(da Esame del 10/02/2009 http://goo.gl/7rVV9 – Esercizio 1)

Siano f e g due funzioni primitive ricorsive tali che $f(x) \neq g(x)$, per ogni $x \in \mathbb{N}$.
Si dimostri che esiste una funzione totale e non calcolabile h tale che

$$h(x) \in \{f(x), g(x)\}, \text{ per ogni } x \in \mathbb{N}. \qquad (1)$$

Esiste più di una funzione h totale e non calcolabile soddisfacente la condizione (1)?

<u>Soluzione</u> **(*di* reversengineer)**

Si consideri $h(x) \stackrel{\text{def}}{=} \begin{cases} f(x) & \text{se } \phi_x(x) = g(x) \\ g(x) & \text{se } \phi_x(x) \neq g(x) \end{cases}$ (questo caso include implicitamente anche $\phi_x(x)\uparrow$ poiché g è totale)

Questa funzione è non calcolabile.
Per assurdo, se $h(x)$ fosse calcolabile, sia e l'indice di un programma che calcola h.
Se $\phi_e(e) = g(e)$ (*), allora:
$\phi_e(e) =_1 h(e) =_2 f(e) \neq_3 g(e) = \phi_e(e) \Longrightarrow \phi_e(e) \neq \phi_e(e) \Longrightarrow 0 \neq 0$ Assurdo
Se $\phi_e(e) \neq g(e)$ (**), allora:
$\phi_e(e) =_1 h(e) =_2 g(e) \neq_4 \phi_e(e) \Longrightarrow \phi_e(e) \neq \phi_e(e) \Longrightarrow 0 \neq 0$ Assurdo

$=_1$: poiché e è il codice di un programma che calcola h per ipotesi assurda
$=_2$: per definizione di h
$\neq_3$: per ipotesi (*)
$\neq_4$: per ipotesi (**)

In tutti i possibili casi, si ottiene un assurdo. L'ipotesi che avevamo fatto arbitrariamente era che h fosse calcolabile, perciò $\boldsymbol{h}$ **non è calcolabile.**

Inoltre, i due casi per cui è definita $h(x)$ sono mutualmente esclusivi ed esauriscono le possibilità, perciò $h(x)$ è totale (sempre definita) e assume sempre uno dei valori $f(x)$ oppure $g(x)$.

Esiste una seconda possibile definizione di funzione che abbia le caratteristiche richieste, ed è:

$h'(x) \stackrel{\text{def}}{=} \begin{cases} g(x) & \text{se } \phi_x(x) = f(x) \\ h(x) & \text{se } \phi_x(x) \neq f(x) \end{cases}$ (questo caso include implicitamente anche $\phi_x(x)\uparrow$ poiché f è totale)

Si vede che $h' \neq h$ poiché per i valori x: $\phi_x(x)\uparrow\ h(x) = g(x) \neq f(x) = h'(x)$
Tali valori x esistono, basta considerare $x =$ "indice di $f_\emptyset$" $\in \mathbb{N}$. ■

(da Esame del 10/02/2009 http://goo.gl/7rVV9 – Esercizio 2)

(a) Si definisca l'operatore *minimalizzazione* e si enunci una sua proprietà.
(b) Si dimostri che se h è una funzione totale e calcolabile, allora anche la seguente funzione risulta calcolabile:

$$\ell(x) = \begin{cases} h(2x) + 1 & \text{se } h(x) \text{ e } h(x^2) \text{ sono entrambi pari} \\ \uparrow & \text{altrimenti.} \end{cases}$$

(c) La funzione ℓ può essere primitiva ricorsiva per qualche scelta opportuna della funzione h?

Soluzione **(*di* reversengineer)**

(a) Domanda di teoria (non esercizio) a cui si è risposto in aula il 20/04/2011, che comunque è solo la definizione già presente sulle slide;

(b) ℓ si può scrivere nella forma $\ell(x) = h(2x) + \underline{1}\Big(\mu z\big(\mathrm{Even}(h(x)) \wedge \mathrm{Even}(h(x^2))\big)\Big)$

$h(x)$ è calcolabile per ipotesi
x^2 è calcolabile (primitiva ricorsiva)
$h(x^2)$ è calcolabile per composizione
Even è decidibile, in quanto la sua funzione caratteristica è calcolabile (addirittura primitiva ricorsiva): infatti la stessa funzione si può definire per ricorsione primitiva da f e g

$$\begin{cases} \mathrm{Even}(0) = 0 \\ \mathrm{Even}(n+1) = \overline{\mathrm{sg}}(\mathrm{Even}(n)) \end{cases} \text{ ponendo } \begin{matrix} f(x) = \underline{0}(x) \\ g(y,z) = \overline{\mathrm{sg}}(z) \end{matrix}$$

AND di predicati decidibili è decidibile
l'operatore di minimalizzazione applicato a predicati decidibili produce funzioni calcolabili
$\underline{1}$ è calcolabile per composizione ($\underline{1}(x) = \mathrm{succ}\big(\underline{0}(x)\big)$
$2x$ è calcolabile (primitiva ricorsiva)
$h(2x)$ è calcolabile per composizione
la somma è calcolabile (primitiva ricorsiva) perciò **ℓ è calcolabile**.
Inoltre, se l'operatore di minimo (che non dipende da z) trova che almeno uno fra $h(x)$ e $h(x^2)$ non è pari, banalmente andrà in loop, non definendo ℓ per tale valore x, mentre se entrambi sono pari, in tal punto x la minimalizzazione sarà definita (e pari banalmente a 0) e composta con il resto dell'espressione produrrà esattamente e correttamente il valore $h(2x) + 1$.

(c) Per avere una ℓ primitiva ricorsiva, è necessario innanzitutto che h sia primitiva ricorsiva; inoltre, per avere la totalità di ℓ (necessaria quando si parla di primitiva ricorsività) bisogna imporre che h produca sempre e solo valori pari.

■

non è necessario!

Es. $\begin{cases} h(2x) = 2 \\ h(2x+1) = A(x,x) \end{cases}$ (A è la funzione di Ackermann)

$\ell(x) = 3, \forall x$, ma h non è primitiva ricorsiva.

Più semplicemente, rispondere così:

(c) Se h è primitiva ricorsiva e $h(x)$ è pari per ogni $x \in \mathbb{N}$, allora $\ell(x) = h(2x)+1$ e quindi ℓ è primitiva ricorsiva.

(da Esame del 03/03/2009 http://goo.gl/wh8jE - Esercizio 2)

Una funzione f si dice *quasi-costante* se f è costante a meno di un numero finito di valori, cioè se esiste $X \subseteq \mathbb{N}$ tale che $\mathbb{N} \backslash X$ è finito ed f è costante su X.

Si stabilisca se le seguenti asserzioni sono corrette o meno, giustificando con un esempio o una dimostrazione – a seconda del caso – le risposte date:

(a) esiste una funzione totale e quasi-costante che non è calcolabile;
(b) esiste una funzione totale che non è né calcolabile né quasi-costante.

Soluzione **(*di* reversengineer)**

(a) è corretto, ma un po' complicato; lo risolverei come nella pagina seguente.

(a) No; per assurdo esista questa funzione f totale e quasi-costante che non è calcolabile.
Sia $B = \mathbb{N} \backslash X$ l'insieme **finito** su cui f non è costante (ma comunque totale per ipotesi) e sia k il valore costante che viene assunto dalla funzione nei punti di X.
Siccome B è finito, allora esiste una biiezione $h: B \to \{0, 1, \dots, |B| - 1\}$. Sia h sia la sua inversa h^{-1} sono banalmente calcolabili (è una definizione di funzione per casi a partire da predicati mutualmente esclusivi e la cui disgiunzione esclusiva (XOR) è sempre vera, in cui ogni caso è un valore calcolabile).
Adesso consideriamo la funzione:

$$g(x) = \begin{cases} f\left(h\left(\mu_{z<|B|}(h^{-1}(z) = x)\right)\right) & \text{se } x \in B \\ k & \text{altrimenti} \end{cases}$$

(formulazione alternativa in un'unica espressione:

$$g(x) = f\left(h\left(\mu_{z<|B|}(h^{-1}(z) = x)\right)\right) \cdot \text{rm}\big(|B| + 1, \mu_{z<|B|}(h^{-1}(z) = x) + 1\big)$$
$$+ \overline{\text{sg}}\left(\text{rm}\big(|B| + 1, \mu_{z<|B|}(h^{-1}(z) = x) + 1\big)\right) \cdot k$$

)
Si osserva che $g(x) = f(x)$, tuttavia g è calcolabile (oltre che totale), perciò anche f (che è lo stesso identico oggetto funzione g) dev'essere calcolabile $\Longrightarrow$ assurdo, doveva essere f non calcolabile.
L'ipotesi assurda che abbiamo fatto arbitrariamente era che f esistesse, perciò ***f* siffatta non esiste**.

(b) Dire che una funzione non è quasi-costante significa che tale funzione ha codominio infinito.
(siccome il codominio può avere solo numeri interi non negativi, tale infinità dovrà necessariamente essere numerabile).Ora si consideri la funzione:

non è corretto! Es. la funzione $x \bmod 2$ non è quasi-costante, ma ha codominio finito.

$$f(x) = \begin{cases} \phi_x(x) + 1 & \text{se } \phi_x(x) \downarrow \\ 0 & \text{altrimenti} \end{cases}$$

Tale funzione è non calcolabile.
Se, per assurdo, f fosse calcolabile, sia e il codice di un programma che calcola f.
Se $\phi_e(e) \downarrow$, allora $\phi_e(e) = f(e) = \phi_e(e) + 1 \Longrightarrow \phi_e(e) = \phi_e(e) + 1 \Longrightarrow 0 = 1$ Assurdo
Se $\phi_e(e) \uparrow$, allora $\phi_e(e) = f(e) = 0$ ma $(\phi_e(e) \uparrow \wedge f(e) \downarrow) \Longrightarrow \phi_e \neq f$ Assurdo
In ogni caso, otteniamo un assurdo. L'assurdo nasce dall'unica ipotesi azzardata, cioè che f fosse calcolabile, di conseguenza ***f* non è calcolabile**.

Però ogni funzione con codominio infinito non è quasi-costante.

Inoltre, consideriamo la successione di codici di programma:
$\{e_a\}_{a \in \mathbb{N}}$ ove e_a = "codice del programma canonico che calcola la funzione costante $a(x) = a, \forall x \in \mathbb{N}$"
Tutte le funzioni costanti sono calcolabili, di conseguenza per ciascuna di esse esiste il codice del programma canonico che la calcola.
Quindi $\forall a \in \mathbb{N}, \phi_{e_a}(a) = a \Longrightarrow \forall a \in \mathbb{N}, f(e_a) = \phi_{e_a}(e_a) + 1 = a + 1$
Detto in altre parole, $\forall a \in \mathbb{N}, \exists n = e_a : a \in E_{\phi_n} \Longrightarrow \forall a \in \mathbb{N}, \exists n \in \mathbb{N} : f(n) = (a + 1)$
Quindi, $\cancel{E_f = \mathbb{N}^+}$ che è ~~numerabile~~ infinito. $\text{Ran}(f) = \mathbb{N}$
Perciò, f definita come sopra è una funzione totale che non è né calcolabile né quasi-costante.■

↑ corretto.

(a) Sia $f : \mathbb{N} \to \mathbb{N}$ una funzione totale e quasi-costante.

Siano $X \subseteq \mathbb{N}$, $k \in \mathbb{N}$ tali che

- $\mathbb{N} \setminus X$ finito
- $f(x) = k$, per ogni $x \in X$

Sia $\mathbb{N} \setminus X = \{y_1 \ldots y_n\}$ e siano

$c_1 = f(y_1), \ldots, c_n = f(y_n)$.

Allora

$$f(x) = \begin{cases} c_1 & \text{se } x = y_1 \\ \vdots & \\ c_n & \text{se } x = y_n \\ k & \text{altrimenti} \end{cases} \qquad (*)$$

Poichè i predicati $x = y_i$ sono decidibili e insieme al predicato $x \neq y_1 \wedge \ldots \wedge x \neq y_n$ (caso "altrimenti") mutuamente esclusivi, da $(*)$ segue che f è calcolabile.

(da Esame del 16/06/2009 http://goo.gl/kpbij - Esercizio 1)

Si dimostri che esiste una funzione f unaria, totale e *non* calcolabile tale che $f(x) = x^2$, per ogni $0 \leq x \leq 100$.

<u>Soluzione</u> **(*di* reversengineer)**

$$f(x) = \begin{cases} x^2 & \text{se } 0 \leq x \leq 100 \\ \phi_{x-101}(x) + 1 & \text{se } x \geq 101 \end{cases}$$

f è non calcolabile.
Per assurdo, se f fosse calcolabile, sia e il codice di un programma che calcola f. Siccome se esiste un programma che calcola f allora ne esistono infiniti, allora esistono infiniti codici associati a programmi che calcolano f. Posso, quindi, scegliere $e \geq 101$ senza perdere generalità. Sia $e \geq 101$.
Considero $\phi_e(e + 101) =_1 f(e + 101) =_2 \phi_{e+101-101}(e + 101) + 1 =_3 \phi_e(e + 101) + 1$

$=_1$: perché e è codice di un programma che calcola f
$=_2$: per definizione di f
$=_3$: per semplificazione del pedice

Quindi, congiungendo gli estremi, $\phi_e(e + 101) = \phi_e(e + 101) + 1 \Longrightarrow 0 = 1$ Assurdo.
L'assurdo è nato dopo aver sfruttato l'ipotesi assurda che f fosse calcolabile, quindi $\boldsymbol{f}$ **è non calcolabile**.

Inoltre, si vede subito dalla sua definizione, che $f(x) = x^2$ per ogni $0 \leq x \leq 100$. ■

(da Esame del 08/07/2009 http://goo.gl/I4vzY – Esercizio 1)

Utilizzando il metodo diagonale di Cantor, si dimostri che esiste una funzione f unaria, totale e non calcolabile il cui codominio è contenuto nell'insieme $\{101, 102, 103\}$.

<u>Soluzione</u> **(*di* reversengineer)**

Si consideri $f(x) = \begin{cases} 103 & \text{se } \phi_x(x) = 101 \vee \phi_x(x) = 102 \\ 101 & \text{altrimenti} \end{cases}$

Tale funzione è non calcolabile.
Se per assurdo f fosse calcolabile, sia e il codice di un programma che la calcola.
Se $\phi_e(e) = 101 \vee \phi_e(e) = 102$ (*), allora:
$\phi_e(e) =_1 f(e) =_2 103 \neq_3 \phi_e(e) \Longrightarrow \phi_e(e) \neq \phi_e(e) \Longrightarrow 0 \neq 0$ Assurdo
Se $\phi_e(e) \neq 101 \wedge \phi_e(e) \neq 102$ (**), allora:
$\phi_e(e) =_1 f(e) =_2 101 \neq_4 \phi_e(e) \Longrightarrow \phi_e(e) \neq \phi_e(e) \Longrightarrow 0 \neq 0$ Assurdo

$=_1$: perché e è codice di un programma che calcola f
$=_2$: per definizione di f
$\neq_3$: per la condizione (*)
$\neq_4$: per la condizione (**)

In entrambi i casi, otteniamo un assurdo. L'assurdo proviene dall'aver fatto l'unica ipotesi arbitraria, che f fosse calcolabile. Quindi $\boldsymbol{f}$ **non è calcolabile**.

Inoltre, f è totale e il suo codominio è $\{101, 103\} \subseteq \{101, 102, 103\}$. ■

(da Esame del 15/09/2009 http://goo.gl/Fx2I1 – Esercizio 1)

Utilizzando il metodo diagonale di Cantor, si dimostri che esiste una funzione $f: \mathbb{N} \to \mathbb{N}$ totale e non calcolabile avente come codominio l'insieme $\{0, 1, 2, \dots, 9\}$.

Soluzione **(*di* reversengineer)**

Si consideri $f(x) = \begin{cases} x & \text{se } 0 \le x \le 9 \\ 9 - \phi_x(x) & \text{se } x \ge 10 \land 0 \le \phi_x(x) \le 9 \\ \mathrm{rm}\big(10, \phi_x(x)\big) & \text{se } x \ge 10 \land \phi_x(x) \ge 10 \\ 0 & \text{se } x \ge 10 \land \phi_x(x) \uparrow \end{cases}$

Tale funzione è non calcolabile.
Se per assurdo f fosse calcolabile, sia e il codice di un programma che la calcola. Siccome se esiste un programma che calcola f allora ne esistono infiniti, allora esistono infiniti codici associati a programmi che calcolano f. Posso, quindi, scegliere $e \ge 10$ senza perdere generalità. Sia $e \ge 10$.

- Se $0 \le \phi_e(e) \le 9$, allora:
  $\phi_e(e) =_1 f(e) =_2 9 - \phi_e(e) \ne_3 \phi_e(e) \Longrightarrow \phi_e(e) \ne \phi_e(e)$ Assurdo
- Se $\phi_e(e) \ge 10$, allora:
  $\{10, 11, \dots\} \ni \phi_e(e) =_1 f(e) =_2 \mathrm{rm}\big(10, \phi_e(e)\big) \in \{0, 1, \dots, 9\} \Longrightarrow$
  $\Longrightarrow \phi_e(e) \in \{10, 11, \dots\} \land \phi_e(e) \in \{0, 1, \dots, 9\} \Longrightarrow \phi_e(e) \in \{10, 11, \dots\} \cap \{0, 1, \dots, 9\} = \emptyset \Longrightarrow \nexists \phi_e(e)$
  Assurdo, perché era $\phi_e(e) \in \mathbb{N}$
- Se $\phi_e(e) \uparrow$, allora $\phi_e(e) = f(e) = 0$ ma $(\phi_e(e) \uparrow \land f(e) \downarrow) \Longrightarrow \phi_e \ne f$ Assurdo

In tutti i casi, ottengo un assurdo. L'assurdo nasce dall'aver fatto l'unica ipotesi arbitraria, che f fosse calcolabile, di conseguenza **f è non calcolabile**.

Inoltre:

1) Se $0 \le x \le 9$, allora $0 \le f(x) = x \le 9 \Longrightarrow f(x) \in \{0, 1, \dots, 9\}$
2) Se $x \ge 10 \land 0 \le \phi_x(x) \le 9$, allora $0 \le f(x) = 9 - \phi_x(x) \le 9 \Longrightarrow f(x) \in \{0, 1, \dots, 9\}$
3) Se $x \ge 10 \land \phi_x(x) \ge 10$, allora $f(x) = \mathrm{rm}\big(10, \phi_x(x)\big) \in \{0, 1, \dots, 9\}$
4) Se $x \ge 10 \land \phi_x(x) \uparrow$, allora $f(x) = 0 \in \{0, 1, \dots, 9\}$

In tutti i casi, si ha che $f(x) \in \{0, 1, \dots, 9\}$, e in più, dal caso **1**), $\forall y \in \{0, 1, \dots, 9\} \exists x = y: f(x) = y$, perciò il codominio di f coincide con tutto $\{0, 1, \dots, 9\}$. ■