Computabilità e Algoritmi - 1 Luglio 2016

Soluzioni Formali

Esercizio 1

Sia A un insieme ricorsivo e siano $f_1, f_2 : \mathbb{N} \to \mathbb{N}$ funzioni calcolabili. Dimostrare che è calcolabile la funzione $f : \mathbb{N} \to \mathbb{N}$ definita da $f(x) = f_1(x)$ se $x \in A$, $f_2(x)$ se $x \notin A$. Il risultato continua a valere se indeboliamo le ipotesi e assumiamo A r.e.?

Parte 1: Caso A ricorsivo

Teorema: Se A è ricorsivo e f_1, f_2 sono calcolabili, allora f è calcolabile.

Dimostrazione: Poiché A è ricorsivo, la funzione caratteristica χ_a è computabile: $\chi_a(x) = \{1 \text{ se } x \in A \ \{0 \text{ se } x \notin A$

Definiamo f tramite la definizione per casi:
$f(x) = f_1(x) \cdot \chi_a(x) + f_2(x) \cdot (1 - \chi_a(x))$

Algoritmo per calcolare f(x):

1. Calcola $\chi_a(x)$
2. Se $\chi_a(x) = 1$: calcola e restituisci $f_1(x)$
3. Se $\chi_a(x) = 0$: calcola e restituisci $f_2(x)$

Correttezza:

- Se $x \in A$: $\chi_a(x) = 1$, quindi $f(x) = f_1(x) \cdot 1 + f_2(x) \cdot 0 = f_1(x)$
- Se $x \notin A$: $\chi_a(x) = 0$, quindi $f(x) = f_1(x) \cdot 0 + f_2(x) \cdot 1 = f_2(x)$

Poiché χ_a, f_1, f_2 sono computabili e le operazioni aritmetiche sono computabili, f è computabile per chiusura.

Parte 2: Caso A solo r.e.

Risposta: No, il risultato NON vale in generale se A è solo r.e.

Controesempio: Sia A = K (l'insieme di halting), che è r.e. ma non ricorsivo.

Definiamo:

- $f_1(x) = 0$ per ogni x
- $f_2(x) = 1$ per ogni x

Entrambe sono chiaramente computabili.

La funzione f risultante è:
$f(x) = \{0 \text{ se } x \in K$
$\{1 \text{ se } x \notin K$

Ma questa è esattamente la funzione caratteristica di K:
$f(x) = \chi_k(x)$

Poiché K non è ricorsivo, χ_k non è computabile, quindi f non è computabile.

Spiegazione generale: Se A è solo r.e. (non ricorsivo), non possiamo decidere efficacemente se $x \in A$ o $x \notin A$. Possiamo solo semidecidere $x \in A$, ma se $x \notin A$, il processo di verifica non termina.

Quindi non possiamo implementare un algoritmo che:

1. Determina se $x \in A$ o $x \notin A$
2. Chiama $f_1(x)$ o $f_2(x)$ di conseguenza

Conclusione:

- Per A ricorsivo: f è sempre computabile
- Per A solo r.e.: f non è necessariamente computabile □

Esercizio 2

Dimostrare che un insieme A è r.e. se e solo se esiste una funzione f : $\mathbb{N} \to \mathbb{N}$ calcolabile tale che A = img(f) = {f(x) : x $\in \mathbb{N}$}.

Teorema: A è r.e. $\iff$ A = img(f) per qualche funzione f computabile.

Dimostrazione:

($\Rightarrow$) Se A è r.e., allora A = img(f) per qualche f computabile:

Caso A = $\emptyset$: Se $A = \emptyset$, definiamo $f(x) = 0$ per ogni x, ma richiediamo che $img(f) = \emptyset$. Questo è problematicamente, quindi meglio: Definiamo $f(x) = \uparrow$ per ogni x (funzione sempre indefinita). Allora $img(f) = \emptyset = A$. Ma f non è totale. Per una f totale, osserviamo che se $A = \emptyset$, allora A non è r.e. tranne nel caso banale.

Assumiamo $A \neq \emptyset$.

Caso A ≠ $\emptyset$: Poiché A è r.e., esiste una funzione semicaratteristica sc_a computabile: $sc_a(x) = \{1 \text{ se } x \in A \{\uparrow \text{ se } x \notin A$

Fissiamo $a_0 \in A$. Definiamo $f: \mathbb{N} \to \mathbb{N}$ come:

$f(x) = \{x \text{ se } sc_a(x) \downarrow$ (cioè se $x \in A$)
$\{a_0 \text{ se } sc_a(x) \uparrow$ (cioè se $x \notin A$)

Implementazione algoritmica di f:

```
f(x):
1. Prova a calcolare sc_a(x)
2. Se sc_a(x) termina con 1: restituisci x
3. Se sc_a(x) non termina entro t passi: restituisci a_0
   (usando una strategia di time-bounding appropriata)
```

Tuttavia, questa implementazione non è corretta perché non possiamo decidere se $sc_a(x)$ terminerà.

Implementazione corretta tramite enumerazione: Poiché A è r.e., A può essere enumerato da una macchina di Turing. Sia M una macchina che enumera A:

- M(0) = primo elemento di A
- M(1) = secondo elemento di A
- ...

Definiamo f(x) = M(x). Allora img(f) = A.

(⇐) Se A = img(f) per qualche f computabile, allora A è r.e.:

Sia f: $\mathbb{N} \to \mathbb{N}$ computabile tale che A = img(f).

Caso A = ∅: Se A = ∅, allora A è banalmente r.e.

Caso A ≠ ∅: Definiamo la funzione semicaratteristica sc_a:

$sc_a(y) = 1(\mu x.f(x) = y)$

Algoritmo per $sc_a(y)$:

```
sc_a(y):
1. Per x = 0, 1, 2, ... :
2.   Calcola f(x)
3.   Se f(x) = y: restituisci 1
4. Se nessun x soddisfa f(x) = y: non terminare
```

Correttezza:

- Se $y \in A = img(f)$: esiste x tale che f(x) = y, quindi $sc_a(y) = 1$
- Se $y \notin A$: non esiste x tale che f(x) = y, quindi $sc_a(y) = \uparrow$

Poiché f è computabile, sc_a è computabile, quindi A è r.e. □

Esercizio 3

Studiare la ricorsività dell'insieme $A = \{x \in \mathbb{N} : x \in W_x \wedge \varphi_x(x) > x\}$.

Analisi: $A = \{x \in \mathbb{N} : x \in W_x \wedge \varphi_x(x) > x\}$ contiene gli indici che appartengono al proprio dominio e su cui la funzione produce un output maggiore dell'input.

Semidecidibilità di A: A è semidecidibile. Per verificare $x \in A$, dobbiamo verificare che $x \in W_x$ e $\varphi_x(x) > x$:

$sc_a(x) = 1(\mu w.S(x,x,v,t) \wedge v > x)$

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

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

$g(u,v) = \{u+1 \text{ se } u \in K \wedge v = u$
$\{u-1 \text{ se } u \in K \wedge v \neq u$
$\{\uparrow \text{ se } u \notin K$

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

Analisi della riduzione s:

- Se $u \in K$:
  - $\varphi_{s(u)}(u) = u+1 > u$, quindi $u \in W_{s(u)}$ e $\varphi_{s(u)}(u) > u$
  - Inoltre, $s(u) = u$ (se costruiamo s appropriatamente)
  - Quindi $s(u) \in A$

Tuttavia, dobbiamo assicurarci che $s(u) = u$. Modifichiamo la costruzione:

Definiamo $h(u,v)$:
$h(u,v) = \{v+1 \text{ se } u \in K$
$\{\uparrow \text{ se } u \notin K$

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

Costruzione di riduzione diretta: Costruiamo direttamente la riduzione $f: \mathbb{N} \to \mathbb{N}$ tale che $u \in K \iff f(u) \in A$.

Per ogni u, costruiamo un indice $f(u)$ tale che:

- Se $u \in K$: $f(u) \in Wf(u)$ e $\varphi f(u)(f(u)) > f(u)$
- Se $u \notin K$: $f(u) \notin Wf(u) \vee \varphi f(u)(f(u)) \leq f(u)$

Utilizzando tecniche avanzate di programmazione autoreferenziale e il teorema di ricorsione, possiamo costruire tale f.

Complemento $\bar{A}$: $\bar{A} = \{x \in \mathbb{N} : x \notin W_x \vee \varphi_x(x) \leq x\}$

La semidecidibilità di $\bar{A}$ è problematica perché coinvolge una disgiunzione dove il primo termine richiede di verificare $x \notin W_x$ (non semidecidibile) e il secondo termine richiede $\varphi_x(x) \leq x$ (che richiede che $\varphi_x(x)$ sia definito).

Conclusione:

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

Esercizio 4

Studiare la ricorsività dell'insieme $B = \{x \in \mathbb{N} : \forall y \in W_x.\ \exists z \in W_x.(y < z) \wedge (\varphi_x(y) > \varphi_x(z))\}$.

Analisi: B contiene gli indici x per cui: per ogni elemento y nel dominio di φ_x, esiste un elemento $z > y$ nel dominio tale che $\varphi_x(y) > \varphi_x(z)$.

In altre parole, per ogni input, esiste un input maggiore che produce un output minore.

Saturazione: B è saturato: $B = \{x \mid \varphi_x \in \mathcal{B}\}$ dove $\mathcal{B}$ è l'insieme delle funzioni che soddisfano la proprietà descritta.

Non ricorsività per Rice:

- $B \neq \mathbb{N}$: la funzione identità $\varphi_x(y) = y$ non soddisfa la proprietà (per ogni y, tutti gli $z > y$ danno $\varphi_x(z) = z > y = \varphi_x(y)$)
- $B \neq \emptyset$: possiamo costruire funzioni che soddisfano la proprietà

Per Rice, B non è ricorsivo.

Analisi della semidecidibilità: B coinvolge una quantificazione universale ($\forall y \in W_x$) seguita da una quantificazione esistenziale ($\exists z \in W_x$).

La presenza di quantificazione universale su domini infiniti generalmente rende gli insiemi non semidecidibili.

Dimostrazione che B non è r.e.: Intuitivamente, per verificare $x \in B$, dovremmo:

1. Enumerare tutti gli elementi di W_x
2. Per ogni $y \in W_x$, trovare $z \in W_x$ con $z > y$ e $\varphi_x(y) > \varphi_x(z)$

Il problema è che non possiamo mai essere sicuri di aver trovato tutti gli elementi di W_x, quindi non possiamo verificare la proprietà universale.

Riduzione formale $\bar{K} \leq_m B$: Definiamo g(u,v):

g(u,v) = {v se u ∉ K
{↑ se u ∈ K

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

- Se $u \notin K$: $\varphi_{s(u)}$ = identità su $\mathbb{N}$, che non soddisfa la proprietà di B
- Se $u \in K$: $\varphi_{s(u)}$ ha dominio vuoto, quindi soddisfa vacuamente la proprietà

Quindi: $u \notin K \iff s(u) \notin B$, cioè $\bar{K} \leq_m \bar{B}$.
Poiché $\bar{K}$ non è r.e., $\bar{B}$ non è r.e.

Complemento $\bar{B}$: $\bar{B} = \{x \in \mathbb{N} : \exists y \in W_x.\ \forall z \in W_x.(z \leq y) \vee (\varphi_x(y) \leq \varphi_x(z))\}$

$\bar{B}$ è semidecidibile: per verificare $x \in \bar{B}$, cerchiamo un $y \in W_x$ tale che per tutti gli $z > y$ in W_x, $\varphi_x(y) \leq \varphi_x(z)$.

Conclusione:

- B non è ricorsivo
- B non è r.e.
- $\bar{B}$ è r.e. ma non ricorsivo □

Esercizio 5

Enunciare il secondo teorema di ricorsione. Utilizzarlo per dimostrare che esiste un indice $e \in \mathbb{N}$ tale che $W_e = \{e^n : n \in \mathbb{N}\}$.

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

Dimostrazione dell'esistenza dell'indice:

Vogliamo costruire e tale che $W_e = \{e^n : n \in \mathbb{N}\} = \{1, e, e^2, e^3, ...\}$.

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

h(x,n) = {x^n se input è n-mo tentativo di calcolare x^n
{↑ altrimenti (per controllare il dominio)

Più precisamente, vogliamo $\varphi_{s(x)}$ tale che:

- $\varphi_{s(x)}(n) = x^n$ per ogni $n \in \mathbb{N}$
- $W_{s(x)} = \mathbb{N}$ (per assicurare che il dominio contenga tutti gli n)

Implementazione tramite SMN: Definiamo g(x,y): $g(x,y) = x^y$

Questa funzione è computabile (l'esponenziazione è primitiva ricorsiva).

Per il teorema SMN, esiste $s: \mathbb{N} \to \mathbb{N}$ totale e computabile tale che:
$\varphi_{s(x)}(y) = g(x,y) = x^y$

Quindi: $W_{s(x)} = \mathbb{N}$ e $E_{s(x)} = \{x^n : n \in \mathbb{N}\}$.

Ma noi vogliamo $W_e = \{e^n : n \in \mathbb{N}\}$, non $E_e = \{e^n : n \in \mathbb{N}\}$.

Correzione: costruzione per il dominio: Ridefinimo g(x,y) per controllare il dominio:

g(x,y) = {y se $y = x^n$ per qualche $n \in \mathbb{N}$
{↑ altrimenti

Implementazione: g(x,y) = {y se $\exists n \leq \log_x(y).\ x^n = y$ {↑ altrimenti

= $\mu n.((x^n = y) \to y, \uparrow)$

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

Quindi: $W_{s(x)} = \{y : y = x^n \text{ per qualche } n\} = \{x^n : n \in \mathbb{N}\}$.

Applicazione del Secondo Teorema di Ricorsione: Applicando il secondo teorema di ricorsione alla funzione s, esiste $e \in \mathbb{N}$ tale che: $\varphi_e = \varphi_{s(e)}$

Quindi:
$W_e = W_{s(e)} = \{e^n : n \in \mathbb{N}\}$

Verifica: L'indice e soddisfa la proprietà richiesta: il suo dominio è esattamente l'insieme delle potenze di e. □