Computabilità e Algoritmi - 03 Settembre 2015

Soluzioni Formali

Esercizio 1

Definire l'operazione di minimalizzazione illimitata e dimostrare che l'insieme C delle funzioni URM-calcolabili è chiuso rispetto a tale operazione.

Definizione di Minimalizzazione Illimitata: Data una funzione f: $\mathbb{N}^{(k+1)} \to \mathbb{N}$, la minimalizzazione illimitata $\mu y.f(\bar{x},y)$ è definita come:

$$\mu y.f(\bar{x},y) = \begin{cases} \min\{y : f(\bar{x},y) = 0\} & \text{se esiste tale } y \\ \uparrow & \text{altrimenti} \end{cases}$$

Teorema di Chiusura: Se f: $\mathbb{N}^{(k+1)} \to \mathbb{N}$ è URM-calcolabile, allora g: $\mathbb{N}^k \to \mathbb{N}$ definita da $g(\bar{x}) = \mu y.f(\bar{x},y)$ è URM-calcolabile.

Dimostrazione: Assumiamo che f sia calcolabile dal programma URM P. Costruiamo un programma Q che calcola $\mu y.f(\bar{x},y)$:

```
Input: x₁, ..., xₖ in registri R₁, ..., Rₖ
Algoritmo:
1. Inizializza Rₖ₊₁ ← 0 (contatore y)
2. LOOP:
   a. Copia x₁,...,xₖ,y nei registri appropriati
   b. Esegui il programma P per calcolare f(x̄,y)
   c. Se f(x̄,y) = 0, termina restituendo y
   d. Altrimenti, incrementa y e torna a LOOP
```

Implementazione URM formale: Sia m il numero di registri utilizzati da P. Il programma Q:

```
I₁: C(k+1)              // y ← 0
I₂: J(m+1, m+2, L)      // if f(x̄,y) = 0 goto L
I₃: S(k+1)              // y ← y+1
I₄: J(1, 1, 2)          // goto I₂ (loop)
L:  T(k+1, 1)           // output ← y
```

dove tra I_1 e I_2 inseriamo il programma P modificato per operare sui registri appropriati.

La correttezza segue dal fatto che:

- Se $\exists y$: $f(\bar{x},y) = 0$, l'algoritmo trova il minimo tale y e termina
- Se $\forall y$: $f(\bar{x},y) \neq 0$, l'algoritmo non termina ($\uparrow$)

Quindi C è chiuso rispetto alla minimalizzazione illimitata. □

Esercizio 2

Si dica che una funzione f: $\mathbb{N} \to \mathbb{N}$ è quasi costante se esiste un valore k $\in \mathbb{N}$ tale che l'insieme {x | f(x) ≠ k} è finito. Esiste una funzione quasi costante non calcolabile?

Risposta: Sì, esistono funzioni quasi costanti non calcolabili.

Esempio costruttivo: Definiamo f: $\mathbb{N} \to \mathbb{N}$ come:

f(x) = {1 se x $\in \bar{K}$ (x non è nel problema di halting)
{0 altrimenti

Verifica che f è quasi costante: Poiché K è infinito e $\bar{K}$ è infinito, ma uno dei due ha cardinalità maggiore, possiamo assumere che K sia "più piccolo" in senso asintottico. In realtà, costruiamo diversamente:

f(x) = {1 se x ≠ x_0 per qualche $x_0 \in \bar{K}$ fissato
{0 se x = x_0

Questa f è quasi costante con valore k = 1, poiché {x | f(x) ≠ 1} = {x_0} è finito.

Verifica che f non è calcolabile: Per decidere f(x), dovremmo decidere se x = x_0 dove $x_0 \in \bar{K}$. Ma scegliendo x_0 opportunamente (usando la costruzione diagonale), possiamo rendere questa decisione equivalente a risolvere il problema di halting.

Costruzione più diretta: Definiamo f tramite diagonalizzazione:

f(x) = {0 se x = e per qualche e specifico $\in \bar{K}$
{1 altrimenti

dove e è scelto in modo che determinare se x = e richieda di risolvere un problema indecidibile.

Dimostrazione formale di non calcolabilità: Supponiamo f calcolabile. Allora $\chi_{\bar{k}}$ sarebbe calcolabile (decidendo se x $\in \bar{K}$ tramite f), contraddicendo il fatto che $\bar{K}$ non è ricorsivo.

Quindi esistono funzioni quasi costanti non calcolabili. □

Esercizio 3

Studiare la ricorsività dell'insieme A = {x $\in \mathbb{N}$ | P $\subseteq W_x$}, dove P è un insieme finito fissato.

Caso P = ∅: Se P = ∅, allora A = $\mathbb{N}$ (poiché ∅ $\subseteq W_x$ per ogni x), quindi A è ricorsivo.

Caso P ≠ ∅: Assumiamo P = {p_1, p_2, ..., p_n} con n ≥ 1.

Saturazione: A è saturato: A = {x | $\varphi_x \in \mathcal{A}$} dove $\mathcal{A}$ = {f ∈ C : P ⊆ dom(f)}.

Non ricorsività per Rice:

- A ≠ ∅: la funzione identità ha dominio $\mathbb{N} \supseteq$ P
- A ≠ $\mathbb{N}$: la funzione sempre indefinita ha dominio ∅ $\not\supseteq$ P

Per il teorema di Rice, A non è ricorsivo.

Semidecidibilità di A: A è semidecidibile. Per verificare $P \subseteq W_x$, dobbiamo verificare che ogni elemento di P sia nel dominio di φ_x:

$sc_a(x) = 1(\mu w.\forall i \leq n.\ H(x, p_i, (w)_i))$

dove H(x,y,t) verifica se $\varphi_x(y) \downarrow$ in t passi.

Complemento $\bar{A}$: $\bar{A} = \{x \in \mathbb{N} \mid P \not\subseteq W_x\} = \{x \in \mathbb{N} \mid \exists p \in P.\ p \notin W_x\}$

$\bar{A}$ non è semidecidibile. Se lo fosse, con A semidecidibile, A sarebbe ricorsivo.

Conclusione:

- Se $P = \emptyset$: A è ricorsivo
- Se $P \neq \emptyset$: A è semidecidibile ma non ricorsivo, $\bar{A}$ non è semidecidibile □

Esercizio 4

Sia $f: \mathbb{N} \to \mathbb{N}$ una funzione totale calcolabile fissata. Studiare la ricorsività dell'insieme $B = \{x \in \mathbb{N} : f(x) \in E_x\}$.

Analisi: $B = \{x \in \mathbb{N} : f(x) \in E_x\}$ contiene gli indici x tali che f(x) appartiene all'immagine di φ_x.

Dipendenza da f: La ricorsività di B dipende crucialmente dalla funzione f.

Caso f costante: Se f(x) = c per ogni x, allora: $B = \{x \in \mathbb{N} : c \in E_x\}$

Questo insieme è saturato e per Rice è non ricorsivo (assumendo $c \neq 0$ per evitare casi degeneri).

Caso f = identità: Se f(x) = x, allora: $B = \{x \in \mathbb{N} : x \in E_x\}$

Questo è esattamente l'insieme studiato in esercizi precedenti, che è semidecidibile ma non ricorsivo.

Semidecidibilità generale: Per f generica totale calcolabile, B è sempre semidecidibile:

$sc_\beta(x) = 1(\mu w.\exists u,t.\ S(x, u, f(x), t))$

dove S(x,u,v,t) verifica se $\varphi_x(u) = v$ in t passi.

Non ricorsività generale: Per la maggior parte delle funzioni f non triviali, B non è ricorsivo. Dimostriamo $K \leq_m B$ per f appropriata.

Consideriamo f(x) = 0. Definiamo g(u,v):
g(u,v) = {0 se u ∈ K
{↑ altrimenti

Per SMN, esiste s tale che $\varphi_{s(u)}(v) = g(u,v)$.

Allora:

- Se $u \in K$: $0 \in E_{s(u)} = \{0\}$, quindi $s(u) \in B$
- Se $u \notin K$: $E_{s(u)} = \emptyset$, quindi $0 \notin E_{s(u)}$, quindi $s(u) \notin B$

Conclusione:

- B è sempre semidecidibile
- Per f non triviali, B è tipicamente non ricorsivo
- $\bar{B}$ è tipicamente non semidecidibile □

Esercizio 5

Enunciare il secondo teorema di ricorsione ed utilizzarlo per dimostrare che esiste un indice n ∈ ℕ tale che $\varphi_{p_n} = \varphi_n$, dove p_n è l'n-mo numero primo.

Secondo Teorema di Ricorsione (Kleene): Per ogni funzione f: ℕ → ℕ totale e computabile, esiste $e_0 \in \mathbb{N}$ tale che $\varphi_{e_0} = \varphi f(e_0)$.

Dimostrazione dell'esistenza dell'indice:

Sia π: ℕ → ℕ la funzione che calcola l'n-mo numero primo:
$\pi(n) = p_n$

La funzione π è totale e calcolabile (esistono algoritmi efficienti per calcolare numeri primi).

Applicando il secondo teorema di ricorsione alla funzione π, esiste n ∈ ℕ tale che:
$\varphi_n = \varphi\pi(n) = \varphi_{p_n}$

Interpretazione: Questo risultato mostra l'esistenza di un numero naturale n tale che il programma con indice n calcola esattamente la stessa funzione parziale del programma con indice p_n (l'n-mo numero primo).

In altre parole, esistono programmi il cui comportamento computazionale rimane invariato quando il loro indice viene trasformato nel corrispondente numero primo.

Nota sulle applicazioni: Questo tipo di risultato è fondamentale nella teoria della ricorsione per dimostrare proprietà di autoriflessione dei sistemi computazionali e per costruire programmi con specifiche proprietà autoreferenziali. □