Th. Un formalismo che esprime funzioni totali non le esprime tutte.

Dim:

1. Si immagina di scrivere tutte le funzioni primitive ricorsive;

   f_n := funzione ottenuta da n-esima derivazione

   (suppongo di elencarle in ordine lessicografico)

2. Definisco $g(x) = f_x(x) + 1$

3. g non appartiene all'elenco: $\forall n.\ f_n(n) \neq g(n)$

   (ie restituisce sempre un valore diverso)

   $g(n) = f_n(n) + 1 \neq f(n)$

Questo procedimento prende il nome di diagonalizzazione.

Si possono rendere totali tutte le funzioni parziali?

No, si dimostrerà in seguito.

Risulta inoltre che la dimostrazione precedente non è valida su funzioni parziali: $\varphi(x) := \psi_x(x) + 1$

non si può concludere che

$\forall n.\ \varphi(n) \neq \psi(n)$ in quanto possono essere entrambe non definite

Si aggiunge lo SCHEMA di MINIMIZZAZIONE per definire le funzioni parziali.

Si introduce la notazione $\mu\, n.\ I := \text{minimo } n \text{ t.c. } n \in I$

FUNZIONI RICORSIVE GENERALI ($\circ$ μ-RICORSIVE)

La classe delle funzioni μ-ricorsive è la minima classe $\mathcal{R}$ che soddisfa le condizioni:

I-V per le funzioni ricorsive primitive

VI Minimizzazione:

$$\varphi(\vec{x}, y), \psi(\vec{x}) \in \mathcal{R},$$

$$\mu y.\,(\varphi(\vec{x}, y) = 0 \wedge \forall z \le y.\ \varphi(\vec{x}, z)\downarrow)$$

A parole: introduce la parzialità; la computazione potrebbe non terminare mai $\circ$ perché per ogni y esiste m_y t.c. $\varphi(\vec{x}, y) = m_y \neq 0$ $\circ$ perché per ogni valore $k \le y$, $\varphi(\vec{x}, y) = m_k \neq 0$ e $\varphi(\vec{x}, k)\uparrow$.

TESI di CHURCH TURING Le funzioni (intuitivamente)* calcolabili sono tutte e sole le T-calcolabili.

*descrivibili con un algoritmo

TEOREMA 1

Le funzioni calcolabili sono $\#\mathbb{N}$; anche le funzioni calcolabili totali sono $\#\mathbb{N}$,

Dim: Le funzioni calcolabili costanti sono tante quante quelle calcolabili

$\forall n \in \mathbb{N}$ costruisco MdT che restituisce n

$\Rightarrow$ Th

esistono funzioni non calcolabili.

Dim: Suppongo di elencare tutte le funzioni calcolabili $f: \mathbb{N} \to \{0,1\}$ e di metterle in una tabella.

La funzione definita come il complementare di quelle nell'elenco non è nell'elenco ed è calcolabile.

Si nota inoltre che $\#\{f: \mathbb{N} \to \mathbb{N}\} = 2^{\#\mathbb{N}}$,

mentre quelle calcolabili sono $\#\mathbb{N}$.

Definisco i INDICE della macchina M_i che calcola la funzione φ (è l'indice della macchine, risultato di una enumerazione effettiva)

TEOREMA 2 Ogni funzione φ calcolabile ha $\#\mathbb{N}$ indici

$\forall \varphi$ funzione calcolabile. $\exists A$ insieme t.c. $\#A = \#\mathbb{N}$

e $\forall i \in A$. M_i calcola φ_i.

PADDING LEMMA A può essere costruito da una funzione ricorsiva primitiva a partire da uno degli indici di φ.

Dim: banalmente, si scrive un programma WHILE che calcola φ; aggiungo il comando skip una volta, due...

TEOREMA 3 FORMA NORMALE $\exists T(i,x,y)$ predicato,

$U(y)$ funzione calcolabile totale tale che

$\forall i,x.\ \varphi_i(x) = U(\mu y.\ T(i,x,y))$

Dim: Sia T predicato di Kleene:

$T(i,x,y) = 1$ se e soltanto se y è la codifica di una computazione terminante di $M_i(x)$

cioè $M_i(x) \longrightarrow^* (h, D, z)$

$U(y)$ restituisce z.

$\forall i, x.\ \varphi_i(x) = U(\mu y.\ T(i,x,y))$:

distinguo due casi

- $\varphi_i(x) = n \Rightarrow U(\mu y.\ T(i,x,y)) = n$

  $\Updownarrow$

  $\exists y$: $M_i(x) \longrightarrow^* (h, D, z)$, y codifica M_i

  vero per definizione

- $\varphi_i(x)\uparrow \Rightarrow \nexists y$: $M_i(x) \longrightarrow^* (h, D, z)$, y codifica M_i

  $\Rightarrow$ Th