Informatica Teorica

Appunti delle lezioni
su funzioni ricorsive primitive e generali,
e sul rapporto tra ricorsività
e T-computabilità

Funzioni base e operazioni

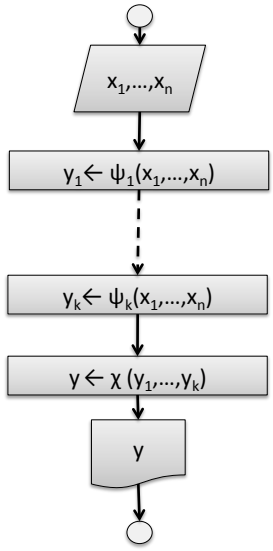
- Funzioni base:
  - funzione zero, $Z:N\rightarrow N$, $Z(x)=0$ per ogni x
  - funzione successore, $s:N\rightarrow N$, $s(x)$ = il successore di x nell'enumerazione di N
  - proiezioni, $P^n_i:N^n\rightarrow N$, $P^n_i(x_1,\ldots,x_n)=x_i$
- Operazione di composizione:
  - date $\chi:N^k\rightarrow N$ e $\psi_1,\ldots,\psi_k:N^n\rightarrow N$
  - si ottiene per composizione la funzione $\phi:N^n\rightarrow N$ quando $\phi(x_1,\ldots,x_n) = \chi(\psi_1(x_1,\ldots,x_n),\ldots,\psi_k(x_1,\ldots,x_n))$
- Operazione di ricorsione:
  - date $\chi:N^n\rightarrow N$ e $\psi:N^{n+2}\rightarrow N$
  - si ottiene per ricorsione la funzione $\phi:N^{n+1}\rightarrow N$ quando

$$\begin{cases} \phi(x_1,\ldots,x_n,0) = \chi(x_1,\ldots,x_n) \\ \phi(x_1,\ldots,x_n,s(y)) = \psi(x_1,\ldots,x_n,y,\phi(x_1,\ldots,x_n,y)) \end{cases}$$

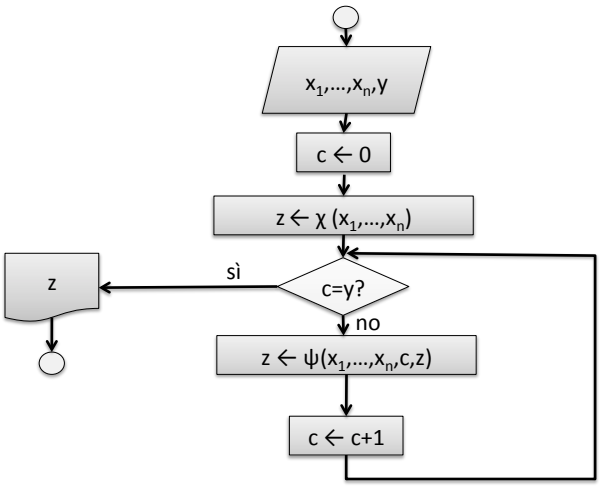
Funzioni ricorsive primitive (RP)

- Si dice ricorsiva primitiva una funzione che si ottiene dalle funzioni base applicando un numero finito di volte le operazioni di composizione e ricorsione
- RP è linsieme di tutte e sole le funzioni ricorsive primitive
- Una derivazione in RP è una sequenza di funzioni, ciascuna delle quali è una funzione base, oppure è stata ottenuta dalle funzioni precedenti tramite composizione o ricorsione
- L'ultima funzione in una derivazione in RP si dice RP-derivabile
- Una fuzione è in RP ⇔ è RP-derivabile
- Esempio di derivazione dell'addizione in RP:
  1. $\phi_1(x) = P^1_1(x)$ [base]
  2. $\phi_2(x) = s(x)$ [base]
  3. $\phi_3(x,y,z) = P^3_3(x,y,z)$ [base]
  4. $\phi_4(x,y,z) = s(P^3_3(x,y,z))$ [composizione di 2 e 3]
  5. $\begin{cases} \phi_5(x,0) = P^1_1(x) \\ \phi_5(x,s(y)) = s(P^3_3(x,y,\phi_5(x,y))) \end{cases}$ [ricorsione di 1 e 4]

La composizione conserva la computabilità e la totalità delle funzioni



La ricorsione conserva la computabilità e la totalità delle funzioni



Le funzioni RP

- Le funzioni base sono computabili e totali
- Le operazioni conservano tali caratteristiche delle funzioni a cui sono applicate
- Perciò le funzioni RP sono computabili e totali

Funzioni computabili totali non RP

- Le funzioni RP (ad 1 argomento, senza perdere in generalità) ϕ_i sono enumerabili perché possiamo costruire una sequenza di funzioni ottenute da quelle base con 0, 1, 2, 3,… applicazioni delle operazioni di composizione e ricorsione
- Data tale sequenza, definiamo $f(x) = \phi_x(x)+1$
- f è computabile e totale
- Se supponiamo che f sia nella sequenza, ovvero che $f \equiv \phi_k$, calcoliamo f(k)
- Per la definizione di f, $f(k) = \phi_k(k)+1$
- Per la posizione di f nella sequenza, $f(k) = \phi_k(k)$
- Si conclude che $\phi_k(k)+1 = \phi_k(k)$, ossia, visto che ϕ_k è totale e quindi $\phi_k(k)$ è un valore definito, che $1 = 0$
- Dobbiamo respingere l'ipotesi che f sia nella sequenza, ossia dobbiamo ammettere che esistono funzioni computabili totali che NON sono in RP

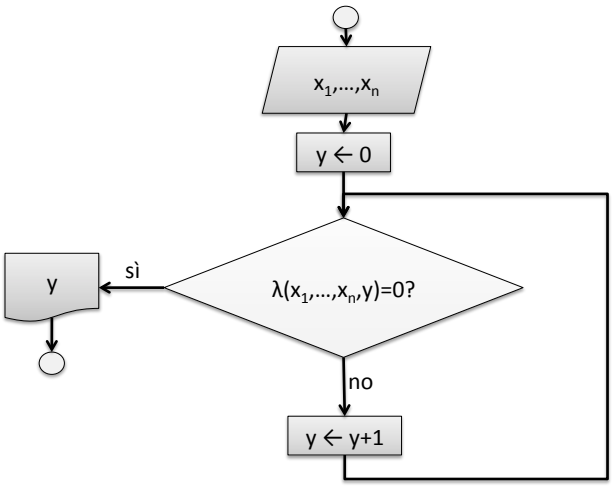
Nuova operazione: minimizzazione

- Operazione di minimizzazione:
  - data $\lambda:N^{n+1}\rightarrow N$
  - si ottiene per minimizzazione la funzione $\phi:N^n\rightarrow N$ quando

    $\phi(x_1,\ldots,x_n) = \mu y(\lambda(x_1,\ldots,x_n,y)=0)$,

    dove $\mu y(P)$ vuol dire "il più piccolo y tale che P"
- Funzioni definite tramite la minimizzazione possono essere parziali, perché, dati $x_1,\ldots,x_n$, non è garantita l'esistenza di un y tale che $\lambda(x_1,\ldots,x_n,y)=0$

La minimizzazione conserva la computabilità delle funzioni



Funzioni ricorsive generali (RG)

- Si dice ricorsiva generale una funzione che si ottiene dalle funzioni base applicando un numero finito di volte le operazioni di composizione, ricorsione, e minimizzazione
- RG è linsieme di tutte e sole le funzioni ricorsive generali
- Una derivazione in RG è una sequenza di funzioni, ciascuna delle quali è una funzione base, oppure è stata ottenuta dalle funzioni precedenti tramite composizione, ricorsione, o minimizzazione
- L'ultima funzione in una derivazione in RG si dice RG-derivabile
- Una fuzione è in RG ⇔ è RG-derivabile
- Esempio di funzione in RG:

$$\phi(x,y)=\begin{cases} x/y, \text{ se x è multiplo di y} \\ \perp, \text{ altrimenti} \end{cases}$$

1. $\phi_1(x) = pr(x)$ [funzione predecessore, definita per ricorsione da z(x) e $P^1_1(x)$]
2. $\phi_2(x,y) = x \dot{-} y$ [differenza simmetrica, definita per ricorsione da $P^1_1(x)$ e pr(x)]
3. $\phi_3(x,y) = dist(x,y) = x \dot{-} y + y \dot{-} x$ [distanza, definita per composizione da + e $\dot{-}$]
4. $\phi_4(x,y) = \mu z(dist(z*y, x) = 0)$ [definita per minimizzazione da una composizione di * e dist]

- ϕ_4 è la funzione ϕ, ed è stata derivata da funzioni RP con operazioni di composizione, ricorsione, e minimizzazione, quindi è in RG

Enumerazione delle RG

- In questo caso non si conclude nulla sull'appartenenza di $f(x) = \phi_x(x)+1$ a RG o meno, perché l'uguaglianza $\phi_k(k)+1 = \phi_k(k)$ non porta ad alcun assurdo
- Semplicemente, dobbiamo ammettere che ϕ_k, se appartiene alla sequenza delle RG, non sia definita in k, il che è compatibile con la parzialità delle RG
- Il tutto è compatibile anche con l'ipotesi che f NON sia nella sequenza delle RG

Estensione del concetto di ricorsività

- Un predicato $R(x_1,\ldots,x_n)$ si dice ricorsivo generale/primitivo quando lo è la sua funzione caratteristica, che corrisponde alla funzione caratteristica dell'insieme determinato da tale predicato:

  $A_R=\{(x_1,\ldots,x_n) \in N^n \mid R(x_1,\ldots,x_n) \text{ è vero}\}$

  $f_R(x_1,\ldots,x_n) = \begin{cases} 0 \text{ se } R(x_1,\ldots,x_n) \text{ è vero} \\ 1 \text{ altrimenti} \end{cases}$

Predicati RP

- x = y è RP, perché $f_{=}(x,y) = sgn(dist(x,y))$,
  dove $sgn(x) = \begin{cases} 0 \text{ se } x = 0 \\ 1 \text{ se } x > 0 \end{cases}$
  ed è definita per ricorsione da z(x) e s(z(x))
- $x \neq y$ è RP, perché $f_{\neq}(x,y) = \overline{sgn}(dist(x,y))$,
  dove $\overline{sgn}(x) = \begin{cases} 1 \text{ se } x = 0 \\ 0 \text{ se } x > 0 \end{cases}$
  definita per ricorsione da s(z(x)) e z(x)
- In generale: $f_R \equiv \overline{sgn} \circ f_{\neg R}$
  $f_{\neg R} \equiv \overline{sgn} \circ f_R$

Congiunzioni e disgiunzioni

- Se f_R e f_S sono le funzioni caratteristiche dei predicati R ed S, allora:

  $f_{R \wedge S} \equiv sgn(f_R + f_S)$

  $f_{R \vee S} \equiv f_R * f_S$

Insiemi ricorsivi

- Un insieme si dice ricorsivo generale/primitivo quando la sua funzione caratteristica è ricorsiva generale/primitiva
- Un insieme A è RG/RP ⇔ lo è $\bar{A}$:
  - infatti vale $f_A \equiv \cancel{sgn} \circ f_{\bar{A}}$ e anche $f_{\bar{A}} \equiv \cancel{sgn} \circ f_A$, quindi se una delle due funzioni è RG/RP lo è anche l'altra perché ottenuta per composizione con ~~sgn~~ che è RP

Insiemi ricorsivamente enumerabili

- Un insieme si dice ricorsivamente enumerabile quando
  - è vuoto, oppure
  - è il rango di una funzione ricorsiva generale totale
- Un insieme A è RG ⇔ A e $\bar{A}$ sono ricorsivamente enumerabili
  (⇒) se A è vuoto, è banale; altrimenti definiamo $\phi(x) = \cancel{sgn}(f_A(x)) * x + f_A(x) * \mu y(P^2_2(x, f_A(y)) = 0)$,
  che restituisce x se $x \in A$, e il più piccolo elemento di A altrimenti
  A è il rango di ϕ, che è RG e totale, cioè A è ricorsivamente enumerabile
  Visto che A è RG, lo è anche $\bar{A}$, e si ripete il ragionamento per dimostrare che anche $\bar{A}$ è ricorsivamente enumerabile

Insiemi ricorsivamente enumerabili (2)

- Un insieme A è RG ⇔ A e $\bar{A}$ sono ricorsivamente enumerabili
  (⇐) se A oppure $\bar{A}$ è vuoto, è banale;
  altrimenti, date:
  $\phi(x)$ funzione RG totale il cui rango è A, e
  $\psi(x)$ funzione RG totale il cui rango è $\bar{A}$, definiamo
  $\lambda(x)=\mu y(dist(\phi(y),x)*dist(\psi(y),x) =0)$,
  ossia una funzione che, dato x, restituisce il più piccolo y tale che $\phi(y)$ sia uguale a x, oppure $\psi(y)$ sia uguale a x, con cui
  definiamo
  $h(x) = sgn(dist(\phi(\lambda(x)),x))$
  che è f_A, dal momento che
  $h(x) = sgn(dist(x,x)) = 0$, se $x \in A$ (perché $\phi(\lambda(x)) = x$), mentre
  $h(x) = sgn(dist(\phi(\lambda(x)),x)) = 1$, se $x \notin A$ (perché $\psi(\lambda(x)) = x$, mentre $\phi(\lambda(x)) \neq x$).
  Poiché h(x) (ossia f_A) è RG, in quanto ottenuta da funzioni RG con composizione e minimizzazione, si dimostra che A è RG.

Sono equivalenti le seguenti proposizioni:

a) L'insieme A è ricorsivamente enumerabile
b) L'insieme A è il campo di esistenza di una funzione RG parziale
c) L'insieme A è il rango di una funzione RG parziale

Teorema (Turing, 1937)

- Una funzione f è RG ⇔ f è T-computabile
- (⇒)
  - si dimostra che le funzioni base sono T-computabili
  - e che le operazioni di composizione, ricorsione, e minimizzazione conservano la T-computabilità

La funzione zero è T-computabile

- La seguente macchina di Turing MTz infatti computa la funzione zero:

  $q_1 \mid s_0\ D\ q_1$

  $q_1\ s_0 \mid C\ q_0$

La funzione successore è T-computabile

- La seguente macchina di Turing MTs infatti computa la funzione successore:

  $q_1 \mid \mid D\ q_1$

  $q_1\ s_0 \mid C\ q_0$

Le funzioni proiezione sono T-computabili

- La seguente macchina di Turing MTp^3_1, ad esempio, computa la funzione $P^3_1(x,y,z) = x$:

  $q_1 \mid \mid D\ q_1$

  $q_1\ s_0\ s_0\ D\ q_2$

  $q_2 \mid s_0\ D\ q_2$

  $q_2\ s_0\ s_0\ D\ q_3$

  $q_3 \mid s_0\ D\ q_3$

  $q_3\ s_0\ s_0\ C\ q_0$

La composizione conserva la T-computabilità

- $\phi(x) = \chi \circ \psi(x) = \chi(\psi(x))$
- Se χ e ψ sono T-computabili, lo è anche ϕ?
- Ossia: se esistono MTχ e MTψ, si riesce a costruire MTϕ?
- Sì: basta aggiungere le istruzioni di MTχ a quelle di MTψ, con i seguenti accorgimenti:
  - rinominare gli stati interni di MTχ per evitare sovrapposizione con gli stati di MTψ
  - aggiungere istruzioni che portino la testina della macchina nella posizione standard dopo che sono state eseguite le istruzioni di MTψ e prima che vengano eseguite quelle di MTχ

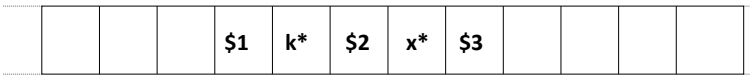
La ricorsione conserva la T-computabilità

$$\begin{cases} \phi(0) = k \\ \phi(s(x)) = \psi(x, \phi(x)) \end{cases}$$

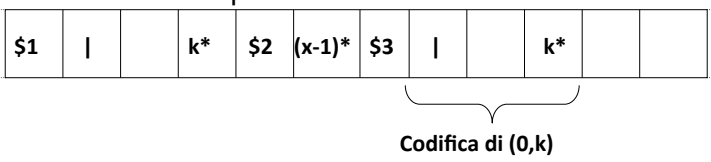
- Se ψ è T-computabile, lo è anche ϕ?
- Ossia: avendo a disposizione MT_{ψ}, si riesce a costruire MT_{ϕ}?
- Sì: viene mostrato nelle seguenti slide
- L'alfabeto include i simboli \$1, \$2, \$3 che sono usati come separatori

La ricorsione con MT

- Per calcolare $\phi(x)$, MT_ϕ inizia scrivendo sul nasto (k* rappresenta k+1 barre):

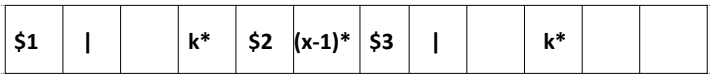


- MT_ϕ cancella una barra da x*
- Se ci sono zero barre tra $2 e $3, l'output è tra $1 e $2 (perché $\phi(0) = k$)
- Altrimenti MT_ϕ configura il nastro così:

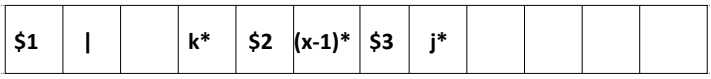


La ricorsione con MT

- MT_ϕ esegue il programma di MT_ψ sulla parte a destra di $3:



ottenendo:

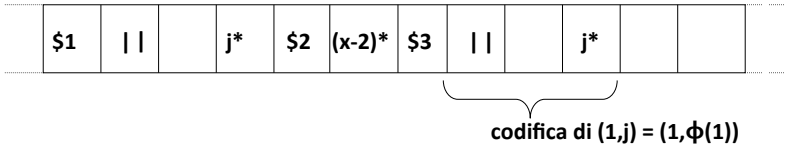


dove j* è la codifica di $j = \psi(0,k) = \psi(0,\phi(0)) = \phi(1)$

La ricorsione con MT

- MT_ϕ cancella una barra tra \$2 e \$3, se non ci sono più barre vuol dire che x=1 e quindi l'output è j = ψ(0,k) = ϕ(1), che si trova a destra di \$3:

- Altrimenti nuova configurazione e si ripete:



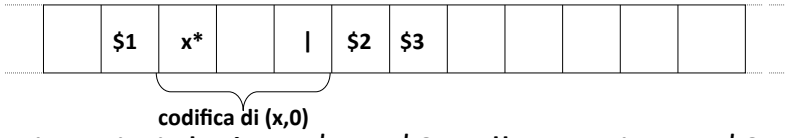
- Eseguendo MT_ψ a destra di \$3 si calcola ψ(1,ϕ(1)) = ϕ(2)...e così via fino a ϕ(x)

La minimizzazione conserva la T-computabilità

- ϕ(x) = il più piccolo y: λ(x,y)=0
- Dobbiamo costruire MT_ϕ usando MT_λ
- MT_ϕ procede eseguendo iterativamente il codice di MTλ per calcolare λ(x,0), λ(x,1), λ(x,2),...,λ(x,y) e restituisce in output il primo y per cui λ(x,y)=0
- Se tale y non esiste MT_ϕ non si ferma mai (e infatti ϕ non è definita per quella x)

La minimizzazione con MT

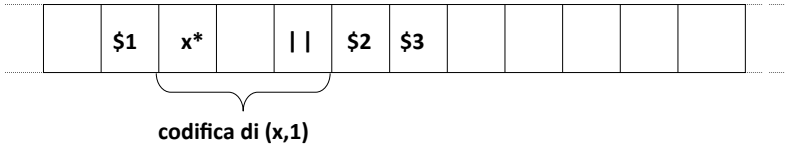
- MT_ϕ configura inizialmente il nastro così:



- Poi copia i dati tra $1 e $2 nello spazio tra $2 e $3 e lì usa il codice di MT_λ per calcolare λ(x,0)
- Se il risultato è zero cancella tutto il resto e lo lascia come output

La minimizzazione con MT

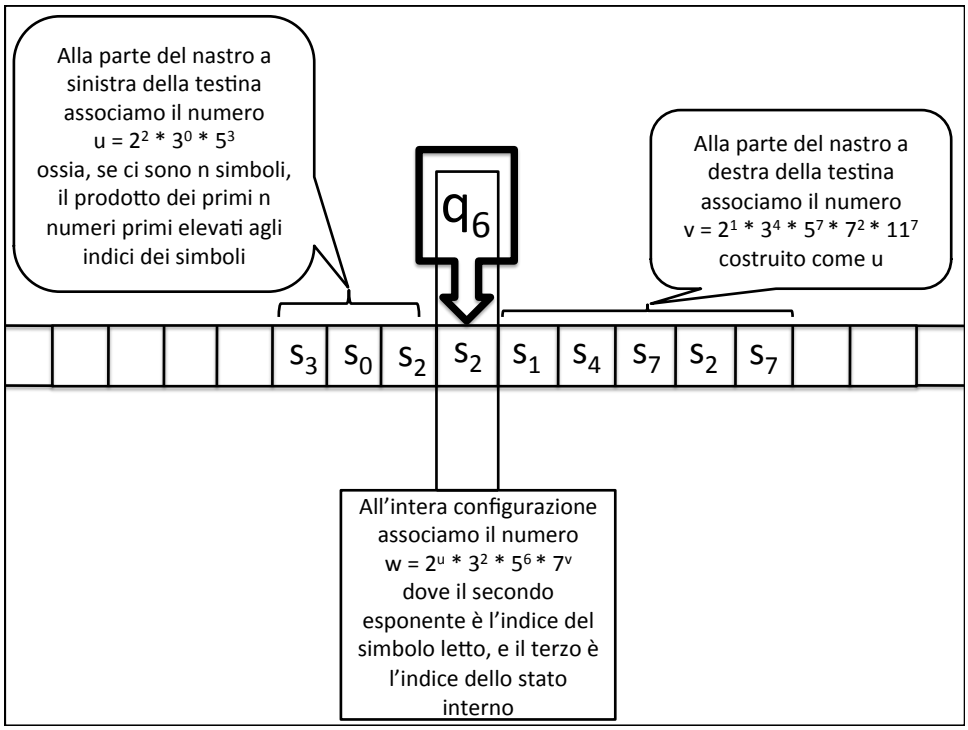
- Altrimenti MT_ϕ riconfigura il nastro così:



  e ripete tutto per calcolare λ(x,1)
- ...e così via fino a trovare (eventualmente) il primo y per cui λ(x,y)=0

Gödelizzazione delle configurazioni di una MT

- Prima di procedere alla dimostrazione (sommaria) di (⇐), introduciamo delle codifiche elaborate da Gödel che permettono di esprimere le computazioni delle MT come applicazioni di funzioni matematiche



Il programma di MT come funzione

- L'indice w è associato alla configurazione della MT in maniera univoca, grazie all'unicità della scomposizione di un numero in fattori primi
- Per mezzo delle sue istruzioni, la MT passa da una configurazione all'altra in modo deterministico: da w la MT non può fare altro che passare a w' (a meno che w non sia una configurazione finale)
- Ad ogni MT possiamo quindi associare una funzione:

$$\rho_{MT}(w)=\begin{cases} w' \text{ se } w \text{ è la codifica di una configurazione non finale} \\ w \text{ altrimenti} \end{cases}$$

Il determinismo di MT garantisce che ρ_{MT} sia una funzione

Si può dimostrare che qualunque sia la MT, ρ_{MT} è in RP

La computazione di un numero finito di passi con MT come funzione

- A partire da ρ_{MT}, definiamo θ_{MT} per ricorsione:

$$\begin{cases} \theta_{MT}(w,0) = w \\ \theta_{MT}(w,s(z)) = \rho_{MT}(\theta_{MT}(w,z)) \end{cases}$$

- Se w è la codifica di una configurazione w, $\theta_{MT}(w,z)$ è la codifica della configurazione che si ottiene dopo z istruzioni a partire da w
- Anche θ_{MT} è RP, perché definita per ricorsione da una funzione RP

f è T-computabile ⇒ f è RG

- Sia MT_f la MT che computa f
- Dato l'input x, lo si codifichi sul nastro di MT_f
- La computazione di MT_f è data dalla funzione RP $\theta_{MTf}(w,z)$, e con un'operazione di minimizzazione si ottiene (se esiste) il minimo numero z_0 di passi per cui, a partire dalla codifica della configurazione di partenza w_1, si ottiene una configurazione finale
- Calcolando $\theta_{MTf}(w_1,z_0)$ si ottiene la codifica della configurazione finale, e invertendo la gödelizzazione, si ricava la codifica dell'output sul nastro
- Decodificando l'output, si ottiene f(x)
- f è stata ottenuta componendo tutti i calcoli di cui sopra, fatti con funzioni RP e con una minimizzazione su una funzione RP, quindi f è RG