\{5/x\}\}\ x := 0;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \{Q\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto.

Suggerimento. *Notate che col sistema deduttivo non esteso si può dimostrare la tripla*

$$\{\mathsf{falso}\}\ \mathsf{c}\ \{\mathsf{Q}\}$$

per ogni c, Q. Questo si può verificare per induzione sulla sintassi di c.

Dopo avere notate questo, costruite una derivazione per

$$\{P\{5/x\}\}\ x := 0;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \{Q\}$$

supponendo di avere a disposizione una derivazione per

$$\{P\{5/x\} \land x = 0\}\ c_2\ \{Q\}$$

Potendo fare questo, se ne deriva che ogni applicazione della regola aggiunta in realtà è sostituibile con una o più regole originali, e che quindi le triple derivate con la regola aggiunta sono derivabili anche senza tale regola. In conclusione, il sistema deduttivo delle tiple di Hoare rimane corretto.

Esercizio 47 *Si aggiunga al sistema deduttivo per le triple di Hoare la regola*

$$\frac{\{P \land x = 0\}\ c_2\ \{Q\}}{\{P\}\ x := 0;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \{Q\}}$$

Stabilire se con la regola aggiunta il sistema deduttivo continua ad essere corretto.

Suggerimento. *Diventa non corretto.*

Soluzione. *(Bozza) Prendete $P =$ "$x = 1$", $Q =$ "$falso$", $c_1 = c_2 = \mathsf{skip}$. A questo punto riuscirete a dimostrare $\{P \land x = 0\}\ c_2\ \{Q\}$ ma la tripla*

$$\{P\}\ x := 0;\ \mathsf{if}\ x \neq 0\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \{Q\}$$

è palesemente scorretta: se parto da un σ tale che soddisfa P cioè $\sigma(x) = 1$, termino in $\sigma[x \mapsto 0]$, che banalmente NON può soddisfare Q.

7 Esercizi di programmazione

Esercizio 48 *Sia $f : \mathbb{Z} \to \mathbb{Z}$ una funzione arbitraria, e aggiungete $f(Exp)$ alle espressioni di IMP. Si scriva un programma c tale per cui*

$$\begin{array}{l} \{a = A \land b = B > A\} \\ c \\ \{x = f(A) + 2 \cdot f(A+1) + 3 \cdot f(A+2) + \cdots + (B - A + 1) \cdot f(B)\} \end{array}$$

Si giustifichi informalmente la correttezza della soluzione proposta, e se ne stimi la complessità asintotica.

Esercizio 49 *Sia $f : \mathbb{Z} \to \mathbb{Z}$ una funzione arbitraria, e aggiungete $f(Exp)$ alle espressioni di IMP. Si scriva un programma c che, dati A e B con $A < B$, conti quanti minimi locali di f ci sono nell'intervallo $[A, B]$. Un minimo locale è un $f(X)$ tale per cui $f(X) \leq f(X-1)$ e $f(X) \leq f(X+1)$. Si stimi infine la complessità asintotica della soluzione proposta.*

Esercizio 50 *Si consideri il linguaggio IMP esteso con i vettori (array). Si scriva un programma in tale linguaggio tale che, dato un vettore in una variabile a, lo modifichi invertendo l'ordine delle componenti. Per esempio, un vettore $\langle 1, 2, 3\rangle$ deve trasformarsi in $\langle 3, 2, 1\rangle$. Si usi l'espressione a.length per indicare la lunghezza del vettore. Si giustifichi informalmente la correttezza della soluzione proposta, e se ne stimi la complessità asintotica.*

Esercizio 51 *Si consideri il linguaggio IMP esteso con i vettori (array). Si scriva un programma in tale linguaggio tale che, dato un vettore in una variabile a, lo modifichi incrementando ogni componente maggiore di 5. Per esempio, un vettore $\langle 1, 20, 3\rangle$ deve trasformarsi in $\langle 1, 21, 3\rangle$. Si usi l'espressione a.length per indicare la lunghezza del vettore. Si giustifichi informalmente la correttezza della soluzione proposta, e se ne stimi la complessità asintotica.*

Esercizio 52 *Si consideri il linguaggio IMP esteso con i vettori (array). Si scriva un programma in tale linguaggio tale che, dato un vettore di interi $\langle a_1, a_2, a_3, \ldots, a_{n-1}, a_n\rangle$ in una variabile a, lo modifichi facendolo diventare $\langle 0, a_1, a_2, a_3, \ldots, a_{n-1}\rangle$. Per esempio, un vettore $\langle 2, 3, 4, 5\rangle$ deve trasformarsi in $\langle 0, 2, 3, 4\rangle$. Si usi l'espressione a.length per indicare la lunghezza del vettore. Si giustifichi informalmente la correttezza della soluzione proposta, e se ne stimi la complessità asintotica.*

Esercizio 53 *Si costruisca un algoritmo per calcolare $\lfloor\sqrt{x}\rfloor$ dato un intero x positivo. Nel farlo, si usino solo le operazioni aritmetiche elementari (non la radice quadrata). Si miri a ottenere un algoritmo efficiente, considerandone la complessità asintotica.*

Suggerimento. *Si può fare in $O(\log x)$. Sfruttate il fatto che la funzione $f(x) = x^2$ è monotona.*

Soluzione. *(Bozza) Basta usare la ricerca binaria su $f(x) = x^2$, e cercare un $0 \leq y \leq x$ per cui $f(y) \leq x$ e $f(y+1) > x$.*

Esercizio 54 *Si considerino i seguenti due algoritmi per eliminare i duplicati da un vettore $\langle a_0, \ldots, a_{n-1}\rangle$, producendo un altro vettore (più corto, in generale) $\langle b_0, \ldots, b_{m-1}\rangle$,*

Algoritmo 1) Scandisco tutte le posizioni di a con $0 \leq i < n$. Per ogni a_i, controllo se esiste un $j \in [0, i)$ tale che $a_j = a_i$. Se esiste, non faccio nulla, altrimenti lo aggiungo al vettore b.

Algoritmo 2) Inizio ordinando il vettore a. Poi ne scandisco le posizioni con $0 \leq i < n$. Per ogni a_i, controllo se $i > 0$ e $a_{i-1} = a_i$. In tal caso, non faccio nulla, altrimenti aggiungo a_i al vettore b.

Sopra, "aggiungere al vettore b" vuole semplicemente indicare l'operazione $b[k] := a[i]; k := k + 1$ dove k è una variabile inizialmente posta a 0.

Si stimi la complessità asintotica di entrambi gli algoritmi.

Soluzione. *(Bozza) L'algoritmo 1 ha una complessità di* $O(n^2)$*, mentre l'algoritmo 2 ha una complessità di* $O(n \cdot \log n)$*.*

Esercizio 55 *Siano dati due vettori* $\langle a_0, \ldots, a_{n-1} \rangle$ *e* $\langle b_0, \ldots, b_{n-1} \rangle$ *contenenti i coefficienti di due polinomi*

$$p(x) = a_0 + a_1 \cdot x + \cdots + a_{n-1} \cdot x^{n-1}$$
$$q(x) = b_0 + b_1 \cdot x + \cdots + b_{n-1} \cdot x^{n-1}$$

Scrivere un programma in IMP, esteso con i vettori (array), che produce in un vettore c (di lunghezza $2 \cdot n - 1$*) i coefficienti del polinomio* $r(x) = p(x) \cdot q(x)$*.*

Si giustifichi informalmente la correttezza della soluzione proposta, e se ne stimi la complessità asintotica.

Suggerimento. *Si noti che*

$$c_k = \sum_{i+j=k} a_i \cdot b_j$$