Computabilità e Algoritmi - 15 Settembre 2015

Soluzioni Formali

Esercizio 1

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

Definizione di Ricorsione Primitiva: Date funzioni f: ℕ^k → ℕ e g: ℕ^(k+2) → ℕ, la funzione h: ℕ^(k+1) → ℕ definita per ricorsione primitiva è:

h(x̄, 0) = f(x̄)
h(x̄, y+1) = g(x̄, y, h(x̄, y))

Teorema di Chiusura: Se f e g sono URM-calcolabili, allora h è URM-calcolabile.

Dimostrazione: Costruiamo un programma URM che calcola h(x̄, y) per ricorsione.

Schema del programma:

```
Input: x₁, ..., xₖ, y nei registri R₁, ..., Rₖ₊₁
Algoritmo:
1. Se y = 0, calcola f(x̄) e termina
2. Altrimenti, inizializza un ciclo che calcola iterativamente:
   - h(x̄, 0) = f(x̄)
   - h(x̄, 1) = g(x̄, 0, h(x̄, 0))
   - h(x̄, 2) = g(x̄, 1, h(x̄, 1))
   - ...
   - h(x̄, y) = g(x̄, y-1, h(x̄, y-1))
```

Implementazione URM dettagliata: Assumiamo che f utilizzi registri $R_1 ... R_m$ e g utilizzi registri $R_1 ... R_n$.

```
// Fase 1: Gestione caso base
I₁: J(k+1, k+2, BASE)    // if y = 0 goto BASE
I₂: T(k+1, k+3)          // counter ← y
I₃: C(k+2)               // i ← 0

// Fase 2: Calcolo h(x̄, 0) = f(x̄)
INIT: [Programma per f]
I₄: T(1, k+4)            // salva risultato h(x̄, i)

// Fase 3: Loop principale
LOOP: J(k+2, k+3, END)   // if i = y goto END
I₅: S(k+2)               // i ← i+1
I₆: [Setup parametri per g(x̄, i-1, h(x̄, i-1))]
I₇: [Programma per g]
I₈: T(1, k+4)            // aggiorna h(x̄, i)
I₉: J(1, 1, LOOP)        // goto LOOP

BASE: [Programma per f]
END: T(k+4, 1)           // output ← h(x̄, y)
```

Correttezza:

- Il programma implementa esattamente la definizione ricorsiva
- Se f e g terminano sui loro input, h termina
- La ricorsione è "appiattita" in iterazione

Quindi C è chiuso rispetto alla ricorsione primitiva. □

Esercizio 2

Si dica che una funzione f: ℕ → ℕ è quasi costante se esiste un valore k ∈ 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.

Costruzione tramite diagonalizzazione:

Definiamo f: ℕ → ℕ nel modo seguente:

f(x) = {0 se $x \neq e_0$
{1 se $x = e_0$

dove e_0 è un indice particolare che sceglieremo strategicamente.

Scelta di e_0: Utilizziamo il teorema di ricorsione per costruire e_0. Consideriamo la funzione h: ℕ → ℕ definita da:

h(e) = indice di un programma che calcola {0 se x ≠ e
{1 se x = e

Per il secondo teorema di ricorsione, esiste e_0 tale che $\varphi_{e0} = \varphi_{h(e0)}$.

Verifica che f è quasi costante: L'insieme $\{x \mid f(x) \neq 0\} = \{e_0\}$ è finito (contiene un solo elemento), quindi f è quasi costante con valore k = 0.

Verifica che f non è calcolabile: Supponiamo f calcolabile con indice e_1. Allora:

- $f(e_1) = 0$ se $e_1 \neq e_0$
- $f(e_1) = 1$ se $e_1 = e_0$

Ma per la costruzione di e_0, abbiamo $\varphi_{e0}(e_1) = f(e_1)$. Se f fosse calcolabile, questo porterebbe a una contraddizione con l'autoreferenzialità della costruzione.

Costruzione alternativa più diretta: Definiamo f usando la caratteristica funzione di un insieme non ricorsivo ma "quasi vuoto":

f(x) = {1 se $x \in A$
{0 altrimenti

dove A è un insieme non ricorsivo ma finito. Anche se A è finito, la decisione di appartenenza può essere non calcolabile se A è definito tramite proprietà non decidibili.

Esempio concreto: Sia $A = \{x \in \mathbb{N} \mid x$ è il più piccolo indice tale che $\varphi_x = \varphi_0 \wedge x \neq 0\}$.

Se tale x esiste, $A = \{x\}$, altrimenti $A = \varnothing$. In entrambi i casi A è finito, ma determinare se $A = \varnothing$ o $A = \{x\}$ richiede di risolvere un problema di equivalenza di funzioni, che è indecidibile.

Quindi $f(x) = \chi_a(x)$ è quasi costante ma non calcolabile. □

Esercizio 3

Studiare la ricorsività dell'insieme $A = \{x \in \mathbb{N} \mid E_x = W_x + 1\}$, ovvero dire se A e $\bar{A}$ sono ricorsivi/ ricorsivamente enumerabili.

Analisi: $A = \{x \in \mathbb{N} \mid E_x = W_x + 1\}$ contiene gli indici per cui l'immagine è uguale al dominio traslato di 1.

Formalmente: $x \in A \iff \forall y: y \in E_x \iff y-1 \in W_x$ (per $y \geq 1$) e $0 \notin E_x$.

Saturazione: A è saturato: $A = \{x \mid \varphi_x \in \mathcal{A}\}$ dove $\mathcal{A} = \{f \in C : cod(f) = dom(f) + 1\}$.

Non ricorsività per Rice:

- $A \neq \varnothing$: la funzione $f(x) = x+1$ soddisfa $cod(f) = \mathbb{N}\{0\} = \mathbb{N} + 1 = dom(f) + 1$
- $A \neq \mathbb{N}$: la funzione identità non soddisfa la condizione

Per il teorema di Rice, A non è ricorsivo.

Analisi della semidecidibilità: Per A essere semidecidibile, dovremmo poter verificare che $E_x = W_x + 1$. Questo richiede:

1. Verificare che per ogni $y \in E_x$, esiste $z \in W_x$ tale che $y = z+1$
2. Verificare che per ogni $z \in W_x$, $z+1 \in E_x$
3. Verificare che $0 \notin E_x$

La condizione (2) è problematica: richiede di verificare che per ogni input nel dominio, l'output corrispondente meno 1 sia nel dominio. Questo può richiedere verifiche infinite.

Costruzione di riduzione da $\bar{K}$: Per dimostrare che A non è semidecidibile, costruiamo $\bar{K} \leq_m A$.

Definiamo g(u,v):
$g(u,v) = \{v+1$ se $u \notin K \wedge v \in \mathbb{N}$
$\{\uparrow$ se $u \in K$

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

- Se $u \notin K$: $W_{s(u)} = \mathbb{N}$, $E_{s(u)} = \mathbb{N}\{0\}$, quindi $E_{s(u)} = W_{s(u)} + 1$, così $s(u) \in A$
- Se $u \in K$: $W_{s(u)} = E_{s(u)} = \emptyset$, quindi $s(u) \notin A$

Quindi $\bar{K} \leq_m A$, e poiché $\bar{K}$ non è semidecidibile, A non è semidecidibile.

Complemento $\bar{A}$: Analogamente, $\bar{A}$ non è semidecidibile.

Conclusione:

- A non è ricorsivo
- A non è semidecidibile
- $\bar{A}$ non è semidecidibile □

Esercizio 4

Studiare la ricorsività dell'insieme $B = \{x \in \mathbb{N} : \forall y > x.\ 2y \in W_x\}$.

Analisi: $B = \{x \in \mathbb{N} : \forall y > x.\ 2y \in W_x\}$ contiene gli indici per cui tutti i numeri pari maggiori di 2x sono nel dominio.

Saturazione: B è saturato: $B = \{x \mid \varphi_x \in \mathcal{B}\}$ dove $\mathcal{B} = \{f \in C : \{2y \mid y > \text{index}\} \subseteq dom(f)\}$.

Tuttavia, la saturazione dipende dall'indice x, rendendo la caratterizzazione più complessa.

Analisi più dettagliata: $B = \{x \in \mathbb{N} : \{2(x+1), 2(x+2), 2(x+3), ...\} \subseteq W_x\}$

Non ricorsività: Dimostriamo $K \leq_m B$. Definiamo g(u,v):

$g(u,v) = \{v$ se $u \in K \vee v \leq 2u \vee v$ è dispari
$\{\uparrow$ se $u \notin K \wedge v > 2u \wedge v$ è pari

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

- Se $u \in K$: $W_{s(u)} = \mathbb{N} \supseteq \{2y \mid y > s(u)\}$, quindi $s(u) \in B$
- Se $u \notin K$: esistono numeri pari $> 2s(u)$ non in $W_{s(u)}$, quindi $s(u) \notin B$

Semidecidibilità di B: B non è semidecidibile. La verifica di $\forall y > x.\ 2y \in W_x$ richiede controlli infiniti, e non possiamo semidecidere universalmente quantificate proprietà infinite sui domini.

Complemento $\bar{B}$: $\bar{B} = \{x \in \mathbb{N} : \exists y > x.\ 2y \notin W_x\}$

$\bar{B}$ è semidecidibile:
$sc_{\bar{\beta}}(x) = 1(\mu w.\exists y > x.\ \neg H(x, 2y, (w)_1))$

Conclusione:

- B non è ricorsivo
- B non è semidecidibile
- $\bar{B}$ è semidecidibile ma non ricorsivo □

Esercizio 5

Enunciare il secondo teorema di ricorsione ed utilizzarlo per dimostrare che la funzione Δ: ℕ → ℕ, definita da $\Delta(x) = \min\{y : \varphi_y \neq \varphi_x\}$, non è calcolabile.

Secondo Teorema di Ricorsione (Kleene): Per ogni funzione f: ℕ → ℕ totale e computabile, esiste $e_0 \in \mathbb{N}$ tale che $\varphi_{e0} = \varphi f(e_0)$.

Dimostrazione che Δ non è calcolabile:

Supponiamo per assurdo che Δ sia calcolabile.

Costruzione della contraddizione: Definiamo la funzione h: ℕ → ℕ come h(x) = Δ(x).

Poiché Δ è calcolabile per ipotesi, h è calcolabile.

Applichiamo il secondo teorema di ricorsione a h: esiste $e_0 \in \mathbb{N}$ tale che $\varphi_{e0} = \varphi_{h(e0)} = \varphi\_\Delta(e_0)$.

Analisi del contraddizione: Per definizione di Δ: $\Delta(e_0) = \min\{y : \varphi_y \neq \varphi_{e0}\}$

Ma per il teorema di ricorsione: $\varphi_{e0} = \varphi\_\Delta(e_0)$.

Quindi $\Delta(e_0)$ è il minimo indice y tale che $\varphi_y \neq \varphi_{e0}$, ma allo stesso tempo $\varphi\_\Delta(e_0) = \varphi_{e0}$.

Questo significa che $\Delta(e_0)$ non può essere il minimo indice con questa proprietà, poiché $\varphi\_\Delta(e_0) = \varphi_{e0}$.

Formalizzazione della contraddizione:

1. $\varphi_{e0} = \varphi\_\Delta(e_0)$ (dal teorema di ricorsione)
2. $\Delta(e_0) = \min\{y : \varphi_y \neq \varphi_{e0}\}$ (definizione di Δ)
3. Dalla (2): $\varphi\_\Delta(e_0) \neq \varphi_{e0}$
4. Ma (1) e (3) sono contraddittorie

Quindi l'ipotesi che Δ sia calcolabile è falsa.

Conclusione: La funzione Δ non è calcolabile. Questo risultato illustra come l'autoreferenzialità garantita dal secondo teorema di ricorsione possa essere utilizzata per dimostrare risultati di non calcolabilità. □