Computabilità e Algoritmi - 15 Luglio 2013

Soluzioni Formali

Esercizio 1

Problema: Considerare la sottoclasse dei programmi URM nei quali, se l'i-ma istruzione è una istruzione di salto $J(m, n, t)$, allora $t > i$. Dimostrare che le funzioni calcolabili dai programmi in tale sottoclasse sono tutte totali.

Soluzione:

Consideriamo la sottoclasse di programmi URM con la restrizione sui salti: se l'istruzione I_i è $J(m, n, t)$, allora $t > i$.

Teorema: Tutte le funzioni calcolabili da programmi in questa sottoclasse sono totali.

Dimostrazione:

Sia P un programma in questa sottoclasse con lunghezza $l(P)$.

Lemma: Per ogni computazione di P, l'indice dell'istruzione da eseguire al passo $t+1$ è maggiore dell'indice dell'istruzione eseguita al passo t.

Dimostrazione del Lemma: Procediamo per induzione sul numero di passi di computazione.

Caso base ($t = 0$): L'istruzione iniziale ha indice 1.

Passo induttivo: Supponiamo che al passo t l'istruzione da eseguire abbia indice i. Consideriamo i casi:

1. **Istruzione aritmetica (Z(n), S(n), T(m,n)):** La prossima istruzione ha indice $i+1 > i$.
2. **Istruzione di salto J(m,n,j):**
   - Se $r_m = r_n$: la prossima istruzione ha indice $j > i$ (per ipotesi della sottoclasse)
   - Se $r_m \neq r_n$: la prossima istruzione ha indice $i+1 > i$

In entrambi i casi, l'indice della prossima istruzione è strettamente maggiore dell'indice corrente.

Conseguenza del Lemma: Poiché l'indice dell'istruzione cresce strettamente ad ogni passo e è limitato superiormente da $l(P)$, la computazione deve terminare entro al più $l(P)$ passi.

Formalizzazione: Al passo t, se l'istruzione eseguita ha indice i_t, allora:

- $i_0 = 1$
- $i_{t+1} > i_t$ per ogni $t \geq 0$
- $i_t \leq l(P)$ per ogni t (altrimenti il programma termina)

Poiché la sequenza $\{i_t\}$ è strettamente crescente e limitata, deve essere finita.

Conclusione sulla totalità: Ogni computazione termina in tempo finito, quindi ogni funzione calcolabile da un programma in questa sottoclasse è totale. ∎

Osservazione: La restrizione $t > i$ impedisce i "loop backwards" che sono la causa principale della non-terminazione nei programmi URM.

Esercizio 2

Problema: Dimostrare che un insieme A è ricorsivo se e solo se esistono due funzioni totali calcolabili f, g : $\mathbb{N} \to \mathbb{N}$ tale che per ogni $x \in \mathbb{N}$: $x \in A$ se e solo se $f(x) > g(x)$.

Soluzione:

Teorema: $A \subseteq \mathbb{N}$ è ricorsivo $\Longleftrightarrow \exists f, g: \mathbb{N} \to \mathbb{N}$ totali calcolabili tali che $\forall x \in \mathbb{N}: x \in A \Longleftrightarrow f(x) > g(x)$.

Dimostrazione:

Direzione (⇐): Se esistono f, g, allora A è ricorsivo

Supponiamo che esistano f, g: $\mathbb{N} \to \mathbb{N}$ totali calcolabili tali che:
$\forall x \in \mathbb{N}: x \in A \Longleftrightarrow f(x) > g(x)$

Definiamo la funzione caratteristica di A:

```
χ_A(x) = {
    1    se f(x) > g(x)
    0    se f(x) ≤ g(x)
}
```

Equivalentemente: $\chi\_A(x) = sg(f(x) - g(x))$

dove sg è la funzione sign: $sg(0) = 0$, $sg(n) = 1$ per $n > 0$.

La funzione sg è primitiva ricorsiva, quindi calcolabile.
Poiché f e g sono calcolabili, per composizione χ_A è calcolabile.
Quindi A è ricorsivo.

Direzione (⇒): Se A è ricorsivo, allora esistono f, g

Supponiamo A ricorsivo. Allora χ_A è calcolabile.

Costruzione 1: Definiamo:

- $f(x) = \chi\_A(x)$
- $g(x) = 0$

Allora:

- Se $x \in A$: $f(x) = 1 > 0 = g(x)$
- Se $x \notin A$: $f(x) = 0 \leq 0 = g(x)$

Quindi $x \in A \iff f(x) > g(x)$.

Costruzione 2 (più interessante): Definiamo:

- $f(x) = \chi_A(x) + 1$
- $g(x) = 1$

Allora:

- Se $x \in A$: $f(x) = 2 > 1 = g(x)$
- Se $x \notin A$: $f(x) = 1 \leq 1 = g(x)$

Costruzione 3 (alternativa simmetrica): Definiamo:

- $f(x) = \chi_A(x)$
- $g(x) = \chi_{\bar{A}}(x)$

Poiché A è ricorsivo, anche $\bar{A}$ è ricorsivo, quindi $\chi_{\bar{A}}$ è calcolabile.

Allora:

- Se $x \in A$: $f(x) = 1$, $g(x) = 0$, quindi $f(x) > g(x)$
- Se $x \notin A$: $f(x) = 0$, $g(x) = 1$, quindi $f(x) \leq g(x)$

Verifica delle proprietà: In tutte le costruzioni, f e g sono totali (definite per ogni input) e calcolabili (composizione di funzioni calcolabili).

Conclusione: A è ricorsivo $\iff \exists f, g$ totali calcolabili tali che $x \in A \iff f(x) > g(x)$. ∎

Osservazione: Questo teorema mostra che i predicati ricorsivi coincidono esattamente con le disuguaglianze calcolabili.

Esercizio 3

Problema: Studiare la ricorsività dell'insieme $A = \{x : \forall y.\ \text{if } y + x \in W_x \text{ then } y \leq \varphi_x(y + x)\}$, ovvero dire se A e $\bar{A}$ sono ricorsivi/ricorsivamente enumerabili.

Soluzione:

A contiene gli indici x tali che per ogni y, se $y + x$ è nel dominio di φ_x, allora $y \leq \varphi_x(y + x)$.

Riscrittura della condizione: $x \in A \iff \forall y \in \mathbb{N}.\ [(y + x \in W_x) \rightarrow (y \leq \varphi_x(y + x))]$

Analisi della struttura:

A è un insieme saturo, poiché la proprietà dipende solo dal comportamento di φ_x.

Ricorsività:

A non è ricorsivo. La condizione coinvolge una quantificazione universale su y, che generalmente rende l'insieme non ricorsivo.

Dimostrazione via Rice: A può essere espresso come $A = \{x \in \mathbb{N} : \varphi_x \in \mathcal{A}\}$, dove: $\mathcal{A} = \{f \in \mathcal{C} : \forall y \in \mathbb{N}. [(\exists x.\ y + x \in dom(f)) \rightarrow (y \leq f(y + x))]\}$

Questa è una proprietà non banale delle funzioni, quindi per il teorema di Rice, A non è ricorsivo.

Enumerabilità ricorsiva di A:

A non è r.e. La condizione universale "$\forall y$" non può essere verificata in tempo finito.

Dimostrazione: Supponiamo per assurdo che A sia r.e. Allora potremmo semi-decidere se φ_x soddisfa la proprietà universale. Questo contrasterebbe con il fatto che le proprietà universali delle funzioni calcolabili generalmente non sono semidecidibili.

Enumerabilità ricorsiva di Ā:

$\bar{A} = \{x \in \mathbb{N} : \exists y \in \mathbb{N}. [(y + x \in W_x) \wedge (y > \varphi_x(y + x))]\}$

Ā è r.e. Possiamo scrivere la funzione semicaratteristica:

```
sc_Ā(x) = 1(μt. ∃y ≤ t. [H(x, y+x, t) ∧ S(x, y+x, v, t) ∧ y > v] per qualche v ≤ t)
```

Più precisamente:

```
sc_Ā(x) = 1(μt. ∃y ≤ t. ∃v ≤ t. [S(x, y+x, v, t) ∧ y > v])
```

Questa funzione cerca un testimone y e un tempo t tali che $\varphi_x(y + x)$ converge a un valore $v < y$.

Analisi più dettagliata:

La condizione di A è relativamente restrittiva. Richiede che φ_x si "comporti bene" nel senso che l'output sia sempre maggiore o uguale all'indice dell'input relativo a x.

Esempi:

- Se $\varphi_x(z) = z$ per ogni $z \in W_x$, allora per $y + x \in W_x$ abbiamo $\varphi_x(y + x) = y + x \geq y$, quindi $x \in A$
- Se $\varphi_x(z) = 0$ per ogni $z \in W_x$ e $0 \in W_x$, allora $y = 0$, $\varphi_x(0) = 0$, e $0 \leq 0$, ma per $y > 0$, se $y \in W_x$ allora $\varphi_x(y) = 0 < y$, quindi $x \notin A$

Conclusione: A non è ricorsivo, A non è r.e., Ā è r.e. ∎

Esercizio 4

Problema: Sia f una funzione calcolabile totale. Studiare la ricorsività dell'insieme $B_f = \{x \in \mathbb{N} : \varphi_x(y) = f(y) \text{ per infiniti } y\}$, ovvero dire se B_f e $\bar{B}_f$ sono ricorsivi/ricorsivamente enumerabili.

Soluzione:

B_f contiene gli indici x tali che φ_x coincide con f su un insieme infinito di punti.

Analisi della struttura:

B_f è un insieme saturo, poiché può essere espresso come $B_f = \{x \in \mathbb{N} : \varphi_x \in \mathcal{B}_f\}$, dove:
$\mathcal{B}_f = \{g \in \mathcal{C} : |\{y \in \mathbb{N} : g(y) = f(y)\}| = \infty\}$

Ricorsività:

Per il teorema di Rice, poiché B_f è saturo, dobbiamo verificare se $B_f = \emptyset, \mathbb{N}$ o né l'uno né l'altro.

- $B_f \neq \emptyset$: La funzione identica a f appartiene a $\mathcal{B}_f$, quindi un suo indice appartiene a B_f
- $B_f \neq \mathbb{N}$: La funzione sempre indefinita non coincide con f su infiniti punti (anzi, su nessun punto), quindi un suo indice non appartiene a B_f

Per il teorema di Rice, B_f non è ricorsivo.

Enumerabilità ricorsiva di B_f:

B_f non è r.e. Utilizziamo il teorema di Rice-Shapiro.

Dimostrazione: Consideriamo f stesso. Abbiamo $f \in \mathcal{B}_f$.

Consideriamo una qualsiasi funzione finita $\theta \subseteq f$. Allora θ coincide con f solo su un insieme finito ($dom(\theta)$), quindi $\theta \notin \mathcal{B}_f$.

Per Rice-Shapiro, esiste $f \in \mathcal{B}_f$ tale che $\forall \theta \subseteq f$ finita, $\theta \notin \mathcal{B}_f$, quindi B_f non è r.e.

Enumerabilità ricorsiva di $\bar{B}_f$:

$\bar{B}_f = \{x \in \mathbb{N} : \varphi_x(y) = f(y) \text{ per al più finiti } y\}$

$\bar{B}_f$ non è r.e. in generale.

Dimostrazione: La condizione "per al più finiti y" è una proprietà cofinita che generalmente non è semidecidibile.

Tuttavia, l'analisi dipende dalla specifica funzione f:

Caso speciale: f funzione costante Se $f(y) = c$ per ogni $y \in \mathbb{N}$, allora:

- $B_f = \{x : \varphi_x(y) = c \text{ per infiniti } y\}$
- $\bar{B}_f = \{x : \varphi_x(y) = c \text{ per al più finiti } y\}$

In questo caso, $\bar{B}_f$ potrebbe essere r.e. se possiamo caratterizzarlo diversamente.

Caso generale: Per una f arbitraria, $\bar{B}_f$ tipicamente non è r.e. perché la verifica della finitezza dell'insieme di accordo non è semidecidibile.

Costruzione di una riduzione: Se $\bar{B}_f$ fosse r.e., potremmo potenzialmente costruire riduzioni da problemi non r.e., ottenendo contraddizioni.

Osservazione tecnica: L'insieme $\{y : \varphi_x(y) = f(y)\}$ per un x fissato non è necessariamente decidibile, rendendo difficile verificare se è finito o infinito.

Conclusione: B_f non è ricorsivo, B_f non è r.e., $\bar{B}_f$ non è r.e. ∎

Esercizio 5

Problema: Enunciare il Secondo Teorema di Ricorsione ed utilizzarlo per dimostrare che esiste $n \in \mathbb{N}$ tale che $W_n = E_n = \{x \cdot n : x \in \mathbb{N}\}$.

Soluzione:

Enunciato del Secondo Teorema di Ricorsione: Per ogni funzione $f: \mathbb{N} \to \mathbb{N}$ totale e calcolabile, esiste $e \in \mathbb{N}$ tale che $\varphi_e = \varphi_{f(e)}$.

Dimostrazione dell'esistenza di n:

Vogliamo trovare n tale che $W_n = E_n = \{x \cdot n : x \in \mathbb{N}\} = \{0, n, 2n, 3n, ...\}$.

Costruzione della funzione: Definiamo $g: \mathbb{N}^2 \to \mathbb{N}$:

```
g(k, y) = {
    y    se ∃x ∈ ℕ. y = x · k ∧ k > 0
    ↑    altrimenti
}
```

Verifica che g è calcolabile:

```
g(k, y) = y · sg(k) · sg(μx ≤ y. |y - x · k|)
```

dove $\mu x \leq y. |y - x \cdot k|$ cerca se esiste $x \leq y$ tale che $x \cdot k = y$.

Più precisamente:

```
g(k, y) = {
    y · 1(μx ≤ y. |y - x · k|)    se k > 0
    ↑                              se k = 0
}
```

Proprietà di g: Per $k > 0$:

- $W_{s(k)} = \{x \cdot k : x \in \mathbb{N}\}$ (dominio)
- $E_{s(k)} = \{x \cdot k : x \in \mathbb{N}\}$ (codominio, perché $g(k, x \cdot k) = x \cdot k$)

Quindi $W_{s(k)} = E_{s(k)} = \{x \cdot k : x \in \mathbb{N}\}$.

Applicazione del teorema smn: Per il teorema smn, esiste s: ℕ → ℕ totale calcolabile tale che: $\varphi_{s(k)}(y) = g(k, y)$

Applicazione del Secondo Teorema di Ricorsione: Per il Secondo Teorema di Ricorsione applicato alla funzione s, esiste $n \in \mathbb{N}$ tale che: $\varphi_n = \varphi_{s(n)}$

Verifica della proprietà richiesta: Da $\varphi_n = \varphi_{s(n)}$, abbiamo:

- $W_n = W_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$
- $E_n = E_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$

Caso n = 0: Se n = 0, allora $\{x \cdot 0 : x \in \mathbb{N}\} = \{0\}$. Ma per la definizione di g, $g(0, y) \uparrow$ per ogni y, quindi $W_{s(0)} = \emptyset \neq \{0\}$.

Gestione del caso n = 0: Per evitare problemi con n = 0, modifichiamo g:

```
g(k, y) = {
    y    se y = 0 ∧ k = 0
    y    se ∃x ∈ ℕ. y = x · k ∧ k > 0
    ↑    altrimenti
}
```

Allora:

- Per k = 0: $W_{s(0)} = E_{s(0)} = \{0\} = \{x \cdot 0 : x \in \mathbb{N}\}$
- Per k > 0: $W_{s(k)} = E_{s(k)} = \{x \cdot k : x \in \mathbb{N}\}$

Conclusione: Esiste $n \in \mathbb{N}$ tale che $W_n = E_n = \{x \cdot n : x \in \mathbb{N}\}$. ∎