Simone Martini
Università di Udine

Anno Accademico 1995/96
September 17, 1996

Esercizi e complementi di Metodi per il Trattamento dell'Informazione[1]

Tutte le discipline a carattere matematico e sperimentale hanno bisogno di un congruo numero di esercizi per poter essere appieno comprese e padroneggiate. Gli argomenti che costituiscono il programma del Corso di Metodi per il Trattamento dell'Informazione non fanno eccezione, con il problema che non è facile trovare raccolte di esercizi adeguate alla preparazione dell'esame. Queste pagine cercano di essere un primo contributo in tal senso, raccogliendo esercizi di diversa difficoltà, quasi tutti corredati da estese soluzioni. La raccolta comprende tutte le prove d'esame dell'A.A. 1994/95 (vedi il paragrafo 4). In aggiunta a questi esercizi di formato "classico", si sono inseriti anche alcuni brevi argomenti, non compresi nei manuali adottati per il corso, scelti per il loro interesse o per la loro utilità per gli esercizi.

Come tutte le prime versioni, anche queste note conterranno molti refusi e, forse, qualche errore. Per cercare di eliminarli dalle versioni successive metto in palio una *taglia*: il primo studente a segnalare un refuso (o un errore), con relativa correzione, riceverà un quarto di punto di incremento sulla votazione del suo scritto sufficiente.

Ringrazio Fabio Alessi per aver contribuito esercizi e soluzioni per la parte di linguaggi formali.

1 Teoria della Calcolabilità

1.1 Notazioni

$\mathcal{C}$ sono le funzioni ricorsive unarie; φ è un'enumerazione canonica di $\mathcal{C}$; Ω è la funzione ovunque indefinita; $|A|$ è la cardinalità dell'insieme A; π_0 (π_1) è la prima (seconda) proiezione sulle coppie codificate.

1.2 Insiemi ricorsivi e r.e.

1.2.1. Dato un insieme r.e. R, esiste g totale ricorsiva tale che, per ogni x,

$$W_{g(x)} = \begin{cases} N & \text{se } x \in R \\ \emptyset & \text{se } x \notin R. \end{cases}$$

Sol. La richiesta g manda x nell'indice del seguente programma:

```
begin
input y;
esegui x ∈? R;
if (x ∈? R)↓ then output y
end
```

Si tratta ovviamente di una funzione calcolabile e totale coi richiesti requisiti.

Più formalmente, la funzione g si ottiene tramite il teorema s-m-n. Definiamo la funzione

$$f(x,y) \simeq \begin{cases} y & \text{se } x \in R \\ \uparrow & \text{altrimenti.} \end{cases}$$

ed applichiamo s-m-n per ottenere una funzione totale e calcolabile g tale che $\varphi_{g(x)}(y) \simeq f(x,y)$.

[1]Note ad uso esclusivo degli studenti.

Si osservi che *non* esiste una funzione ricorsiva h tale che, per ogni x,

$$h(x) = \begin{cases} \text{un indice di } N & \text{se } x \in R \\ \text{un indice di } \emptyset & \text{se } x \notin R. \end{cases}$$

1.2.2. (Generalizzazione del precedente). Sia R un insieme r.e. e siano $W_i \subseteq W_j$. Dimostrare che esiste g totale ricorsiva tale che, per ogni x,

$$W_{g(x)} = \begin{cases} W_j & \text{se } x \in R \\ W_i & \text{se } x \notin R. \end{cases}$$

Sol. La funzione g manda x nell'indice del seguente programma:

```
begin
input y;
esegui in parallelo x ∈? R and φ_i(y);
{  if (x ∈? R)↓ then if φ_j(y)↓ then output 0
   if φ_i(y)↓ then output 0
}
end.
```

1.2.3. Esiste una funzione calcolabile g tale che

$$W_{g(i,m)} = \begin{cases} E \subseteq W_i,\ E \text{ finito} & \text{se } m \in K \\ W_i & \text{se } m \notin K. \end{cases}$$

Sol. La funzione g manda i ed m nell'indice del seguente programma:

```
begin
input y;
if φ_m(m)↓_y then output φ_i(y)
       else loop forever
end.
```

1.2.4. (Suzuki) Dimostrare che non esiste una funzione totale ricorsiva f tale che, per ogni x per cui W_x è ricorsivo,

$$W_{f(x)} = \overline{W_x}.$$

Sol. Supponiamo che tale f esista. Per l'esercizio 1.2.1, esiste g, tale che

$$W_{g(x)} = \begin{cases} N & \text{se } x \in K \\ \emptyset & \text{se } x \notin K. \end{cases}$$

L'insieme $W_{g(x)}$ è ricorsivo, e dunque avremmo $W_{f(g(x))} = \overline{W_{g(x)}}$, ovvero

$$W_{f(g(x))} = \begin{cases} N & \text{se } x \notin K \\ \emptyset & \text{se } x \in K. \end{cases}$$

Ma questo è assurdo, perché equivale a

$$x \in \overline{K} \iff W_{f(g(x))} \neq \emptyset,$$

che implica la ricorsiva enumerabilità di $\overline{K}$:

$$\begin{aligned} x \in \overline{K} &\iff \exists y. y \in W_{f(g(x))} \\ &\iff \exists y \exists p. \varphi_{f(g(x))}(y)\downarrow_p \end{aligned}$$

da cui $\overline{K}$ r.e. per il teorema di proiezione.

1.2.5. Sia f una funzione totale calcolabile. Per ogni insieme r.e. W_x è possibile determinare in modo effettivo un indice per $f^{-1}(W_x)$.

Sol. Si consideri la funzione $h(x,y) \simeq \varphi_x(f(y))$. Il teorema s-m-n fornisce una funzione totale calcolabile s tale che $\varphi_{s(x)}(y) \simeq h(x,y) \simeq \varphi_x(f(y))$. Dunque $W_{s(x)} = \{y \mid f(y) \in W_x\} = f^{-1}(W_x)$. La funzione s stabilisce la tesi.

1.2.6. Dire se esiste una funzione totale ricorsiva h che, per ogni funzione calcolabile (cioè per ogni suo indice), calcola il suo minimo indice (cioè $h(i) = \min_j[\varphi_i = \varphi_j]$).

Sol. Tale funzione non può esistere, altrimenti sarebbe decidibile l'equivalenza di programmi, sarebbe cioè ricorsivo l'insieme $\{(i,j) \mid \varphi_i = \varphi_j\}$: basterebbe vedere se $h(i) \simeq h(j)$. L'insieme in questione non è invece ricorsivo (perché?).

1.2.7. Si classifichino secondo la teoria della calcolabilità l'insieme

$$A = \{i \mid \exists y\ \varphi_i(y) \simeq \varphi_y(i)\}$$

ed il suo complementare.

Sol. L'insieme è banalmente ricorsivo: per ogni i, è certo $\varphi_i(i) \simeq \varphi_i(i)$. Dunque $A = \mathbb{N}$; il complementare di un insieme ricorsivo è ricorsivo (in questo caso è vuoto).

1.2.8. Sia A un insieme r.e. Si mostri che esiste R ricorsivo tale che

$$\begin{array}{c} \forall i \in A\ \exists j \in R\ \varphi_i = \varphi_j \quad \text{and} \\ \forall j \in R\ \exists i \in A\ \varphi_i = \varphi_j \end{array}$$

(un R siffatto si dice *rappresentazione ricorsiva* di A).

Sol. Sia $A = \text{cod}\ g$ e si definisca per ricorsione primitiva la funzione

$$\begin{array}{lcl} f(0) & = & g(0) \\ f(n+1) & = & Pad(g(x), f(x)), \end{array}$$

dove $Pad(x,y)$ indica un indice $z > y$ per φ_x, ottenuto con la tecnica del *padding*. Si definisca $R = \text{cod}\ f$; la funzione f è totale e calcolabile perché Pad lo è (vedi poi). R è ricorsivo perché enumerato in modo crescente; la verifica che soddisfa le condizioni richieste non è difficile. La definizione formale di Pad è semplice: si ricordi che il lemma del *padding* assicura che esiste una funzione totale calcolabile p tale che, per ogni indice i, $\varphi_{p(i)} = \varphi_i$ ed inoltre $p(i) > i$. Definiamo prima la sua iterazione (per ricorsione primitiva):

$$\begin{array}{lcl} h(0,x) & \simeq & x \\ h(n+1,x) & \simeq & p(h(n,x)) \end{array}$$

ed infine $Pad(x,y) \simeq \mu z.h(z,x) > y$.

1.2.9. (Longo) Si consideri la seguente nozione debole di uguaglianza estensionale:

$$\varphi_i \overset{w}{\simeq} \varphi_j \text{ sse } \forall x[\varphi_i(x)\downarrow \wedge \varphi_j(x)\downarrow \Rightarrow \varphi_i(x) \simeq \varphi_j(x)].$$

Diciamo che i è il *primo indice debole* di φ_i sse $i = \min_j \varphi_i \overset{w}{\simeq} \varphi_j$. Si dimostri che l'insieme dei primi indici deboli di tutte le funzioni parziali ricorsive, cioè l'insieme

$$PID_\varphi = \{k_i \mid k_i = \min_j[\varphi_i \overset{w}{\simeq} \varphi_j] \text{ and } i \in N\}$$

è ricorsivo (PID_φ dipende ovviamente dall'enumerazione scelta φ). Si dimostri, poi, che *non* si può effettivamente costruire un indice per la sua funzione caratteristica ("effettivamente" deve essere qui inteso: "in modo uniforme a partire dall'enumerazione fissata delle parziali ricorsive").

Sol. Detta Ω la funzione ovunque indefinita, per ogni i si ha $\Omega \overset{w}{\simeq} \varphi_i$. Ma allora *PID* è finito (e dunque ricorsivo), essendo contenuto nell'intervallo $[0, u]$, dove u è il minimo indice di Ω.

Per la seconda parte dell'esercizio, sia

$$f(x,y) \simeq \begin{cases} 1 & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

Sia s la funzione del teorema s-m-n: $\varphi_{s(x)}(y) \simeq f(x,y)$ (dunque $\varphi_{s(x)}$ è la funzione costante 1 (se $x \in K$), oppure è la funzione ovunque indefinita Ω). Scelto m qualsiasi, è possibile enumerare le ricorsive parziali "scambiando tra loro le funzioni φ_0 e $\varphi_{s(m)}$". In modo più preciso: definiamo la funzione

$$\sigma(x) \simeq \begin{cases} 0 & \text{se } x = s(m) \\ s(m) & \text{se } x = 0 \\ x & \text{altrimenti.} \end{cases}$$

La σ è certo calcolabile e dunque l'enumerazione data da $\mu = \varphi \circ \sigma$ è una buona gödelizzazione di $\mathcal{C}$. Si osservi che μ_0 è la funzione ovunque indefinita sse $m \notin K$. Se fossimo in grado di produrre effettivamente la funzione caratteristica di PID_μ, essa deciderebbe dell'appartenenza di m a $\overline{K}$:

$$PID_\mu = \{0\} \text{ sse } m \in \overline{K}.$$

Per concludere basta ora osservare che σ (e dunque μ) dipende in modo uniforme da m: il preteso modo uniforme di ottenere la funzione caratteristica di *PID* darebbe un algoritmo di decisione per $\overline{K}$. (Quest'ultima parte dovrebbe essere data con maggior dettaglio: $\sigma(m, x)$, poi s-m-n ancora una volta, ecc.).

1.3 Riduzioni, insiemi creativi, produttivi, m-completi

1.3.1. Si classifichino secondo la teoria della calcolabilità l'insieme

$$\mathsf{Rec} = \{i \mid W_i \text{ ricorsivo}\}$$

ed il suo complementare.

Sol. È immediato osservare che Rec è un insieme di indici che rispetta le funzioni. Il teorema di Rice ci assicura che sia Rec che il complementare non sono ricorsivi. È poi semplice rendersi conto che l'insieme delle funzioni con dominio ricorsivo non è chiuso all'insù (si fornisca un controesempio). Lo stesso vale per $\overline{\mathsf{Rec}}$. Per il teorema di Rice-Shapiro i due insiemi assegnati non sono r.e. (e la dimostrazione del teorema prova, in realtà, che si tratta di insiemi produttivi). Forniamo comunque delle riduzioni da $\overline{K}$ che mostrino direttamente la produttività di Rec e $\overline{\mathsf{Rec}}$. Per Rec si definisca

$$h(x,y) \simeq \begin{cases} \varphi_y(y) & \text{se } x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

Dal teorema s-m-n si ottiene una funzione totale calcolabile g tale che $\varphi_{g(x)}(y) \simeq h(x,y)$. Se ora $x \in K$, $\varphi_{g(x)}$ è definita per quegli y per cui $\varphi_y(y)\downarrow$, ovvero $W_{g(x)} = K$, che implica $g(x) \notin \mathsf{Rec}$. Se, viceversa, $x \notin K$, $\varphi_{g(x)} = \Omega$, ovvero $W_{g(x)} = \emptyset$, che implica $g(x) \in \mathsf{Rec}$. Riassumendo: $x \in \overline{K}$ sse $g(x) \in \mathsf{Rec}$, ovvero g stabilisce la m-riduzione $\overline{K} \leq_m \mathsf{Rec}$.

Per $\overline{\mathsf{Rec}}$ si definisca

$$k(x,y) \simeq \begin{cases} y & \text{se } x \in K \text{ oppure } y \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

Ancora dal teorema s-m-n, si ottiene una funzione totale calcolabile t tale che $\varphi_{t(x)}(y) \simeq k(x,y)$. Si osservi ora che $x \in K$ sse $t(x) \in \mathsf{Rec}$, ovvero $K \leq_m \mathsf{Rec}$, che è lo stesso che dire $\overline{K} \leq_m \overline{\mathsf{Rec}}$.

1.3.2. Sia $A = \{i \mid \varphi_i(i) \simeq 0\}$. Si dimostri in modo diretto che $\overline{A}$ è produttivo (cioè esibendo la sua funzione produttiva, e non per riduzione).

Sol. Si definisca, tramite il teorema s-m-n, la funzione g tale che

$$\varphi_{g(x)}(y) \simeq \begin{cases} 0 & \varphi_x(y)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

e si osservi che $g(x) \in A$ sse $g(x) \in W_x$ (infatti $g(x) \in A$ sse $\varphi_{g(x)}(g(x)) \simeq 0$ sse $\varphi_x(g(x))\downarrow$ sse $g(x) \in W_x$). Un semplice ragionamento per casi mostra che g è la funzione produttiva di $\overline{A}$: se $W_x \subseteq \overline{A}$, allora $g(x) \in \overline{A} \backslash W_x$. Supposto infatti $W_x \subseteq \overline{A}$, certo $g(x) \in \overline{A}$; se fosse $g(x) \in A$, si avrebbe $g(x) \in W_x$, ovvero $g(x) \in \overline{A}$, che è assurdo. D'altra parte $g(x) \notin W_x$, perché altrimenti si avrebbe sia $g(x) \in A$ che $g(x) \notin A$.

1.3.3. Si classifichino secondo la teoria della calcolabilità l'insieme

$$B = \{i \mid \exists y\ \varphi_i(y)\downarrow \wedge \varphi_y(i)\downarrow \wedge \varphi_i(y) \simeq \varphi_y(i)\}$$

ed il suo complementare.

Sol. L'insieme B è r.e. per il teorema di proiezione:

$$x \in B \text{ sse } \exists y \exists p_1 \exists p_2\ \varphi_i(y)\downarrow_{p_1} \wedge\ \varphi_y(i)\downarrow_{p_2} \wedge\ \varphi_i(y) \simeq \varphi_y(i).$$

Il suo complementare è produttivo: mostriamo una riduzione $K \leq_m B$ (e dunque $\overline{K} \leq_m \overline{B}$). Si definisca

$$h(x,y) \simeq \begin{cases} 0 & \text{se } x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

e sia g la funzione totale calcolabile che si ottiene dal teorema s-m-n: $\varphi_{g(x)}(y) \simeq h(x,y)$. Ora $x \in K$ implica che $\varphi_{g(x)}$ è la costante 0 e, per esempio, $\varphi_{g(x)}(g(x))\downarrow$ e $\varphi_{g(x)}(g(x)) \simeq \varphi_{g(x)}(g(x))$, ovvero $g(x) \in B$. Viceversa, $x \notin K$ implica $\varphi_{g(x)} = \Omega$ e dunque, per ogni y, $\varphi_{g(x)}(y) \uparrow$ e $g(x) \notin B$. Dunque $K \leq_m B$. L'insieme B è quindi m-completo (creativo), perché r.e. e $K \leq_m B$; $\overline{B}$ è produttivo.

1.3.4. Si classifichino secondo la teoria della calcolabilità gli insiemi

$$\begin{array}{lcl} A & = & \{< i,j > \mid \forall x\ \varphi_i(\varphi_j(x)) \simeq x\} \\ B & = & \{< i,j > \mid \exists x\ \varphi_i(\varphi_j(x)) \simeq x\} \end{array}$$

ed i loro complementari. Si dica poi quali m-riduzioni tra questi quattro insiemi si possono derivare dalla classificazione così ottenuta.

Sol. B è r.e. per proiezione. $\overline{B}$ è produttivo:

$$\varphi_{h(x)}(y) \simeq \begin{cases} y & \text{se } x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

La funzione $h'(x) \simeq < h(x), h(x) >$ stabilisce la riduzione $K \leq_m B$. $\overline{A}$ è produttivo, con la stessa riduzione di $\overline{B}$. A è produttivo:

$$\varphi_{g(x)}(y) \simeq \begin{cases} y & \text{se } \varphi_x(x) \text{ non converge in al più } y \text{ passi} \\ \uparrow & \text{altrimenti.} \end{cases}$$

$g'(x) \simeq < g(x), g(x) >$ è la riduzione cercata.

1.3.5. **Numeri casuali** (Kolmogorov)
Possiamo definire la complessità di un numero x come il minimo indice di un programma che, sull'argomento 0, stampa x: $Cpl(x) = \mu k.\varphi_k(0) \simeq x$. Diremo allora che un numero x è *casuale* (secondo Kolmogorov) se la sua complessità è maggiore o uguale ad x stesso:

$$\mathsf{Rnd} = \{x \mid Cpl(x) \geq x\} = \{x \mid \forall i\ \varphi_i(0) \simeq x \Rightarrow i \geq x\}.$$

Gli elementi di Rnd sono dunque quei numeri che non hanno "descrizioni più piccole di loro". In altre parole, non esiste alcun modo più compatto per definirli che elencarne esplicitamente le cifre.

Proveremo che: (i) esistono infiniti numeri casuali; (ii) ogni funzione calcolabile (totale) g che enumera solo numeri casuali (cioè il cui codominio è contenuto in Rnd), ne enumera un sottoinsieme finito (cioè cod g è finito). Ne segue che Rnd è infinito e non contiene insiemi r.e. infiniti: è dunque un insieme *immune*. Il suo complementare

$$\overline{\mathsf{Rnd}} = \{x \mid \exists i\ \varphi_i(0) \simeq x \wedge i < x\}$$

è r.e., come si vede facilmente per proiezione: $\overline{\mathsf{Rnd}}$ è dunque un insieme *semplice*.

Dimostriamo dunque (i): Rnd è infinito. Scegliamo k arbitrario e consideriamo il minimo x che non sia stampato dai programmi da 0 a k (sull'argomento 0). Certo è $x \leq k+1$ (il caso peggiore si ha quando tutti i programmi sono definiti e stampano tutti i valori da 0 a k: in questo caso x è proprio $k+1$; se qualche programma non è definito, o se più di un programma stampa lo stesso valore, ci deve essere un numero compreso tra 0 e k che non viene stampato: esso è lo x che ci interessa). D'altra parte, ogni programma che stampa x deve avere indice maggiore o uguale a $k+1$ (ovvero $Cpl(x) \geq k+1$). Ne segue che $Cpl(x) \geq k+1 \geq x$ e dunque x è casuale. Abbiamo pertanto dimostrato che, per k *arbitrario*, esiste un numero casuale di complessità maggiore di k. Ne segue che Rnd è infinito.

Dimostriamo ora (ii), nella sua contrapposizione: per ogni f totale calcolabile con cod f infinito, cod $f \not\subseteq \mathsf{Rnd}$. Data f, definiamo la funzione

$$g(z,x) \simeq f(\mu y.f(y) > z).$$

La g è totale, perché cod f è infinito. Dal teorema s-m-n e da quello di ricorsione si ottiene e tale che

$$\varphi_e(x) \simeq g(e,x) \simeq f(\mu y.f(y) > e).$$

Osserviamo ora che $\varphi_e(0) \in$ cod f (banalmente) ed anche $\varphi_e(0) \notin \mathsf{Rnd}$, dato che $\varphi_e(0) > e$.

1.3.6. Si dimostri che l'immunità di Rnd implica l'indecidibilità del problema della fermata.

Sol. Se $Halt(x,y) \equiv \varphi_x(y)\downarrow$ fosse decidibile, sarebbe decidibile anche $Halt'(x,y,z) \equiv \varphi_x(y)\downarrow z$ e dunque calcolabile e totale la funzione $Cmp(z) \simeq \mu k.Halt'(k,0,z)$. Ma questo fornirebbe un algoritmo per decidere l'appartenenza a Rnd: $x \in \mathsf{Rnd}$ sse $x > Cmp(x)$.

1.4 Teorema di ricorsione

1.4.1. Si dimostri che esiste un indice i tale che

$$W_i = \{i, i^2, i^3, i^4, \ldots\}.$$

Sol. Sia

$$\varphi_{h(x)}(y) \simeq \begin{cases} 1 & \text{se esiste } k,\ y = x^k \\ \uparrow & \text{altrimenti.} \end{cases}$$

La funzione h è totale e calcolabile. Per il teorema di ricorsione esiste i per cui $\varphi_{h(i)} = \varphi_i$. Tale i soddisfa le richieste dell'esercizio.

1.4.2. Esiste un intero e tale che, per ogni n,

$$\varphi_e(n) \simeq \varphi_n(e)$$

Sol. Sia

$$g(k,n) \simeq \begin{cases} \varphi_n(k) & \text{se } \varphi_n(k)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

Per il teorema s-m-n, esiste s totale calcolabile per cui $\varphi_{s(i)}(n) \simeq g(k,n)$. Per il teorema di ricorsione esiste e per cui $\varphi_{s(e)} = \varphi_e$, ovvero, per ogni n,

$$\varphi_e(n) \simeq \varphi_{s(e)}(n) \simeq \begin{cases} \varphi_n(e) & \text{se } \varphi_n(e)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

che è come dire $\varphi_e(n) \simeq \varphi_n(e)$.

1.4.3. Si dimostri che l'insieme dei punti fissi di una funzione totale calcolabile non è mai finito. (Suggerimento: mostrare che in caso contrario una funzione totale calcolabile non avrebbe alcun punto fisso).

Sol. Supponiamo che, per una data f, $F = \{i \mid \varphi_i = \varphi_{f(i)}\}$ sia finito. La finitezza di F assicura l'esistenza di una funzione calcolabile g diversa da tutte quelle con indici in F ($\forall i \in F\ g \neq \varphi_i$). Si definisca ora la funzione

$$h(j,x) \simeq \begin{cases} g(x) & \text{se } j \in F \\ \varphi_{f(j)}(x) & \text{se } j \notin F. \end{cases}$$

Applicando il teorema s-m-n si ottiene una funzione totale e calcolabile s per la quale il teorema di ricorsione fornisce un indice i tale che

$$\varphi_i = \varphi_{s(i)} = \begin{cases} g & \text{se } i \in F \\ \varphi_{f(i)} & \text{se } i \notin F. \end{cases}$$

Ma questo è assurdo: se $i \in F$, $\varphi_i = g$ e ciò non può essere per la scelta di g; se $i \notin F$, $\varphi_i = \varphi_{f(i)}$, ovvero $i \in F$.

1.4.4. (Generalizzazione del precedente) Sia A l'insieme dei punti fissi di una funzione totale calcolabile f. Si mostri che se A è ricorsivo e non banale, allora contiene indici per tutte le funzioni calcolabili (cioè $\{\varphi_i \mid i \in A\} = \mathcal{C}$).

Sol. Come l'esercizio precedente.

1.4.5. Se A è ricorsivo ed è l'insieme dei punti fissi di una funzione totale calcolabile f, allora $\overline{A}$ non è un insieme di punti fissi.

Sol. Si supponga che $\overline{A}$ sia l'insieme dei punti fissi di g e si definisca la funzione

$$\varphi_{h(i)}(x) \simeq \begin{cases} \varphi_{f(i)} & \text{se } i \notin A \\ \varphi_{g(i)} & \text{se } i \in A. \end{cases}$$

La funzione h non ha punti fissi: assurdo.

1.4.6. Sia A un insieme produttivo che rispetta le funzioni. Si dimostri che $A \not\leq_m \overline{A}$. (Suggerimento: la pretesa riduzione non ha punti fissi).

Sol. Supponiamo che $h : A \leq_m \overline{A}$. Applicando il teorema di ricorsione si ottiene n tale che $\varphi_n = \varphi_{h(n)}$. Ma ora, essendo h una m-riduzione, si ha $n \in A$ sse $f(n) \in \overline{A}$; d'altra parte, siccome A rispetta le funzioni, $n \in A$ sse $f(n) \in A$. Dunque $f(n) \in A$ sse $f(n) \in \overline{A}$: assurdo.

1.4.7. Un insieme A è produttivo sse $\overline{K} \leq_m A$.

Sol. Parte centrale della dimostrazione del teorema di Myhill.

1.4.8. **Teorema di mutua ricorsione** (Smullyan)
Date due funzioni totali calcolabili f e g di due variabili esistono a e b tali che $\varphi_a = \varphi_{f(a,b)}$ e $\varphi_b = \varphi_{g(a,b)}$.

Sol. Si applichi alla funzione f il secondo teorema di ricorsione con parametri, ottenendo una funzione totale calcolabile n tale che, per ogni x, $\varphi_{n(x)} = \varphi_{t(n(x),x)}$. Consideriamo ora la funzione (totale calcolabile) $h(z) \simeq g(n(z), z)$, ed applichiamo ad essa il teorema di ricorsione; si ottiene b tale che $\varphi_b = \varphi_{g(n(b),b)}$. Si prenda infine $a = n(b)$.

Negli esercizi è talvolta conveniente usare questo teorema sulle coppie codificate, ovvero nell'enunciato: date due funzioni totali calcolabili unarie f e g, esistono a e b tali che $\varphi_a = \varphi_{f(<a,b>)}$ e $\varphi_b = \varphi_{g(<a,b>)}$.

1.4.9. Enunciare e dimostrare un teorema di mutua ricorsione (vedi 1.4.8) con parametri (si veda l'esercizio 1.5.7 per un esempio d'uso).

1.5 Riduzioni che sfruttano il teorema di ricorsione

1.5.1. Si classifichino secondo la teoria della calcolabilità l'insieme

$$A = \{i \mid \forall y\ \varphi_i(y) \simeq \varphi_y(i)\}$$

ed il suo complementare.

Sol. A è produttivo, per riduzione da $\overline{K}$. Sia

$$h(x,z,y) \simeq \begin{cases} \varphi_y(z) & \text{se } \varphi_x(x) \text{ non converge in al più } y \text{ passi} \\ \uparrow & \text{altrimenti.} \end{cases}$$

Applicando il teorema s-m-n si ottiene una funzione totale e calcolabile g tale che $\varphi_{g(x,z)}(y) \simeq h(x,z,y)$, per ogni x, y, z. Si osservi che quando $x \in \overline{K}$ si ha $\varphi_{g(x,z)}(y) \simeq \varphi_y(z)$; quando invece $x \in K$, la funzione $\varphi_{g(x,z)}$ ha dominio finito (perché a partire da un certo y_0, per ogni $y \geq y_0$, $\varphi_x(x)\downarrow y$. Applicando il teorema di ricorsione con parametro si ottiene una funzione totale e calcolabile n tale che, per ogni x, $\varphi_{n(x)} = \varphi_{g(x,n(x))}$. Se dunque $x \in \overline{K}$, $\varphi_{n(x)}(y) \simeq \varphi_y(n(x))$ e $n(x) \in A$. Se, invece, $x \in K$, $\varphi_{n(x)}$ ha dominio finito e dunque esisterà certo un y per il quale $\varphi_{n(x)}(y) \not\simeq \varphi_y(n(x))$ (per esempio un indice abbastanza grosso di una funzione costante). La funzione n stabilisce la riduzione cercata: $\overline{K} \leq_m A$.

L'insieme $\overline{A}$ è produttivo. Mostriamo infatti una riduzione $K \leq_m A$ (da cui $\overline{K} \leq_m \overline{A}$). Sia

$$h(x,z,y) \simeq \begin{cases} \varphi_y(z) & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

Applicando, come nel caso precedente, prima il teorema s-m-n e poi quello di ricorsione con parametro si ottiene una funzione totale calcolabile m tale che

$$\varphi_{m(x)}(y) \simeq \begin{cases} \varphi_y(m(x)) & \text{se } x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

È chiaro che se $x \notin K$, $m(x)$ è un indice della funzione ovunque divergente, che non appartiene certo ad A (stesso argomento del caso precedente).

1.5.2. Si classifichino, dal punto di vista della teoria della calcolabilità, gli insiemi

$$\begin{array}{rcl} A & = & \{< i,j > \mid \varphi_i \text{ assume il valore } i \text{ per esattamente } j \text{ argomenti distinti}\} \\ B & = & \{< i,j > \mid \varphi_i \text{ assume il valore } i \text{ per almeno } j \text{ argomenti distinti}\} \end{array}$$

ed i loro complementari.

Sol. L'insieme A è produttivo, per riduzione da $\overline{K}$. Sia

$$h(x,y) \simeq \begin{cases} y & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

Dal teorema s-m-n otteniamo una funzione totale calcolabile k tale che $\varphi_{k(x)}(y) \simeq h(x,y)$. Si ha: $x \in K \Rightarrow \varphi_{k(x)} = id \Rightarrow < k(x), 0 > \notin A$; d'altra parte, $x \notin K \Rightarrow \varphi_{k(x)} = \Omega \Rightarrow < k(x), 0 > \in A$. Ovvero $\overline{K} \leq_m A$, attraverso la funzione $f(x) \simeq < k(x), 0 >$.

L'insieme $\overline{A}$ è produttivo. Dimostriamo che $K \leq_m A$, da cui $\overline{K} \leq_m \overline{A}$ e, dunque, la tesi. Definiamo la funzione

$$h(z,x,y) \simeq \begin{cases} z & \text{se } (x = 0 \text{ or } \ldots \text{or } x = z-1) \text{ and } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti.} \end{cases}$$

Applicando s-m-n e poi, come prima, il teorema di ricorsione con parametro, si ottiene una funzione totale calcolabile m tale che: $x \in K$ implica $\varphi_{m(x)}$ assume il valore $m(x)$ esattamente $m(x)$ volte; $x \in \overline{K}$ implica $\varphi_{m(x)} = \Omega$. Dunque, supposto che l'enumerazione delle funzioni *non* assegni 0 come indice di Ω (si osservi che è sempre possibile scegliere un'enumerazione siffatta), $x \in K$ sse $< m(x), m(x) > \in A$.

L'insieme B è r.e. per proiezione:

$$B = \{< i,j > \mid \exists q \exists p \ \mathrm{Seq}(q) \wedge \mathrm{lung}(q) < j \wedge \forall x < j \ \varphi_i((q)_x)\underset{p}{\downarrow}\}$$

L'insieme $\overline{B}$ è produttivo: $x \in K$ sse $< n(x), 1 > \in B$, dove n è la funzione ottenuta nella prima parte dell'esercizio.

Ne segue che B, essendo r.e. con complementare produttivo, è creativo e, dunque, m-completo per il teorema di Myhill.

1.5.3. Dati due insiemi di naturali A e B, si definisce l'insieme $A \cdot B$ nel modo seguente:

$$A \cdot B = \{z \mid \exists a \in A \ \exists b \in B \ \varphi_a(b) \simeq z\}.$$

Si classifichino secondo la teoria della calcolabilità gli insiemi

$$\begin{array}{rcl} C & = & \{< i,j >\mid W_i \cdot W_j = \emptyset\} \\ D & = & \{< i,j >\mid W_i \cdot W_j = \{i\}\} \end{array}$$

ed i loro complementari.

Sol. L'insieme $\overline{C}$ è r.e., come facilmente si vede dalla sua definizione

$$\overline{C} = \{< i,j >\mid \exists a \in W_i \ \exists b \in W_j \ \varphi_a(b)\downarrow\}$$

e dal teorema di proiezione, ricordando che l'appartenenza ad un insieme r.e. (per esempio $a \in W_i$) è un predicato r.e. (in a e i).

L'insieme C è produttivo (e dunque $\overline{C}$ è creativo, ovvero m-completo). Si osservi infatti che

$$x \in \overline{K} \Longleftrightarrow \{x\} \cdot \{x\} = \emptyset.$$

Rimane solo da esplicitare la dipendenza tra x ed un indice per $\{x\}$: la funzione totale h tale che

$$\varphi_{h(x)}(y) \simeq \begin{cases} 0 & \text{se } y = x \\ \uparrow & \text{altrimenti} \end{cases}$$

serve allo scopo: per ogni x, $W_{h(x)} = \{x\}$. Ricapitolando, si ha

$$x \in \overline{K} \Longleftrightarrow < h(x), h(x) > \in C.$$

Gli insiemi D e $\overline{D}$ sono entrambi produttivi. Per mostrare una riduzione da $\overline{K}$ a $\overline{D}$, si può partire da quella appena data per C: invece di un singoletto $\{x\}$, con $x \in \overline{K}$, dobbiamo ora formare singoletti $\{n(x)\}$ tali che $\varphi_{n(x)}(n(x)) \simeq n(x)$ quando $x \in K$. La funzione n si ottiene da una variante della funzione h:

$$\varphi_{k(x,z))}(y) \simeq \begin{cases} z & \text{se } y = z \wedge x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

e dal secondo teorema di ricorsione con parametro, che dà una funzione totale calcolabile n tale che, per ogni x, $\varphi_{n(x)} = \varphi_{k(x,n(x)))}$, ovvero

$$W_{n(x)} = \begin{cases} \{n(x)\} \text{ t.c. } \varphi_{n(x)}(n(x)) \simeq n(x) & \text{se } x \in K \\ \emptyset & \text{altrimenti.} \end{cases}$$

La funzione che manda x in $< n(x), n(x) >$ fornisce la riduzione $K \leq_m D$.

Per la riduzione $\overline{K} \leq_m B$, possiamo modificare la definizione della funzione k, in modo da farla comunque convergere su z (ovvero, sull'indice "di se stessa", dopo l'applicazione del teorema di ricorsione):

$$\varphi_{t(x,z))}(y) \simeq \begin{cases} z & \text{se } y = z \\ y & \text{se } x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

Procedendo ora come prima, si ottiene una funzione totale calcolabile m tale che:

$$W_{m(x)} = \begin{cases} \{m(x)\} \text{ t.c. } \varphi_{m(x)}(m(x)) \simeq m(x) & \text{se } x \in \overline{K} \\ N & \text{altrimenti.} \end{cases}$$

Osservando che $N \cdot N = N$, la riduzione cercata è stabilita da $p(x) \simeq < m(x), m(x) >$.

1.5.4. Si classifichino secondo la teoria della calcolabilità gli insiemi

$$\begin{array}{rcl} C & = & \{< i, j >| \ |W_i \cdot W_j| = 2 \wedge i \in W_i \cdot W_j\} \\ D & = & \{< i, j >| \ |W_i \cdot W_j| = i \wedge i \in W_i \cdot W_j\} \end{array}$$

ed i loro complementari (si veda l'esercizio 1.5.3 per la definizione di $A \cdot B$)..

Sol. Gli insiemi assegnati sono tutti produttivi. Una riduzione $K \leq_m C$ può essere stabilita a partire dalla funzione

$$\varphi_{h(x,z))}(y) \simeq \begin{cases} y & \text{se } (y = z \vee y = e) \wedge x \in K \\ \uparrow & \text{altrimenti,} \end{cases}$$

dove e è un indice (fissato) di Ω. Dal secondo teorema di ricorsione con parametro si ottiene una funzione totale calcolabile n tale che, per ogni x, $\varphi_{n(x)} = \varphi_{k(x,n(x)))}$ è l'identità ristretta a

$$W_{n(x)} = \begin{cases} \{n(x), e\} & \text{se } x \in K \\ \emptyset & \text{altrimenti.} \end{cases}$$

La funzione che manda x in $< n(x), n(x) >$ fornisce la riduzione $K \leq_m C$ (si osservi che $n(x) \neq e$).

Per la riduzione $\overline{K} \leq_m C$, usiamo la funzione

$$\varphi_{k(x,z))}(y) \simeq \begin{cases} y & \text{se } y = z \vee y = e \vee x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

ed il teorema di ricorsione (dettagli lasciati per esercizio).

La riduzione $K \leq D$ si può ottenere con la funzione

$$\varphi_{s(x,z))}(y) \simeq \begin{cases} y & \text{se } (y = z \vee y \in \{e_1, \ldots, e_{z-1}\}) \wedge x \in K \\ \uparrow & \text{altrimenti,} \end{cases}$$

dove $e_1, \ldots, e_{z-1}$ sono $z - 1$ indici di Ω, ottenuti in modo ricorsivo (per esempio usando il lemma del *padding*) a partire da un indice e fissato (formalmente, la condizione $y \in \{e_1, \ldots, e_{z-1}\}$ si scriverà $y \in W_{p(z)}$, con p funzione totale calcolabile che si ottiene dal *padding* e $W_{p(z)}$ finito). Dal teorema di ricorsione si ottiene una funzione totale calcolabile q tale che $\varphi_{q(x)}$ è l'indentità ristretta a

$$W_{q(x)} = \begin{cases} \{q(x), e_1, \ldots, e_{q(x)-1}\} & \text{se } x \in \overline{K} \\ \emptyset & \text{altrimenti.} \end{cases}$$

La riduzione cercata è stabilita dalla funzione che manda x in $< q(x), q(x) >$.

Infine, la riduzione $\overline{K} \leq_m D$ si stabilisce con la funzione

$$\varphi_{s(x,z))}(y) \simeq \begin{cases} y & \text{se } y = z \vee y \in \{e_1, \ldots, e_{z-1}\} \vee x \in K \\ \uparrow & \text{altrimenti.} \end{cases}$$

1.5.5. Si classifichino secondo la teoria della calcolabilità gli insiemi

$$\begin{array}{rcl} A & = & \{i \mid \forall x.\varphi_i(x)\downarrow \Rightarrow \varphi_i(x) \leq i\} \\ B & = & \{i \mid |\{x \mid \varphi_i(x)\downarrow \wedge \varphi_i(x) > i\}| = 4\} \end{array}$$

ed i loro complementari ($|A|$ denota la cardinalità dell'insieme A).

Sol. $\overline{A} = \{i \mid \exists x.\varphi_i(x)\downarrow \wedge \varphi_i(x) > i\}$ è r.e. per il teorema di proiezione. Mostriamo una riduzione $K \leq_m \overline{A}$, col che $\overline{A}$ è creativo e A produttivo.

$$\varphi_{h(x,z))}(y) \simeq \begin{cases} z+1 & \text{se } x \in K \\ \uparrow & \text{altrimenti,} \end{cases}$$

Dal secondo teorema di ricorsione con parametro si ottiene una funzione totale calcolabile n tale che, per ogni x, $\varphi_{n(x)} = \varphi_{h(x,n(x))}$; ora si ha $x \in K$ sse $n(x) \in \overline{A}$.

Sia B che $\overline{B}$ sono produttivi.

1.5.6. Si classifichino secondo la teoria della calcolabilità l'insieme

$$A = \{i \mid W_i = \{\pi_0(i) + i\}\}$$

ed il suo complementare.

1.5.7. Si classifichino secondo la teoria delle calcolabilità l'insieme

$$C = \{x \mid |\{y \mid \varphi_{\pi_0(x)}(y) \simeq \pi_1(x)\}| = x\}$$

ed il suo complementare.

Sol. Stabiliamo che $K \leq_m C$. Sia

$$\varphi_{h(x,z)}(y) \simeq \begin{cases} \pi_1(z) & \text{se } x \in K \text{ e } y < z \\ \uparrow & \text{altrimenti,} \end{cases}$$

e si applichi il teorema di mutua ricorsione con parametro (1.4.9) ad h e alla proiezione $p(x,z) \simeq x$. Si ottiene una funzione m tale che

$$\begin{array}{rcl} \varphi_{\pi_0(m(x))} & = & \varphi_{h(x,m(x))} \\ \varphi_{\pi_1(m(x))} & = & \varphi_{p(x,m(x))} = \varphi_x \end{array}$$

Se $x \in K$, si ha

$$\varphi_{\pi_0(m(x))}(y) \simeq \begin{cases} \pi_1(m(x)) & \text{se } y < z \\ \uparrow & \text{altrimenti,} \end{cases}$$

da cui $m(x) \in C$. Se $x \notin K$, $\varphi_{\pi_0(m(x))} = \Omega$, ovvero $m(x) \notin C$. Dunque $m : K \leq_m C$.

Una riduzione $\overline{K} \leq_m C$ si ottiene partendo dalla funzione

$$\varphi_{k(x,z)}(y) \simeq \begin{cases} \pi_1(z) & \text{se } x \in K \text{ oppure } y < z \\ \uparrow & \text{altrimenti,} \end{cases}$$

e procedendo in modo analogo.

In conclusione sia C che $\overline{C}$ sono produttivi.

1.5.8. Si considerino i seguenti insiemi ed i loro complementari:

$$\begin{array}{rcl} A &=& \{< i,j > \mid |E_i| = j\}, \\ B &=& \{< i,j > \mid |E_i| < j\}, \\ C &=& \{< i,j > \mid |E_i| = 2\}. \end{array}$$

Si dimostri se ognuno di questi sei insiemi è o meno $\leq_m$ rispetto a ciascuno dei cinque rimanenti.

Sol. Per risolvere il problema è opportuno individuare la classificazione di alcuni insiemi, da cui dedurre l'esistenza (o meno) di alcune relazioni; potremo poi determinare le eventuali riduzioni mancanti. Sia $\mathsf{Div} = \{i \mid \varphi_i = \Omega\}$.

(a): $\mathsf{Div} \leq_m A$ è facile: $i \in \mathsf{Div}$ sse $< i, 0 > \in A$. Dalla produttività di Div segue quella di A.

(b): $\mathsf{Div} \leq_m B$, con la stessa riduzione di (a): B è produttivo.

(c): $\overline{B}$ è r.e., dal teorema di proiezione.

(d): $C \leq_m A$ è evidente: $< i,j > \in C$ sse $< i, 2 > \in A$.

(e): C e $\overline{C}$ sono entrambi produttivi, come si vede senza grande difficoltà modificando leggermente le riduzioni della dimostrazione del teorema di Rice-Shapiro.

(f): Da (e) e (d) si ottiene che $\overline{A}$ è produttivo.

Da questi primi fatti possiamo già dedurre l'esistenza di molte riduzioni; le riassumiamo nella tabella seguente ($\surd$ indica l'esistenza della riduzione; $\times$ la non esistenza):

$\leq_m$	A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$
A	$\surd$	$\times$			$\times$	
B	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$
C	$\surd$	$\times$	$\surd$		$\times$	

Le riduzioni della riga B si ottengono in virtù dell'esercizio 1.4.7.

(g): $A \leq_m C$. Si tratta di una riduzione non banale. Introduciamo in primo luogo la notazione $\mathrm{Bcod}(\varphi_i, y)$ per indicare il "codominio di φ_i ristretta a $[0, y]$ per y passi":

$$\mathrm{Bcod}(\varphi_i, y) = \{z \mid \exists x \in [0, y]\ \varphi_i(x)\downarrow_y z\}$$

Definiamo ora la funzione

$$\varphi_{s(<i,j>)}(y) \simeq \begin{cases} 0 & |\mathrm{Bcod}(\varphi_i, y)| < j \\ 1 & |\mathrm{Bcod}(\varphi_i, y)| = j \\ 2 & |\mathrm{Bcod}(\varphi_i, y)| > j \end{cases}$$

Osserviamo che, purché $j \neq 0$, $|E_i| = j$ sse $|E_{s(<i,j>)}| = 2$. Per ottenere la riduzione cercata occorre trattare il caso $j = 0$. Ma si osservi che $< i, 0 > \in A$ sse $< i, 0 > \in B$ sse $f(< i, 0 >) \in C$, dove $f : B \leq_m C$. In conclusione, la funzione

$$g(x) \simeq \begin{cases} f(x) & \text{se } \pi_1(x) = 0 \\ s(x) & \text{altrimenti} \end{cases}$$

stabilisce la cercata riduzione $A \leq_m C$. (**Problema:** Esiste una riduzione più semplice?)

(h): (g) e (d) danno $A \equiv_m C$.

(i): per riempire le quattro caselle rimaste, è sufficiente domandarsi ora se $C \leq_m \overline{C}$: questo problema è infatti equivalente, in virtù di (h), agli altri tre. Si ha che $C \not\leq_m \overline{C}$ (si supponga il contrario con funzione r e si applichi il teorema di mutua ricorsione alla funzione r e all'identità, ottenendo n tale che $\varphi_{\pi_0(n)} = \varphi_{r(n)}$ e derivando un assurdo, secondo lo schema dell'esercizio 1.4.6).

In conclusione, la tabella seguente riassume le risposte al quesito posto:

$\leq_m$	A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$
A	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$
B	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$
C	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$

1.6 Operatori ricorsivi, λ-calcolo

1.6.1. Data una funzione totale calcolabile f, si consideri l'insieme

$$\mathrm{Fix}(f) = \{i \mid \varphi_i = \varphi_{f(i)}\}.$$

Si diamo esempi di funzioni per cui $\mathrm{Fix}(f)$ è ricorsivo, r.e. non ricorsivo, produttivo.

Sol. Il ragionamento è semplificato scegliendo f estensionale, perché in tal modo $\mathrm{Fix}(f)$ è un insieme che rispetta le funzioni e vi possiamo applicare i noti teoremi.

- Sia $f = id$. Allora $\mathrm{Fix}(f) = \mathbb{N}$. Inoltre, per f estensionale, $\mathrm{Fix}(f)$ è ricorsivo se e solo se coincide con $\mathbb{N}$ (per il teorema di Rice, osservando che $\mathrm{Fix}(f)$ non è mai vuoto, per il teorema di ricorsione).
- Sia

  $$h(i,x) \simeq \begin{cases} 0 & \text{se } x = 0 \\ \varphi_i(x) & \text{altrimenti.} \end{cases}$$

  Dal teorema s-m-n otteniamo una funzione totale calcolabile estensionale t (un'operazione effettiva) tale che $\varphi_{t(i)}(x) \simeq h(i,x)$. $\mathrm{Fix}(t) = \{i \mid \varphi_i(0) = 0\}$, che è r.e. per proiezione e non ricorsivo per il teorema di Rice.
- Sia f estensionale tale che l'operatore ricorsivo associato $\hat{f}$ abbia come minimo punto fisso una funzione totale g (e dunque sia unico). In tal caso $\mathrm{Fix}(f)$ è l'insieme degli indici di g, che non è r.e. per il teorema di Rice-Shapiro (dalla cui dimostrazione si ottiene anche una riduzione da $\overline{K}$ che mostra che $\mathrm{Fix}(f)$ è produttivo).

1.6.2. È dato l'operatore

$$\Psi(f)(x,y,z) \simeq \begin{cases} y \cdot y & \text{se } x = 0 \\ f(x-1,y,z)+1 & \text{se } x > 0 \text{ e } z = 0 \\ f(x,y,z-1) & \text{se } x > 0 \text{ e } z > 0. \end{cases}$$

Ψ (ristretto alle funzioni calcolabili) è un operatore ricorsivo? In caso negativo, mostrare che il suo minimo punto fisso non è calcolabile. In caso positivo, fornire un lambda-termine che lambda-definisce il minimo punto fisso di Ψ.

Sol. Ψ è ricorsivo: segue dal teorema di Myhill-Shepherdson, dal momento che Ψ corrisponde ad un'operazione effettiva (in virtù della calcolabilità della definizione per casi ed il teorema s-m-n). Quanto al suo minimo punto fisso, si vede immediatamente che il terzo argomento della funzione $\Psi(f)$ non è significativo (cioè non contribuisce al risultato). Il minimo punto fisso di Ψ può dunque essere espresso nel modo seguente, come si dimostra per induzione su x:

$$\begin{aligned} h(0,y,z) &\simeq y \cdot y \\ h(x+1,y,z) &\simeq h(x,y,z)+1 \end{aligned} \tag{1}$$

ovvero

$$h(x,y,z) \simeq y^2 + x.$$

Un termine che λ-definisce h è dunque

$$\lambda xyz.\underline{+}\ (\underline{*}\ y\ y)x$$

dove $\underline{+}$ e $\underline{*}$ sono, rispettivamente, i λ-termini che definiscono somma e prodotto.

Alternativamente (ma in modo assai più lungo...), osservando che la (1) è una definizione per ricorsione primitiva, si può usare il procedimento standard visto a lezione.

1.6.3. Per ogni naturale n, possiamo definire induttivamente il lambda-termine $\ulcorner n \urcorner$ come segue:

$$\ulcorner 0 \urcorner = \lambda x.x \qquad \ulcorner n+1 \urcorner = \lambda x.x\ulcorner n \urcorner$$

Dato ora il linguaggio

$$L = \{a^n b^m c^p \mid \ulcorner 0 \urcorner \ulcorner n \urcorner =_\beta \ulcorner 0 \urcorner \vee (\lambda x.x\ulcorner 0 \urcorner)\ulcorner m \urcorner =_\beta (\lambda x.\lambda y.yx)\ulcorner p \urcorner\},$$

(a) si dia una definizione di L che non faccia riferimento al lambda-calcolo;
(b) si dimostri se L è regolare, libero, ovvero non libero da contesto.

Sol. (a) $L = \{a^n b^m c^p \mid n = 0 \vee m = n+2\}$.
(b) L è libero perché generato dalla grammatica

$$\begin{array}{lcl} S & ::= & T \mid ARcc \\ T & ::= & bT \mid cT \mid \epsilon \\ R & ::= & bRc \mid \epsilon \\ A & ::= & a \mid aA \end{array}$$

L non è regolare, perché altrimenti lo sarebbe anche L^R. Sia N la costante per L^R di cui il *pumping lemma* asserisce l'esistenza e consideriamo la stringa $z = c^N b^{N+2} a$. In ogni scomposizione $z = uvw$ con $|uv| \leq N$ si ha $v = c^k$, che non può essere pompata.

1.6.4. **Teorema di ricorsione ed operatore di punto fisso**
L'operatore di punto fisso del λ-calcolo

$$\mathsf{Y} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

corrisponde in modo preciso alla dimostrazione del secondo teorema di ricorsione. Si ricordi che nel λ-calcolo non c'è distinzione tra programmi ed argomenti (non c'è enumerazione esplicita dei programmi): l'applicazione del programma x all'argomento y (che nel caso classico si dovrebbe intendere "del programma di indice x all'argomento y" e scrivere $\varphi_x(y)$) si denota semplicemente con xy. Ne segue che anche il teorema s-m-n diviene "banale", corrispondendo a nient'altro che l'aggiunta di un λ: data un'espressione (λ-termine) $P_{x,y}$ che dipende da x ed y, la "funzione s del teorema s-m-n" (che nel caso classico soddisfa la relazione $\varphi_{s(x)}(y) \simeq P_{x,y}$) corrisponde direttamente a $S = \lambda x.P_{x,y}$. Infatti, per ogni x, $Sx =_\beta P_{x,y}$. La possibilità di non distinguere tra numeri come argomenti e numeri come indici di programmi, rende la costruzione di Y particolarmente semplice e mette in luce la presenza delle diagonalizzazioni (autoapplicazioni) nella dimostrazione del secondo teorema di ricorsione. Ripercorreremo dunque tale dimostrazione, costruendo nel contempo l'operatore Y.

Data f totale calcolabile,	f
si consideri l'autoapplicazione $\varphi_x(x)$,	xx
e quindi la funzione s tale che $\varphi_{s(x)} = \varphi_{f(\varphi_x(x))}$;	$\lambda x.f(xx)$
detto e un indice di s, si consideri $n = \varphi_e(e)$;	$N \equiv (\lambda x.f(xx))(\lambda x.f(xx))$
n è l'indice cercato: $\varphi_n = \varphi_{f(n)}$.	N soddisfa $N =_\beta (fN)$

Se adesso si astrae N rispetto ad f, si ottiene Y. Si osservi che quest'ultimo passaggio corrisponde ad un rafforzamento dell'usuale enunciato del teorema di ricorsione: l'indice n si ottiene in modo uniforme (cioè totale e calcolabile) da un indice per f. Nel λ-calcolo questa uniformità è espressa, in modo assai perspicuo, con una semplice λ-astrazione.

2 Linguaggi formali

2.0.1. È dato il linguaggio $L = \{0^n 1^n \mid n \geq 0\}$.

(a) Dire, dimostrando la risposta, se i seguenti linguaggi sono regolari:
$L_0 = L(0+1)^* L^R$
$L_1 = 0^* L$
$L_2 = 1^* L$
$L_3 = 11^* L$

(b) Determinare una grammatica libera G che generi L_3.

(c) Porre G in forma normale di Greibach.

Sol. L_0 è regolare: $L_0 = (0+1)^*$, dal momento che $\epsilon \in L$. Nessuno degli altri linguaggi è regolare. Per L_1 e L_2 si può usare il *pumping lemma*. Ad esempio, per L_1: sia N la costante di cui il lemma assicura l'esistenza e si consideri la stringa $z = 0^N 1^N$; in ogni scomposizione $z = uvw$ con $|uv| \leq N$ e $|v| \geq 1$ si ha $v = 0^k$, con $k \geq 1$; "pompando all'ingiù" v si ottiene la contraddizione: $uv^0w = 0^{N-k}1^k \notin L$.
Per L_3 il *pumping lemma* non può essere applicato direttamente (perché?). Osserviamo, però, che le le stringhe 101, 1001, 10001,..., 10^m1, definiscono tutte classi di equivalenza diverse ripetto alla relazione indotta dal linguaggio (xRy sse per ogni z, $[xz \in L$ sse $yz \in L])$. Siccome tali classi risultano infinite, il teorema di Myhill-Nerode ci assicura che L non è regolare. Alternativamente, la non regolarità si può stabilire usando un omomorfismo inverso. Si definisca $f : \{a,b,c\}^* \to \{0,1\}^*$ con $f(a) = 1$, $f(b) = 0$ e $f(c) = 1$. Si ha $L' = f^{-1}(L_3) \cap a^*b^*c^* = \{aa^*b^nc^n \mid n \geq 0\}$. Sostituiamo ora ad ogni a la stringa vuota: $g(a) = \epsilon$, $g(b) = 0$, $g(c) = 1$. Si ha $g(L') = L$. Siccome i linguaggi regolari sono chiusi per sostituzione, intersezione ed omomorfismi inversi, se L_3 fosse regolare, lo sarebbe anche L.

2.0.2. Mostrare che, se L è un linguaggio regolare sull'alfabeto Σ, allora

$$L' = \{x \mid \exists y \in \Sigma^*.yx \in L\}$$

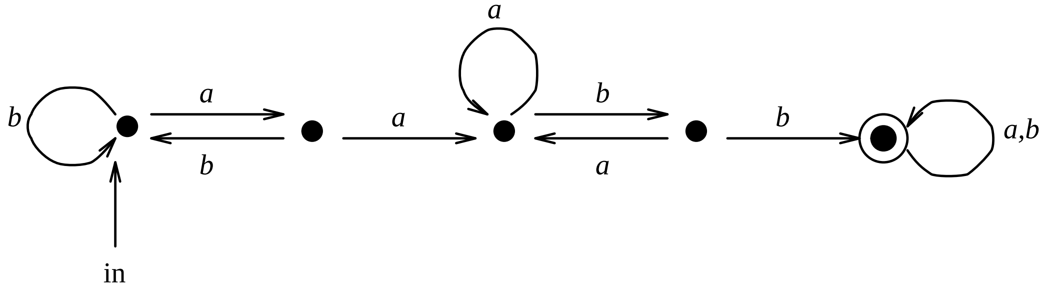
è regolare.

Sol. Ricordiamo che, se L è regolare, allora L^R è regolare e ragioniamo come segue:

$$\begin{array}{rcl} (L')^R & = & \{x^R \mid \exists y \in \Sigma^*.yx \in L\} \\ & = & \{x^R \mid \exists y \in \Sigma^*.(yx)^R \in L^R\} \\ & = & \{x^R \mid \exists y \in \Sigma^*.x^R y^R \in L^R\} \\ & = & L^R/\Sigma^* \end{array}$$

Ne consegue che $(L')^R$ (e dunque L') è regolare, perché quoziente di linguaggi regolari.
In modo alternativo, si poteva costruire un automa M' per L' a partire da uno (M) per L (Idea: detto q_0 lo stato iniziale di M, l'automa M' deve avere un nuovo stato iniziale, con una ϵ-transizione verso q_0 ed ogni altro stato di M raggiungibile da q_0).

2.0.3. Sia L il linguaggio accettato dall'automa seguente:



(a) Si determini L.
(b) Posto $L_1 = bb(a+b)^*$, si determini L/L_1.
(c. Determinare un linguaggio libero non regolare $L_2 \subseteq L/L_1$, giustificando tale risposta.
(d) Si dica se il linguaggio $L_3 = \{www \mid w \in L/L_1\}$ è regolare, libero, ovvero non libero da contesto.

Sol. (a) $L = (b+ab)^*aa(a+ba)^*bb(a+b)^*$
(b) $L/L_1 = (b+ab)^*aa(a+ba)^*$.
(c) Ad esempio $\{(ab)^m aa(ba)^m \mid m \geq 0\}$, generato da $S ::= aa \mid abSba$, per dimostrare la non regolarità del quale si può usare il *pumping lemma* con la stringa $(ab)^N aa(ba)^N$.

2.0.4. Sia

$$L = \{0^n 1^n \mid n \geq 0\}.$$

Determinare i seguenti linguaggi:

$$L^c, L/L, L^c/L^c, L/L^c, L(L^c)^R$$

dimostrando per ciascuno di essi se sia regolare, libero da contesto, ovvero non libero da contesto ($L^c = \Sigma^* - L$, $\Sigma = \{0,1\}$).

Sol.: L^c non è certo regolare, perché i regolari sono chiusi per complementazione e L è noto come non regolare. $L^c = \Sigma^*10\Sigma^* \cup \{0^h1^k \mid h \neq k\}$ e si tratta di un linguaggio libero, essendo generato dalla grammatica seguente:

$$\begin{array}{lcl} S & ::= & R \mid 0Z \mid U1 \\ Z & ::= & 0Z \mid P \\ U & ::= & U1 \mid P \\ P & ::= & 0P1 \mid \epsilon \\ R & ::= & \text{produzioni per l'espressione regolare } \Sigma^*10\Sigma^*. \end{array}$$

$L/L = L$, libero non regolare.
$L^c/L^c = \Sigma^*$, regolare.
$L/L^c = \{0^h1^k \mid h > k\}$, non regolare (*pumping lemma*, per esempio con stringa 0^N1^{N-1}, pompando all'insù), libero (grammatica analoga a quella di L^c, omettendo tutte le produzioni che nominano R ed U).
$L(L^c)^R = \Sigma^* - \{1^n0^n \mid n \geq 0\}$, equivalente ad L^c.

2.0.5. Fissato $\Sigma = \{a,b\}$, per ogni $y \in \Sigma^*$ si definisca

$$r(y) = (\text{numero di } a \text{ in } y) - (\text{numero di } b \text{ in } y).$$

Si studino i seguenti linguaggi:

$$\begin{array}{lcl} L_1 & = & \{x \in \Sigma^* \mid \forall y \text{ segmento iniziale di } x,\ r(y) \geq 0\} \\ L_2 & = & \{x \in \Sigma^* \mid \forall y \text{ segmento iniziale di } x,\ 2 \geq r(y) \geq 0\} \\ L_3 & = & L_1/(L_2\{\epsilon\}). \end{array}$$

Sol. L_1 non è regolare: se lo fosse, sia N la costante del *pumping lemma* e scegliamo $z = a^N b^N$. In ogni scomposizione $z = uvw$ con $|uv| \leq N$ si ha $uv = a^k$. Essendo $|v| \geq 1$, si ottiene $uv^0w = a^hb^N$ con $h < N$, che non appartiene a L_1. L_1 è libero: un semplice automa a pila che lo riconosce ha due stati q_0 (iniziale e finale) e q_1. Da q_0 si mette in pila A tutte le volte che si legge a e si toglie un A quando si legge b e sulla pila c'è almeno un A. Se si legge b e sulla pila vi è il solo simbolo di fine-pila, l'automa transita in q_1 da cui cicla in q_1 con qualsiasi input e qualsiasi carattere in pila.

L_2 è regolare (automa non difficile).

$L_3 = L_1$.

2.0.6. (a) Sia L un linguaggio regolare sull'alfabeto Σ. Stabilire se i seguenti linguaggi sono in generale regolari:

$L_1 = \{xa \mid ax \in L\}$,
$L_2 = \{bx \mid \exists a \in \Sigma.ax \in L\}$,
$L_3 = \{xcy \mid |x| = |y| \ \& \ cxy \in L\}$ ($c \in \Sigma$ carattere fissato).

(b) Si considerino gli insiemi $A_1 = \{3k + 1 \mid k \geq 0\}$ ed $A_2 = \{2k \mid \geq 0\}$. Siano L_1 ed L_2 i linguaggi definiti esprimendo gli elementi di A_1 e A_2, rispettivamente, in base 3 (pertanto $L_1, L_2 \subseteq \{0, 1, 2\}^*$). Si stabilisca se L_1 ed L_2 sono liberi (Suggerimento per L_2: un numero espresso in base 3 è divisibile per 2 sse lo è la somma delle sue cifre).

2.0.7. (a) Stabilire se i seguenti linguaggi sull'alfabeto $\Sigma = \{0, 1\}$ sono regolari o liberi.

- $L_1 = \{x \in \Sigma^* \mid$ il terz'ultimo carattere è $0\}$;
- $L_2 = \{x \in \Sigma^* \mid$ il terz'ultimo carattere è 0 se $|x|$ è pari$\}$;
- $L_3 = \{x \in \Sigma^* \mid \lfloor |x|/2 \rfloor$-esimo elemento è $0\}$;
- $L_4 = \{xwx \mid x, w \in \Sigma^* \setminus \{\epsilon\}\}$;
- $L_5 = \{xwx^R \mid x, w \in \Sigma^* \setminus \{\epsilon\}\}$;
- $(L_5)^c$.

(b) Si determini un linguaggio infinito L'_1 tale che per ogni linguaggio libero L'_2 $\{xwx \mid x, w \in L'_1, w \in L'_2\}$ è libero.

3 Semantica

3.0.1. Nel contesto del linguaggio IMP, si definisca la semantica denotazionale del nuovo comando **wa** b **do** $c_1 \bullet c_2$ in modo che soddisfi la seguente relazione, per ogni $b \in$ BExpr e $c_1, c_2 \in$ Com:

$$\begin{array}{ll}\textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2 \approx\ \textbf{if}\ b & \textbf{then}\ c_1; (\textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2); c_2 \\ & \textbf{else skip}\end{array}$$

Si diano poi regole di semantica operazionale strutturata per lo stesso comando e si dimostri l'equivalenza delle due semantiche.

Sol.

$$\mathcal{C}[\![\textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2]\!] = \mathit{fix} H$$

$$H = \lambda f.\lambda s. COND(\mathcal{B}[\![b]\!]s, \mathcal{C}[\![c_2]\!](f(\mathcal{C}[\![c_1]\!]s)), s)$$

Le regole di semantica operazionale sono:

$$\frac{< b, s > \rightarrow \mathit{true} \quad < c_1, s > \rightarrow s_1 \quad < \textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2, s_1 > \rightarrow s_2 \quad < c_2, s_2 > \rightarrow s'}{< \textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2, s > \rightarrow s'}$$

e

$$\frac{< b, s > \rightarrow \mathit{false}}{< \textbf{wa}\ b\ \textbf{do}\ c_1 \bullet c_2, s > \rightarrow s.}$$

L'equivalenza da dimostrare assume la forma seguente, per ogni $c' \in$ Com:

$$\forall s \in \text{Store} < c', s > \rightarrow s' \iff \mathcal{C}[\![c']\!]\lfloor s \rfloor = \lfloor s' \rfloor$$

L'implicazione da sinistra a destra si stabilisce per induzione sulla derivazione senza grossi problemi (il solo caso da mostrare è, ovviamente, quello relativo a **wa**). Da destra a sinistra l'implicazione si dimostra per induzione strutturale su c'. Nell'unico caso interessante, quando c' è **wa** b **do** $c_1 \bullet c_2$, si procede con un'ulteriore induzione sugli approssimanti di $\mathit{fix} H$.

3.0.2. (a) Sia D un dominio puntato e sia $f : D \rightarrow D$ monotona. Si diano condizioni sufficienti affinché $\forall t \in B_\perp$ e $\forall x, y \in D$

$$f(COND(t, x, y)) = COND(t, f(x), g(x)).$$

(b) Si consideri la seguente dichiarazione in REC:

$$\begin{array}{rcll} h(x) & = & d_1 & \\ g(x) & = & d_2 & \\ p(x) & = & d_3 & \\ f(x,y) & = & \textbf{if}\ p(x) & \textbf{then}\ y \\ & & & \textbf{else}\ h(f(g(x), y)) \end{array}$$

dove d_1, d_2 e d_3 sono generici termini con $FV(d_i) \subseteq \{x\}$. Assumendo la semantica in *call-by-value*, si dimostri che, per ogni coppia di termini chiusi t_1 e t_2,

$$h(f(t_1, t_2)) = f(t_1, h(t_2)).$$

Cosa si può dire nel caso della semantica in *call-by-name*?

Sol. (a). In generale si ha

$$f(COND(t, x, y)) \geq COND(t, f(x), g(x))$$

L'uguaglianza si ottiene se f è stretta: $f(\perp_D) = \perp_D$.

(b). Si dimostra per induzione di punto fisso (il predicato è certo inclusivo per il teorema di Manna-Veuillemen), sfruttando il fatto dimostrato in (a).

Nel caso del *call-by-name*, si dimostra la più debole $h(f(t_1, t_2)) \geq f(t_1, h(t_2))$.

3.0.3. Si dimostri che, per ogni f, g funzioni continue tra gli opportuni c.p.o. puntati, vale

$$\mathit{fix}(f \circ g) = f(\mathit{fix}(g \circ f)).$$

Sol.: Dimostriamo per prima cosa che

$$\mathit{fix}(f \circ g) \leq f(\mathit{fix}(g \circ f)). \tag{2}$$

A tale scopo basterà mostrare che il membro destro è un punto fisso di $f \circ g$: siccome $\mathit{fix}(f \circ g)$ è il *minimo* punto fisso, si ottiene la disuguaglianza voluta. Dunque:

$$\begin{array}{rcl} (f \circ g)(f(\mathit{fix}(g \circ f))) & = & f(g(f(\mathit{fix}(g \circ f)))) \\ & = & f(\mathit{fix}(g \circ f)). \end{array}$$

Per dimostrare l'altra disuguaglianza, cioè

$$\mathit{fix}(f \circ g) \geq f(\mathit{fix}(g \circ f)),$$

osserviamo che questa, per definizione di punto fisso, può essere riscritta come

$$f(g(\mathit{fix}(f \circ g))) \geq f(\mathit{fix}(g \circ f)),$$

che è implicata, per la monotonia di f, da

$$g(\mathit{fix}(f \circ g)) \geq \mathit{fix}(g \circ f).$$

Quest'ultima, infine, non è che la (2), a meno dello scambio tra f e g.

3.0.4. Data la seguente definizione alternativa della semantica denotazionale del comando while di IMP:

$$\mathcal{C}'[\![\textbf{while } b \textbf{ do } c]\!] = (\mathit{fix} H)\mathcal{C}[\![c]\!]$$

$$\begin{array}{rcl} H & = & \lambda f \in [\mathrm{Store}_\perp \to \mathrm{Store}_\perp] \to [\mathrm{Store}_\perp \to \mathrm{Store}_\perp]. \\ & & \lambda K \in [\mathrm{Store}_\perp \to \mathrm{Store}_\perp].\lambda s \in \mathrm{Store}_\perp. \\ & & \mathit{COND}(\mathcal{B}[\![b]\!]s, fK(Ks), s), \end{array}$$

si dimostri che per ogni $b \in \mathrm{BExp}$ ed ogni $c \in \mathrm{Com}$ si ha

$$\mathcal{C}'[\![\textbf{while } b \textbf{ do } c]\!] = \mathcal{C}[\![\textbf{while } b \textbf{ do } c]\!].$$

Sol. Detto G il funzionale il cui punto fisso costituisce la semantica standard di while:

$$\begin{array}{rcl} G & = & \lambda g \in [\mathrm{Store}_\perp \to \mathrm{Store}_\perp] \\ & & \lambda s \in \mathrm{Store}_\perp. \\ & & \quad \mathit{COND}(\mathcal{B}[\![b]\!]s, g(\mathcal{C}[\![c]\!]s), s), \end{array}$$

e definito il predicato binario

$$P(f, g) \equiv \forall s \in \mathrm{Store}_\perp \; f\mathcal{C}[\![c]\!]s = gs,$$

si chiede dunque di dimostrare che vale

$$P(\mathit{fix} H, \mathit{fix} G).$$

Trattandosi di un predicato inclusivo, possiamo stabilire la tesi per induzione di punto fisso rispetto ai due funzionali H e G.

Base: $P(\perp, \perp)$ è ovvio.

Passo: supponiamo che valga $P(f, g)$; si ha

$$\begin{array}{rcl} P(Hf, Gg) & \equiv & \forall s \in \mathrm{Store}_\perp \\ & & \mathit{COND}(\mathcal{B}[\![b]\!]s, f\mathcal{C}[\![c]\!](\mathcal{C}[\![c]\!]s), s) = \mathit{COND}(\mathcal{B}[\![b]\!]s, g(\mathcal{C}[\![c]\!]s), s) \\ & \equiv & \forall s \in \mathrm{Store}_\perp \; f\mathcal{C}[\![c]\!](\mathcal{C}[\![c]\!]s) = g(\mathcal{C}[\![c]\!]s) \\ & \equiv & \textbf{true} \end{array}$$

dove l'ultima equivalenza si ottiene per ipotesi induttiva.

3.0.5. Nel contesto di IMP con errori aritmetici, si definisca la semantica denotazionale del comando **stop** in modo che soddisfi la seguente relazione, per ogni comando c_1 e c_2:

$$c_1\,;\mathbf{stop} \approx c_1\,;\mathbf{stop}; c_2 \tag{1}$$

Si dimostri la (1) per la semantica data.

Sol.

$$\begin{array}{rcl} \mathcal{C}[\![\mathbf{stop}]\!] & = & \underline{\lambda} s.\text{case } s \text{ of} \\ & & \quad ER(s').ER(s') \mid \\ & & \quad OK(s').ER(s') \\ & & \quad \text{end} \end{array}$$

Se $\mathcal{C}[\![c_1]\!]s = \bot$ è facile vedere che membro sinistro e destro della (1) hanno entrambi valore $\bot$. In caso contrario, per $s = L(s')$, con $s' \in$ Store e $L \in \{OK, ER\}$, si ha

$$\mathcal{C}[\![c_1; \mathbf{stop}; c_2]\!]s = \mathcal{C}[\![c_2]\!](ER(s')) = ER(s') = \mathcal{C}[\![c_1; \mathbf{stop}]\!]s$$

dove la seconda uguaglianza sfrutta una proprietà generale della semantica di IMP con errori:

$$\forall s \in \text{Store}, \mathcal{C}[\![c]\!]ER(s) = ER(s).$$

3.0.6. Nel contesto del linguaggio IMP, si definisca la semantica denotazionale del nuovo comando **wa** b_1 **do** $b2 \bullet c_1 c_2$ in modo che soddisfi la seguente relazione, per ogni $b_1, b_2 \in$ BExpr e $c_1, c_2 \in$ Com:

$$\begin{array}{lll} \mathbf{wa}\ b_1\ \mathbf{do}\ b_2 \bullet c_1 c_2 \approx \mathbf{if}\ b_1 & \mathbf{then}\ c_1; \mathbf{wa}\ b_1\ \mathbf{do}\ b_2 \bullet c_1 c_2 & \\ & \mathbf{else}\ \ \mathbf{if}\ b_2 & \mathbf{then}\ c_2; \mathbf{wa}\ b_1\ \mathbf{do}\ b_2 \bullet c_1 c_2 \\ & & \mathbf{else\ skip} \end{array}$$

Si diano poi regole di semantica operazionale strutturata per lo stesso comando e si dimostri l'equivalenza delle due semantiche.

4 Prove d'esame

15 febbraio 1995, compitino: 1.5.1; 1.6.1; 1.6.2.
30 maggio 1995, compitino: 2.0.1; 2.0.2; 3.0.1.
6 giugno 1995, esame: 1.3.4; 1.6.3; 2.0.3; 3.0.2.
27 giugno 1995, esame: 1.5.3; 2.0.4; 3.0.3.
10 luglio 1995, esame: 1.5.4; 2.0.5; 3.0.4.
5 settembre 1995, esame: 1.5.5; 2.0.6; 3.0.5.
3 ottobre 1995, esame: 1.5.6; 2.0.7; 3.0.6.