Computabilità e Algoritmi (Computabilità)
Prova Intermedia - 20 Novembre 2017

Esercizio 1

Enunciare il teorema s-m-m. Utilizzarlo per dimostrare che esiste una funzione calcolabile totale $k : \mathbb{N} \to \mathbb{N}$ tale che $W_{k(n)} = \{x \in \mathbb{N} \mid x \geqslant n\}$ e $E_{k(n)} = \{y \in \mathbb{N} \mid y \text{ pari}\}$ per ogni $n \in \mathbb{N}$.

Soluzione: Si inizia definendo una funzione calcolabile di due argomenti $f(n,x)$ che rispetti le condizioni quando vista come funzione di x, con n considerato come parametro, ad es.

$$f(n,x) = \begin{cases} 2 * (x \dot{-} n) & \text{se } x \geqslant n \\ \uparrow & \text{altrimenti} \end{cases} = 2 * (x \dot{-} n) + \mu z.(n \dot{-} x)$$

Per il teorema smn esiste una funzione totale calcolabile $k : \mathbb{N} \to \mathbb{N}$ tale che $\varphi_{k(n)}(x) = f(n,x)$ per ogni $n, x \in \mathbb{N}$. Pertanto, come desiderato

- $W_{k(n)} = \{x \mid f(n,x) \downarrow\} = \{x \mid x \geqslant n\}$;
- $E_{k(n)} = \{f(n,x) \mid x \in \mathbb{N}\} = \{2(x \dot{-} n) \mid x \geqslant n\} = \{2(n+z) \dot{-} n) \mid z \geqslant 0\} = \{2z \mid z \in \mathbb{N}\}$.

$\square$

Esercizio 2

Si consideri una variante della macchina URM, che comprende le istruzioni di salto, trasferimento e due nuove istruzioni

- *A(m,n)* che scrive nel registro m la somma dei registri m e n, ovvero $r_m \leftarrow r_m + r_n$;
- *C(n)* che scrive nel registro n il valore del suo segno, ovvero $r_n \leftarrow sg(r_n)$.

Dire quale relazione sussiste tra l'insieme $\mathcal{C}'$ delle funzioni calcolabili con la nuova macchina e l'insieme $\mathcal{C}$ delle funzioni calcolabili con la macchina URM. Sono uno contenuto nell'altro? L'inclusione è stretta? Motivare le risposte.

Soluzione: Indichiamo con URM* la macchina modificata. Osserviamo che le istruzioni della macchina URM* sono codificabili nella macchina URM.

L'istruzione $I_j : A(m,n)$ si può sostituire con un salto alla seguente routine: detto q l'indice del primo registro non usato dal programma (quindi inizialmente a 0)

$$\begin{array}{lll} SUB & : & J(n,q,j+1) \\ & & S(m) \\ & & S(q) \\ & & J(1,1,SUB) \end{array}$$

Allo stesso modo, indicando ancora con q l'indice di un registro non utilizzato, un'istruzione $I_j : C(m)$ si può sostuire con un salto alla subroutine

$$\begin{array}{lll} SUB & : & J(n,q,ZERO) \\ & & Z(n) \\ & & J(1,1,j+1) \\ ZERO & : & S(n) \\ & & J(1,1,j+1) \end{array}$$

Più formalmente, si prova che $\mathcal{C}^* \subseteq \mathcal{C}$ dimostrando che, per ogni numero di argomenti k per ogni un programma P che utilizzi entrambi i set di istruzioni si può ottenere un programma URM P' che calcola la stessa funzione ovvero tale che $f_{P'}^{(k)} = f_P^{(k)}$.

La prova procede per induzione sul numero h di istruzioni A e C nel programma. Il caso base $h = 0$ è banale, dato che P con 0 istruzioni A e C è già un programma URM. Supposto vero il risultato per h dimostriamolo per $h+1$. Il programma P contiene certamente almeno una istruzione A o una istruzione C. Supponiamo che sia l'istruzione di indice j e sia una istruzione A.

$$\begin{array}{lll} 1 & : & I_1 \\ & & \dots \\ j & & A(m,n) \\ & & \dots \\ \ell(P) & : & I_{\ell(P)} \end{array}$$

Costruiamo un programma P'', utilizzando un registro non riferito da P, $q = \max\{\rho(P), k\} + 1$

$$\begin{array}{lll} 1 & : & I_1 \\ & & \dots \\ j & & J(1,1,SUB) \\ & & \dots \\ \ell(P) & : & I_{\ell(P)} \\ & & J(1,1,END) \\ SUB & : & J(n,q,ZERO) \\ & & Z(n) \\ & & J(1,1,j+1) \\ ZERO & : & S(n) \\ & & J(1,1,j+1) \\ END & : & \end{array}$$

Il programma P'' è tale che $f_{P''}^{(k)} = f_P^{(k)}$ e contiene h istruzioni di tipo A o C. Per ipotesi induttiva, esiste un programma URM P' tale che $f_{P'}^{(k)} = f_{P''}^{(k)}$ che quindi è il programma desiderato.

Se l'istruzione I_j è di tipo C si procede in modo completamente analogo, rimpiazzando l'istruzione con la sua codifica e usando l'ipotesi induttiva.

L'inclusione è stretta, ovvero $\mathcal{C} \nsubseteq \mathcal{C}^*$. Ad esempio si può facilmente vedere che la funzione successore non è URM* calcolabile. Infatti, si può dimostrare che partendo da una configurazione con i registri tutti a zero, un qualunque programma URM*, dopo un qualunque numero di passi, produrrà una configurazione con i registri ancora tutti a zero. Una prova completamente formale dimostra la tesi suddetta per induzione sul numero di passi. □

Esercizio 3

Esiste una funzione totale non calcolabile $f : \mathbb{N} \to \mathbb{N}$, tale che la sua immagine $cod(f) = \{y \mid \exists x \in \mathbb{N}.\ f(x) = y\}$ sia finito? Fornire un esempio o mostrare che una tale funzione non esiste.

Soluzione: Si esiste. Ad esempio, basta considerare:

$$f(x) = \begin{cases} \overline{sg}(\varphi_x(x)) & \text{se } x \in W_x \\ 0 & \text{se } x \notin W_x \end{cases}$$

Allora la funzione f

- è totale;
- non è calcolabile in quanto per ogni $x \in \mathbb{N}$ si ha che $f(x) \neq \varphi_x(x)$; infatti, se $\varphi_x(x) \downarrow$ allora $f(x) = \overline{sg}(\varphi_x(x)) \neq \varphi_x(x)$, e se invece $\varphi_x(x) \uparrow$ allora $f(x) = 0 \neq \varphi_x(x)$;
- chiaramente $cod(f) \subseteq \{0, 1\}$.

□