PROBLEMI

Risolvere problemi è calcolare una funzione.

Definisco 3 alfabeti:

$$\Sigma_1, \Sigma_0, \Sigma_1 \text{ t.c. } \#, \triangleright \notin \Sigma_0 \cup \Sigma_1 \subset \Sigma$$

$$\text{e } f: \Sigma_0^* \to \Sigma_1^*$$

f è calcolata dalla macchina $M = (Q, \Sigma, \delta, q_0)$

ossia f risulta Turing calcolabile se

$$\forall x \in \Sigma_0^*.\ f(x) = y \Leftrightarrow (q_0, \underline{\triangleright} x) = M(x) \xrightarrow{*} (h, \triangleright y \underline{\#})$$

"g. Calcola $g: Var \to \mathbb{N}$

(se $\exists$ tale C allora g è WHILE calcolabile)

$$\forall \sigma \in Var \to \mathbb{N}.\ g(x) = n \Leftrightarrow (c, \sigma) \xrightarrow{*} \sigma' \wedge \sigma'(x) = n$$

g potrebbe non essere definita per un linguaggio FOR

(potrebbe dividere per 0)

La nozione di calcolabilità non dipende dal tipo di dati

(possono essere benissimo non naturali e non riguardare funzioni $\mathbb{N} \to \mathbb{N}$ ma in questo corso si assume lo sia)

Suppongo $f: A \to B$

- $\forall a \in A$: codifico a come numero naturale
- Calcolo $f(n) = y$
- Decodifico y in B

Esempi di funzioni di codifica: carattere ASCII $\to$ numero

tassellare il piano

	0	1	2	3	4	5	...
0	0	2	5	9	14	20	
1	1	4	8	13	19	...	
2	3	7	12	18	...		
3	6	11	17	...			
4	10	16	...				
5	15	...					
⋮							

dove tail (coda di colomba): prende due numeri naturali e li converte in uno, è biunivoca

$$(x,y) \mapsto \left(\frac{1}{2} \; x^2+2xy+y^2+3x+y\right)$$

$$n \mapsto \left(n-\frac{1}{2}k(k+1),\; k-\left(n-\frac{1}{2}k(k+1)\right)\right)$$

con $k := \left\lfloor \frac{1}{2}\left(\sqrt{1+8n}-1\right)\right\rfloor$

<u>FUNZIONE</u>

Dati due insiemi A, B definisco $f: A \to B$ un sottinsieme del prodotto cartesiano $A \times B$ t.c.

$f: A \to B$

$(a,b),(a,c) \in f \Rightarrow b=c$

<u>totali</u>: $\forall a \in A.\ \exists b \in B : f(a)=b$

dominio di f: $Dom(f) = \{a \in A \mid \exists b \in B : f(a)=b\} =$

$= \{a \in A \mid f(a) \downarrow\}$

immagine di f: $Imm(f) = \{b \in B \mid \exists a \in A : f(a)=b\}$

f è iniettiva sse $\forall a, a' \in A.\ a \neq a' \Rightarrow f(a) \neq f(a')$

suriettiva $\forall b \in B.\ \exists a \in A.\ f(a)=b$

biunivoca se è sia iniettiva sia suriettiva

esempi: la funzione somma, la divisione per zero sono totali

$$gb(n) = \begin{cases} 0 & \text{se la congettura di Goldbach è vera} \\ 1 & \text{alt} \end{cases}$$

è T-calcolabile?

Sì: esiste una macchina di Turing che restituisce sempre zero, una che restituisce sempre 1: la funzione è calcolabile (non importa non sapere quale sia la macchina corretta, è importante che esista)

FUNZIONI RICORSIVE

eq $\begin{cases} !(n+1) = !n\,(n+1) \\ !(0) = 1 \end{cases}$

$\lambda x.$ significa che x è argomento della funzione

$\lambda x.(x+y)$ y variabile libera

$\lambda x,y.(x+y)$ funzione che fa la somma

(in opposizione a $somma(x,y) = x+y$ funzione con nome)

La classe delle funzioni ricorsive primitive $\mathcal{C}$ è la minima classe che soddisfa i seguenti schemi:

$\underline{I}$ zero: $\lambda x_1 \dots x_n . 0$ qualsiasi sia il numero di argomenti restituisce zero

$\underline{II}$ successore: $\lambda x. x+1$ associa a ogni valore il successore

$\underline{III}$ Identità-proiezione: $\lambda x_1 \dots x_n . x_i$ con $i \in [1;n]$

IV Composizione: $g_1 \dots g_n \in \mathcal{C}$ funzioni in m variabili,

$h \in \mathcal{C}$ funzione in k variabili

$\lambda x_1 \dots x_n . h(g_1(x_1 \dots x_n), \dots, g_k(x_1 \dots x_n)) \in \mathcal{C}$

IV Ricorsione primitiva: $h \in \mathcal{C}$ in $k+1$ variabili

$g \in \mathcal{C}$ in $k-1$ variabili

$f \in \mathcal{C}$ con $f :=$ $\begin{cases} f(0, x_2 \dots x_k) = g(x_2 \dots x_k) \\ f(x_1+1, x_2 \dots x_k) = h(x_1, f(x_1, x_2 \dots x_k), x_2 \dots x_k) \end{cases}$

Per induzione si può dimostrare che V termina

Occorre e basta che ci sia una successione finita o derivazione della forma: $f_1, f_2 \dots f_n$ t.c.

$(f_i \in \mathcal{C}$ per I, II, III$)$ $\vee$

$(f_i$ ottenibile mediante IV, V da f_j con $j<i)$

i.e. f_i è definita secondo gli schemi di base

o sulla base di funzioni già definite