Complessità

3 Novembre 2003

Funzioni parziali

- $X \subseteq \mathbb{N}^m$
- $f : X \to \mathbb{N}^n$

f è una *funzione parziale* $\mathbb{N}^m \to \mathbb{N}^n$

$$X = \mathbb{N}^m$$
$$\Rightarrow$$
f *funzione totale* $\mathbb{N}^m \to \mathbb{N}^n$

Funzioni iniziali

- **funzione zero**

$$\zeta() = 0$$

- **funzione successivo**

$$\sigma(x) = x + 1$$

- **proiezioni**

$$\pi_v^k : \begin{cases} \mathbb{N}^k \to \mathbb{N} \\ (x_1, \ldots, x_k) \to x_v \end{cases}$$

Definiamo inoltre $\pi_0^k(x) = ()$

Operazioni sulle funzioni

- **combinazione**

$$f : \mathbb{N}^k \to \mathbb{N}^m, \quad g : \mathbb{N}^k \to \mathbb{N}^n$$

$$(f \times g) : \begin{cases} \mathbb{N}^k \to \mathbb{N}^{m+n} \\ x \to (f(x), g(x)) \end{cases}$$

- **composizione**

$$f : \mathbb{N}^k \to \mathbb{N}^m, \quad g : \mathbb{N}^m \to \mathbb{N}^n$$

$$(f \cdot g)(x) := g(f(x))$$

- **ricorsione primitiva**

$$g : \mathbb{N}^k \to \mathbb{N}^m, \quad h : \mathbb{N}^{k+m+1} \to \mathbb{N}^m$$

$$f : \begin{cases} \mathbb{N}^{k+1} \to \mathbb{N}^m \\ f(x, 0) = g(x) \\ f(x, y+1) = h(x, y, f(x, y)) \end{cases}$$

Funzioni primitive ricorsive

$f : \mathbb{N}^m \to \mathbb{N}^n$ *primitiva ricorsiva*

> f costruita a partire da ζ, σ, π con un numero *finito* di combinazioni, composizioni e ricorsioni primitive.

Esempio:

Somma di due numeri naturali

$$\text{plus} : \begin{cases} \mathbb{N}^2 \to \mathbb{N} \\ (x, 0) \to \pi_1^1(x) \\ (x, y+1) \to \sigma \cdot \pi_3^3(x, y, \text{plus}(x, y)) \end{cases}$$

Funzione plus

```
(defun plus (x y)
  (cond
   ((= 0 y) x)
   (t (sigma
    (prj 3 3
     `(,x ,y ,(plus x (- y 1)))
    )))))
```

Traccia esecuzione:

```
(plus 2 3)
1.  Trace:  (PLUS '2 '3)
2.  Trace:  (PLUS '2 '2)
3.  Trace:  (PLUS '2 '1)
4.  Trace:  (PLUS '2 '0)
4.  Trace:  PLUS ==> 2
3.  Trace:  PLUS ==> 3
2.  Trace:  PLUS ==> 4
1.  Trace:  PLUS ==> 5
5
```

Funzioni primitive ricorsive/2

- funzione costante K_m^0

$$K_0^0 = \zeta()$$
$$K_{m+1}^0 = \sigma \cdot K_m^0$$

- funzione costante K_m^n (ha per dominio $\mathbb{N}^n$)

$$K_m^n(x, 0) = K_m^{n-1}(x)$$
$$K_m^n(x, y+1) = \pi_{n+2}^{n+2}(x, y, K_m^n(x, y))$$

- predecessore pred

$$\mathsf{pred}(0) = \zeta()$$
$$\mathsf{pred}(y+1) = \pi_1^2(y, \mathsf{pred}(y))$$

N.B. Il predecessore di 0 è 0

- prodotto di numeri naturali mult

$$\mathsf{mult}(x, 0) = 0$$
$$\mathsf{mult}(x, y+1) = \mathsf{plus}(x, \mathsf{mult}(x, y))$$

Funzione pred

```
(defun pred (x)
 (cond
  ((= 0 x) 0)
  (t (prj 1 2 `(,(- x 1) ,(pred (- x 1)))))
 ))
```

Traccia esecuzione:

```
(pred 4)
1.  Trace:  (PRED '4)
2.  Trace:  (PRED '3)
3.  Trace:  (PRED '2)
4.  Trace:  (PRED '1)
5.  Trace:  (PRED '0)
5.  Trace:  PRED ==> 0
4.  Trace:  PRED ==> 0
3.  Trace:  PRED ==> 1
2.  Trace:  PRED ==> 2
1.  Trace:  PRED ==> 3
3
```

Funzioni primitive ricorsive/3

- funzione mon $\dot{-}$

$$\mathsf{mon}(x, 0) = x$$
$$\mathsf{mon}(x, (y + 1)) = \mathsf{pred}(\mathsf{mon}(x, y))$$

- funzione eq

$$\mathsf{eq}(x, y) = 1 \dot{-} ((y \dot{-} x) + (x \dot{-} y))$$

Più formalmente:

$$\mathsf{mon} \cdot (K_1^2 \times (\mathsf{plus} \cdot ((\mathsf{mon} \cdot (\pi_2^2 \times \pi_1^2)) \times \mathsf{mon} \cdot (\pi_1^2 \times \pi_2^2))))$$

- funzione quo (parte intera di x/y se $y \neq 0$; altrimenti 0)

$$\mathsf{quo}(0, y) = 0$$
$$\mathsf{quo}(x + 1, y) = \mathsf{quo}(x, y) +$$
$$\mathsf{eq}(x + 1, \mathsf{mult}(\mathsf{quo}(x, y), y) + y)$$

Funzioni primitive ricorsive/4

Esempi:

$$\begin{aligned} eq(5,3) &= 1 \dot{-} ((3 \dot{-} 5) + (5 \dot{-} 3)) \\ &= 1 \dot{-} (0 + 2) \\ &= 1 \dot{-} 2 \\ &= 0 \end{aligned}$$

$$\begin{aligned} eq(4,4) &= 1 \dot{-} ((4 \dot{-} 4) + (4 \dot{-} 4)) \\ &= 1 \dot{-} (0 + 0) \\ &= 1 \end{aligned}$$

Funzioni primitive ricorsive/5

- Ogni funzione primitiva ricorsiva è totale
- Esistono funzioni computabili non totali (ad esempio divisione esatta)
- Esistono funzioni computabili totali non primitive ricorsive.

Funzione di Ackermann

$$A : \mathbb{N}^2 \to \mathbb{N}$$

$$A(0, y) = y + 1$$
$$A(x + 1, 0) = A(x, 1)$$
$$A(x + 1, y + 1) = A(x, A(x + 1, y))$$

Funzione di Ackermann

```
(defun ack (x y)
 (cond
  ((= x 0) (+ y 1))
  ((= y 0) (ack (- x 1) 1))
  (t       (ack (- x 1) (ack x (- y 1))))
))
```

$$\log_{10} A(4,2) = 19728.3017958\ldots$$

Funzioni primitive ricorsive/6

Th: Esiste una funzione calcolabile totale

$$f : \mathbb{N} \to \mathbb{N}$$

che *non* è primitiva ricorsiva.

Osservazione:

$$\Sigma = \left\{ \begin{array}{l} \text{Insieme funzioni primitive} \\ \text{ricorsive} \end{array} \right\}$$

- esiste una biiezione $\mathbb{N} \to \Sigma$
- Σ è naturalmente ordinato

Funzioni primitive ricorsive/7

Dim:

1. Σ = insieme funzioni primitive ricorsive

2. Σ è ordinabile

   a. s primitiva ricorsiva $\Rightarrow$ s può essere descritta da una stringa finita di lunghezza s

   b. $|s_1| \leq |s_2| \Rightarrow s_1 \leq s_2$

   c. stringhe di uguale lunghezza sono poste in ordine alfabetico.

3. f_i = i–esima funzione primitiva ricorsiva.

4. $f : \mathbb{N} \rightarrow \mathbb{N}$,

$$f(n) = f_n(n) + 1$$

5. f è totale e calcolabile

6. f non è primitiva ricorsiva

Funzioni primitive ricorsive/8

f primitiva ricorsiva $\Rightarrow$ esiste m tale che

$$f = f_m$$

Allora:

$$f_m(m) + 1 = f(m) = f_m(m)$$

Contraddizione

Funzioni ricorsive parziali: Minimalizzazione

$g : \mathbb{N}^{n+1} \to \mathbb{N}$

$f : \mathbb{N}^n \to \mathbb{N}$

$$f(x) := \mu y[g(x, y) = 0]$$

minimalizza g

se $f(x)$ è il più piccolo intero tale che

- $g(x, f(x)) = 0$
- per ogni $z \leq f(x)$ ($z \in \mathbb{N}$),
  $g(x, z)$ è definita

Minimalizzazione/2

Esempio:

$$\mathsf{div}(x, y) := \mu t[((x + 1) \dot{-} (\mathsf{mult}(t, y) + y)) = 0]$$

$$\mathsf{div}(x, y) := \begin{cases} \text{parte intera di } x/y \text{ se } y \neq 0 \\ \text{non definita se } y = 0 \end{cases}$$

La minimalizzazione può produrre funzioni calcolabili non totali da funzioni calcolabili totali

Tesi di Church/1

La classe delle funzioni parziali ricorsive è l'insieme di tutte le funzioni ottenute a partire dalle funzioni iniziali ζ, σ, π_n^k mediante l'applicazione di un numero finito di volte delle operazioni di

1. combinazione
2. composizione
3. ricorsione primitiva
4. minimalizzazione.

Tesi di Church/2

Ogni funzione calcolabile è una funzione parziale ricorsiva

La tesi di Church e quella di Turing sono fra loro equivalenti

Complessità

- Temporale: un algoritmo è considerato più complesso di un altro se esso richiede più tempo (misurato in termini di impiego di funzioni iniziali o operazioni su di esse) di un altro

- Spaziale: un algoritmo è considerato più complesso di un altro se esso richiede una quantità superiore di memoria (da misurarsi in termini di dimensione degli spazi $\mathbb{N}^n$ considerati)

Classi di complessità

P: Classe di problemi per cui esiste un algoritmo f ed un polinomio $p(x)$ tale che, per ogni preassagnato dato in ingresso w è possibile calcolare $f(w)$ in al più $p(|w|)$ passaggi.

Osservazione:
La classe P è chiusa rispetto la composizione di funzioni

Ordini di crescita

- $f(x)$ è $O(g(x))$ se

  esistono X, $c > 0$ tali che

  $$cf(x) < g(x)$$

  per quasi ogni $x > X$

- $f(x)$ è $\Omega(g(x))$ se

  esistono X, $c > 0$ tali che

  $$cf(x) > g(x)$$

  per quasi ogni $x > X$.

- $f(x)$ è $\Theta(g(x))$ se

  1. f è $O(g(x))$
  2. f è $\Omega(g(x))$

Ordini di crescita: esempi

- $x \in O(x^2)$
- $x^2 \in \Omega(x)$
- $\log x \in O(x)$
- $10x^3 + x^2 \in \Theta(x^3)$

Ordini di crescita/2

Nomi di alcuni ordini di crescita

1	costante
$\log(\log(n))$	log–log
$\log(n)$	logaritmico
$(\log(n))^k$	polilogartimico
$n^{1/k}$	radice
n	lineare
n^2	quadratico
n^k	polinomiale
e^n	esponenziale
$n!$	fattoriale

Ricorrenze e ordini di crescita

Th: Siano

- $a \in \mathbb{N}$, $a \geq 1$
- $b \in \mathbb{R}$, $b > 1$
- $c \in \mathbb{R}$, $c > 0$
- $d \in \mathbb{R}$, $d \geq 0$
- $f(x) \in \Theta(x^c)$

Dato

$$T(n) := \begin{cases} aT(n/b) + f(n) & \text{se } n > 1 \\ d & \text{se } n = 1 \end{cases}$$

per ogni n potenza di b,

1. $\log_b a < c \quad \Rightarrow \quad T(n) \in O(n^c)$
2. $\log_b a = c \quad \Rightarrow \quad T(n) \in O(n^c \log n)$
3. $\log_b a > c \quad \Rightarrow \quad T(n) \in O(n^{\log_b a})$