Computabilità
23 Novembre 2020 - prova intermedia

Esercizio 1
Si consideri una variante URM^p della macchina URM nella quale l'istruzione di azzeramento $Z(n)$ è sostituita dell'istruzione di predecessore $P(n)$ che decrementa il contenuto del registro n, ovvero $r_n \leftarrow r_n \dot{-} 1$. Indicato con $\mathcal{C}^p$ l'insieme delle funzioni calcolabili dalla macchina URM^p, stabilire la relazione tra $\mathcal{C}^P$ e l'insieme delle funzioni URM-calcolabili $\mathcal{C}$. Sono uno contenuto nell'altro? L'inclusione è stretta? Motivare le risposte.

Soluzione: Dato che l'istruzione $P(n)$ è codificabile nella macchina URM, chiaramente $\mathcal{C}^P \subseteq \mathcal{C}$. Più precisamente l'istruzione $I_j : P(n)$ si può sostituire con un salto alla seguente routine. Si indichi con q l'indice del primo registro non usato dal programma (quindi inizialmente a 0)

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

La routine controlla se il registro n contiene 0. In caso affermativo non c'è niente da fare. Altrimenti, con R_q che parte da 0 e R_{q+1} da 1, continua ad incrementare i due registri. Qunando R_{q+1} eguaglia R_n, avremo che R_q contiene il predecessore.

Più formalmente, si prova che $\mathcal{C}^p \subseteq \mathcal{C}$ dimostrando che, per ogni numero di argomenti k per ogni programma URM^p P 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 P nel programma P. Il caso base $h = 0$ è banale, dato che P con 0 istruzioni, è già un programma URM. Supposto vero il risultato per h dimostriamolo per $h + 1$. Il programma P contiene certamente almeno una istruzione P. Supponiamo che sia l'istruzione di indice j.

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

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

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

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

Vale anche l'inclusione opposta e quindi l'uguaglianza. Infatti l'istruzione $Z(n)$ può essere sostituita semplicemente da un'istruzione $T(q,n)$, dove q è un qualunque registro non usato dal programma e quindi a 0. Più precisamente, dato un programma URM P e un numero di argomenti fissato $k \in \mathbb{N}$, detto $q = \max\{\rho(P), k\} + 1$ l'indice del primo registro non usato e quindi inizialmente a 0, sostituendo in P ogni istruzione $Z(n)$ con l'istruzione $T(q,n)$, è un programma URM^p che calcola esattamente la stessa funzione.

Esercizio 2

Definire una funzione totale non calcolabile $f : \mathbb{N} \to \mathbb{N}$ tale che $img(f) = \{2^n \mid n \in \mathbb{N}\}$ (dove $img(f) = \{f(x) \mid x \in \mathbb{N}\}$).

Soluzione: Si procede per diagonalizzazione definendo:

$$
f(x) = \begin{cases} 2^{\varphi_x(x)} & \text{se } x \in W_x \\ 1 & \text{altrimenti} \end{cases}
$$

La funzione f è chiaramente totale. Inoltre

- f non calcolabile dato che per ogni $x \in \mathbb{N}$ risulta $f \neq \varphi_x$, ovvero f è diversa da tutte le funzioni calcolabili. Infatti, se $x \in W_x$ allora $f(x) = 2^{\varphi_x(x)} > \varphi_x(x)$, e mentre se $x \notin W_x$ si ha $f(x) = 1 \neq \varphi_x(x)$, dato che $\varphi_x(x) \uparrow$.
- Vale $img(f) = \{2^n \mid n \in \mathbb{N}\}$. Per definizione $img(f) \subseteq \{2^n \mid n \in \mathbb{N}\}$, dato che se $x \in W_x$ allora $f(x) = 2^{\varphi_x(x)}$ e se $x \notin W_x$ allora $f(x) = 1 = 2^0$. Vale anche l'inclusione

inversa. Infatti, dato un qualunque $n \in \mathbb{N}$, la funzione costante n è chiaramente calcolabile. Indicato con x un qualunque indice per tale funzione, ovvero $\varphi_x(y) = n$ per ogni y, dato che $x \in W_x = \mathbb{N}$, vale $f(x) = 2^{\varphi_x(x)} = 2^n$.

Esercizio 3

Enunciare il teorema smn ed utilizzarlo per dimostrare che esiste una funzione totale calcolabile $s : \mathbb{N} \to \mathbb{N}$ tale che per ogni $x \in \mathbb{N}$ valga $|W_x| = 2^x$ e $|E_x| = x + 1$.

Soluzione: Si definisce una funzione $g : \mathbb{N}^2 \to \mathbb{N}$ come segue:

$$g(x, y) = \begin{cases} \lfloor \log_2(y + 1) \rfloor & \text{se } y < 2^x \\ \uparrow & \text{altrimenti} \end{cases}$$

La funzione g è calcolabile. Infatti, $g(x, y)$, quando definito, è il massimo z tale che $2^z \leqslant y + 1$ e quindi il minimo tale che $2^{z+1} > y + 1$, ovvero

$$g(x, y) = \mu z.\, \overline{sg}(2^{z+1} \dot{-} (y + 1)) + \mu w.(y + 1 \dot{-} 2^x)$$

Pertanto per il teorema smn esiste $s : \mathbb{N} \to \mathbb{N}$ tale che per ogni $x, y \in \mathbb{N}$ risulta

$$g(x, y) = \varphi_{s(x)}(y)$$

e si dimostra che s che è la funzione cercata. Infatti

- $W_x = \{y \mid g(x, y) \downarrow\} = [0, 2^x - 1]$ e quindi $|W_x| = |[0, 2^x - 1]| = 2^x$
- $E_x = \{g(x, y) \mid 0 \leqslant y < 2^x\} = \{\lfloor \log_2(y + 1) \mid 0 \leqslant y < 2^x\} = [0, x]$ e quindi $|E_x| = |[0, x]| = x$.