! l'ordine dei quantificatori è importante $(\forall, \exists) \neq (\exists, \forall)$

NUMERABILITÀ

insieme di arrivo $\equiv$ codominio $\Rightarrow f(\mathbb{N}) \cong A$

A si dice **numerabile** se $\exists$ funzione suriettiva f → **funzione di enumerazione**

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

$\mathbb{N}$ è numerabile (funzione identità)
e quindi ogni suo sottoinsieme

Lemma

A numerabile. $\{*\} \oplus A$ è ancora numerabile. ($\oplus$ → somma/unione disgiunta)
Sia $f: \mathbb{N} \to A$ la funz di enumerazione di A. Definiamo g

$$g(x) = \begin{cases} g(0) = * \\ g(x+1) = f(x) \end{cases}$$

g è suriettiva

Corollario

A num., D finito. $D \oplus A$ è numerabile

Lemma

A, B num.. $A \oplus B$ è numerabile

$$h(x) = \begin{cases} h(2x) = f(x) \\ h(2x+1) = g(x) \end{cases}$$

! NON è un SISTEMA di eq., bensì NOTAZIONE per CASI

Corollario

Un unione finita di insiemi num. è num.

Corollario

A num., D finito. $D \times A$ è numerabile
→ prodotto cartesiano → (d_i, a_j)
└ nel finito → $|D \times A| = |D| \times |A|$

$$\underbrace{A \oplus A \oplus \ldots \oplus A}_{D \text{ volte}}$$

DOVETAILING

Si cerca una funz. biunivoca da $\mathbb{N} \times \mathbb{N}$ in $\mathbb{N}$, definendo una politica di visita delle coppie $\langle i, j \rangle$, ispezionandole una sola volta al passo k. Invertendo la funzione si ha l'enumerazione del prodotto cartesiano

$j \backslash i$	0	1	2	3	4
0	0	1	3	6	10
1	2	4	7	11	
2	5	8	12		
3	9	13			
4	14				

$$\langle i, j \rangle = j + \underbrace{\sum_{k=0}^{i+j} k}_{} = j + \frac{(i+j)(i+j+1)}{2} = \frac{(i+j)^2 + i + 3j}{2}$$

punti nella nuova diagonale   punti del piano visitati nelle diagonali

⊕ abbiamo un modo per "compattare" una coppia di naturali su un unico naturale

⊖ dipende in maniera quadratica dalla somma $i+j$ quindi, dato che la rappr. bin. richiede $\log_2$ bit, la "compressione" cresce come $i+j$, quindi senza risparmiare spazio

$$\log_2 \frac{(i+j)^2 + i + 3j}{2} \geq \log_2 (i+j)^2 = 2 \log_2 i+j$$

⊕ funzioni unarie e n-uarie possono essere astratte allo stesso modo nella calcolabilità

Lemma

$A \times B$ è num.

Lemma

L'unione di un insieme numerabile di insieme numerabile è ancora numerabile.

A num., f sua funz. di enum.

$\{B_a \mid a \in A\}$ collezione di insiemi num., ognuno enumerato da g_a

$$h : \mathbb{N} \times \mathbb{N} \to \bigcup_{a \in A} B_a$$

$$h(n,m) = g_{f(n)}(m) \qquad f(n) = a$$

è suriettiva

Lemma

A num., anche $\bigcup_{i \in \mathbb{N}} A^i$ è num. ($A^i = \underbrace{A \times A \times \ldots \times A}_{i \text{ volte}}$)

$\hookrightarrow \forall i$

Osservazione

Con A finito, $\bigcup_{i \in \mathbb{N}} A^i$ è l'insieme di tutte le stringhe definibili sull'alfabeto A (A^i rappresenta tutte le stringhe di lunghezza i)

con A finito: $|A| = n$ $|\mathcal{P}(A)| = 2^n$

Corollario

L'insieme delle parti finite di un insieme numerabile è num.

$\neq \mathcal{P}(A) \hookrightarrow$ tutti i possibili sottoinsiemi finiti (e.g. non i pari in $\mathbb{N}$)

$\hookrightarrow$ cresce in modo esponenziale

TEOREMA di CANTOR

T.

Dato un insieme arbitrario A, non può esistere una funzione suriettiva da A in $\mathcal{P}(A)$.

(ponendo $A = \mathbb{N}$, esce per definizione che $\mathcal{P}(\mathbb{N})$ non è numerabile)

ⓘ la crescita della cardinalità è troppo forte per mantenere la num.

DIM.

Supponiamo $\exists$ funz. suriettiva $g: A \to \mathcal{P}(A)$. Consideriamo

$$\Delta = \{a \in A \mid a \notin g(a)\} \qquad \Delta \subseteq A$$

$\rightarrow$ tecnica diagonale

$\hookrightarrow$ insieme degli elem. che non appartengono alla propria enumerazione

Siccome abbiamo supposto g suriettiva deve esistere δ t.c. $g(\delta) = \Delta$ (in quanto $\mathcal{P}(A)$ contiene tutti i possibili sottoinsiemi)

$$\delta \in \Delta \iff \delta \notin g(\delta) \quad \text{per def. di } \Delta$$
$$\iff \delta \notin \Delta \quad \text{per def. di } \delta$$

ASSURDO

Quindi g non può essere suriettiva

e.g.

$A = \{0,1,2\}$

TECNICA DIAGONALE

a	0	1	2	
$g(0)$	0	0	0	$= \{\}$
$g(1)$	1	0	0	$= \{0\}$
$g(2)$	0	1	1	$= \{1,2\}$

$a \notin g(a) \rightarrow 1\ 1\ 0$ $\overline{g(a)}$

$\Delta = \{0,1\}$

nuovo elemento che non sta nell'enumerazione

PARADOSSO di RUSSEL

Sia $U = \{x \mid x \notin x\}$. Allora $U \in U \Leftrightarrow U \notin U$ <u>ASSURDO</u>
↓
bisogna abbandonarlo

Principio di compressione (Cantor)

$P(t) \Leftrightarrow t \in \{x \mid P(x)\}$ sembra
insieme $\Leftrightarrow$ proprietà innocuo/intuitivo

DEFINIBILITÀ

La nozione stessa di definibilità non è ben definita
└ dipende dal formalismo scelto → è per forza limitato

Siccome per definire si usa una stringa (num.), le funz. definibili da $\mathbb{N}$ in $\mathbb{N}$ sono una q.tà num. Fissiamo f_i enumerazione

$g(x) = f_x(x) + 1$ definisco funzione con tecnica diagonale

Dovrebbe essere presente nell'enumerazione, ad es. $g = f_K$

$f_K(K) = g(K) = f_K(K) + 1$ <u>ASSURDO</u>

Quindi due possibilità:
- la funzione che ho definito non era presente nell'enum., quindi il linguaggio considerato è incompleto.
  Si ammette l'esistenza di funz. ben definite, non definibili nel ling. scelto. Questo vale $\forall$ linguaggio scelto, per via della diagonalizzazione.
- la nozione di definibilità non è ben definita

PARADOSSO di BERRY

"Sia n il più piccolo intero positivo non definibile con meno di 100 caratteri"
→ < di 100 caratteri <u>ASSURDO</u>

RICORSIONE PRIMITIVA

FUNZIONI INIZIALI

$\vec{x} = (x_1, x_2, \dots, x_k)$

- funzioni costanti

$$c^k_m(\vec{x}) = m$$

- proiezioni (identità generalizzate)

$$\pi^k_i(\vec{x}) = x_i$$

- successore

$$s(x) = x+1$$

SCHEMI COMPOSIZIONALI

- Composizione

se $h: \mathbb{N}^n \to \mathbb{N}$, $g_1, g_2, \dots, g_n : \mathbb{N}^k \to \mathbb{N}$ $\in \mathcal{L}$, allora $f: \mathbb{N}^k \to \mathbb{N} \in \mathcal{L}$

$$f(\vec{x}) = h(g_1(\vec{x}), g_2(\vec{x}), \dots, g_n(\vec{x}))$$

- Ricorsione primitiva

Ho bisogno di un argomento su cui iterare, posso avere poi un vettore di argomenti ausiliari

se $g: \mathbb{N}^k \to \mathbb{N}$, $h: \mathbb{N}^{k+2} \to \mathbb{N}$ $\in \mathcal{L}$, allora $f: \mathbb{N}^{k+1} \to \mathbb{N} \in \mathcal{L}$

$$f: \begin{cases} f(0, \vec{x}) = g(\vec{x}) \\ f(y+1, \vec{x}) = h(y, \underbrace{f(y, \vec{x})}_{\text{risultato della chiamata ricorsiva}}, \vec{x}) \end{cases}$$

- g gestisce il caso base
- h " " " induttivo

DEF.

Una funz. f è primitiva ricorsiva sse $\exists f_1, f_2, \dots, f_n = f$ t.c. $\forall f_i$ o è una funz. base oppure è ottenuta da funz. f_j con $j < i$ tramite composizione o ricorsione primitiva

FUNZIONI PRIMITIVE

- Addizione

$f_1(x) = x$
$f_2(x) = x+1$
$f_3(x,y,z) = y$
$f_4(x,y,z) = f_2(f_3(x,y,z)) = y+1$
$f_5(x,y,z) = \begin{cases} f_5(0,x) = f_1(x) = x \\ f_5(y+1,x) = f_4(y, f_5(y,x), x) = f_5(y,x)+1 \end{cases}$

$f \equiv f_6 = f_5(f_1(x), f_1(y)) = f_5(x,y)$

- Moltiplicazione

$mult(0,x) = 0$
$mult(y+1,x) = add(x, mult(y,x))$

- Predecessore

$pred(0) = 0$
$pred(y+1) = y$

- Fattoriale

$fatt(0) = 1$
$fatt(y+1) = mult(s(y), fatt(y))$

- Zero

$zero(0) = 1$
$zero(y+1) = 0$

- Sottrazione

$sub(0,m) = m$
$sub(n+1,m) = pred(sub(n,m))$

- Confronto

$eq(n,m) = zero(add(sub(n,m), sub(m,n)))$

SOMME e PRODOTTI LIMITATI

$$\sigma_f(z,\vec{x}) = \sum_{y \le z} f(y,\vec{x}) : \begin{cases} \sigma_f(0,\vec{x}) = f(0,\vec{x}) \\ \sigma_f(z+1,\vec{x}) = \sigma_f(z,\vec{x}) + f(z+1,\vec{x}) \end{cases}$$

$$\pi_f(z,\vec{x}) = \prod_{y \le z} f(y,\vec{x}) : \begin{cases} \pi_f(0,\vec{x}) = f(0,\vec{x}) \\ \pi_f(z+1,\vec{x}) = \pi_f(z,\vec{x}) * f(z+1,\vec{x}) \end{cases}$$

PREDICATI PRIMITIVI
Ritornano un valore booleano

Lemma

I predicati primitivi ricorsivi sono chiusi rispetto ai connettivi logici

$$c_{\neg P}(\vec{x}) = 1 - c_P(\vec{x}) = sub(c_P(\vec{x}), 1)$$

$$c_{P \wedge Q}(\vec{x}) = c_P(\vec{x}) * c_Q(\vec{x}) = mult(c_P(\vec{x}), c_Q(\vec{x}))$$

Lemma

upper-bound

I p.p.r. sono chiusi rispetto alla quantificazione limitata
Ovvero, se $P(z, \vec{x})$ p.p.r., allora $\forall_{z \leq y} P(z, \vec{x})$ $\exists_{z \leq y} P(z, \vec{x})$ anche

$$c_{\forall z \leq y P}(z, \vec{x}) = \prod_{z \leq y} c_P(z, \vec{x})$$

APPLICAZIONI

- Divisibilità
  $divide(x, y) \longleftrightarrow \exists_{n \leq y}\, eq(mult(n, x), y)$
  // se x è divisore di y

- Primalità
  $prime(x) \Leftrightarrow x \geq 2 \wedge \forall y \leq x\, (divide(y, x) \Rightarrow y = 1 \vee y = x)$

- Calcolare n-esimo n° primo

$$\begin{cases} Pr(0) = 2 \\ Pr(n+1) = \min\{p \mid prime(p) \wedge (p > Pr(n))\} \end{cases}$$

MINIMIZZAZIONE LIMITATA (μ - ricorsione)

Ricerca del minimo intero positivo $\mu_y P(y)$ che soddisfa il predicato $P(y)$. Limitata se viene fornito un bound

si risolve con

È l'analogo del while, con suoi pregi e difetti (divergenza)

Lemma

Le f.p.r. sono chiuse rispetto alla minimizzazione limitata

$\mu_{y \leq z} R(y, \vec{x})$ è p.p.r. Dimostriamo:

$h(y, \vec{x}) = \forall_{w \leq y} \neg R(w, \vec{x})$ $\longrightarrow$ vale 1 per w che non soddisfano R

Suppongo $y_0 = \mu_{y \leq z} R(y, \vec{x})$ (soluzione)

Allora $h(y, \vec{x}) = \begin{cases} 0 & \text{per } y < y_0 \\ 1 & \text{per } y_0 \leq y < z \end{cases}$

Dunque $y_0 = \underbrace{\Sigma_{y<z}}_{\text{sommatoria di tutti gli } y \text{, sicuramente} < z} \forall_{w \leq y} \neg R(w, \vec{x})$ y_0 è pari al n° di interi $< z$ per cui h vale 1

COPPIE e PROIEZIONI

$$<i, j> = [(i+j)^2 + i + 3j] / 2$$

$fst(p) = \mu_{x \leq p} \exists_{y \leq p} <x, y> = p \quad \rightarrow i$

$snd(p) = \mu_{y \leq p} \exists_{x \leq p} <x, y> = p \quad \rightarrow j$

FIBONACCI

$$\begin{cases} fibo(0) = 1 \\ fibo(1) = 1 \\ fibo(n+2) = fibo(n) + fibo(n+1) \end{cases}$$

doppia chiamata $\Rightarrow$ non primitiva ricorsiva

$\downarrow$

creo funz. aux fibo' basata su codifica a coppie

$fibo'(0) = <1, 1>$

$fibo'(x+1) = <fibo(x+1), fibo(x+2)> =$

$= <snd(fibo'(x)), fst(fibo'(x)) + snd(fibo'(x))>$

primitiva ricorsiva

$\Downarrow$

$fibo(n) = fst(fibo'(n))$ $\rightarrow$ anche fibonacci è p.r.

RICORSIONE MULTIPLA

Si dice storia di $f(y,\vec{x})$

$$\hat{f}(y,\vec{x}) = \langle f(0,\vec{x}), f(1,\vec{x}), \dots, f(y,\vec{x}) \rangle_{y+1}$$

$$\equiv \langle\langle f(0,\vec{x}), f(1,\vec{x}) \rangle, \dots \rangle, f(y,\vec{x}) \underbrace{\rangle_2 \rangle_2 \dots \rangle_2}_{y}$$

Che permette di definire il seguente schema di ricorsione

$$\begin{cases} f(0,\vec{x}) = g(\vec{x}) \\ f(y+1,\vec{x}) = h(y, \hat{f}(y,\vec{x}), \vec{x}) \end{cases}$$

È ricorsiva primitiva? Sì, infatti:

$$\begin{cases} \hat{f}(0,\vec{x}) = f(0,\vec{x}) = g(x) \\ \hat{f}(y+1,\vec{x}) = \langle \hat{f}(y,\vec{x}), f(y+1,\vec{x}) \rangle = \langle \hat{f}(y,\vec{x}), h(y, \hat{f}(y,\vec{x}), \vec{x}) \rangle \end{cases}$$

Infine:

$$\begin{cases} f(0,\vec{x}) = \hat{f}(0,\vec{x}) \\ f(y+1,\vec{x}) = snd(\hat{f}(y+1,\vec{x})) \end{cases}$$

RICORSIONE di CODA e ITERAZIONE

T.

Le funz. prim. ricors. sono tutte e solo quelle for-calcolabili, cioè esprimibili in un linguaggio imperativo (del primo ordine) utilizzando l'if, il for e chiamata di funzione non ricorsiva.

[slide 36]

[NO 37
38
39]

FUNZIONE di ACKERMANN

$$
\begin{aligned}
ack(0,0,y) &= y\\
(0,x+1,y) &= ack(0,x,y)+1\\
(1,0,y) &= 0\\
(z+2,0,y) &= 1\\
(z+1,x+1,y) &= ack(z, ack(z+1,x,y), y)
\end{aligned}
$$

NON è esprimibile nel formalismo primitismo ricorsivo

Con un ordinamento lessicografico degli argomenti, intuitivo dimostrare la terminazione. Gli argomenti mandati in ricorsione sono complessivamente più piccoli.

- $ack(0,x,y) = x+y$
- $ack(1,x,y) = x * y$
- $ack(2,x,y) = y^x$
- $ack(3,x,y) = \left.y^{(y^{\cdot^{\cdot^{\cdot y)}}}}\right\}$ x volte

La funz. di Ackermann ha una complessità che trascende il potere espressivo del linguaggio primitivo ricorsivo

iterare un funzionale (funz. di ord. sup.)

↳ ogni z successiva comporta un'iterazione del livello precedente → crescita asintotica >> di una qualsiasi f.p.r.

Esistono formalismi (totali) che consentono di scrivere la funz. di Ackermann.

IL PROBLEMA dell' INTERPRETE

Sia data numerazione effettiva (e.g. lessicografica) P_n dei programmi del linguaggio $\mathcal{L}$, e sia φ_n la funz. calcolata da P_n.

$I(n,m) = \varphi_n(m)$ simula P_n con input m

$\exists u$ t.c. $I = \varphi_u$? tecnica diagonale di Cantor

ovvero, un programma $\equiv$ interprete ?

DIM.

Supponiamo $\exists u \mid I(n,m) = \varphi_u(n,m) = \varphi_n(m)$

Definisco

$$f(x) = \varphi_u(x,x) + 1 = \varphi_x(x) + 1$$

Se il ling. $\mathcal{L}$ è chiuso, $f \in \mathcal{L}$ per composizione, quindi $\exists i \mid \varphi_i = f$

Ma allora

$$\varphi_i(i) = f(i) = \varphi_i(i) + 1$$

ASSURDO $\rightarrow$ l'interprete non è calcolabile

TEOREMA

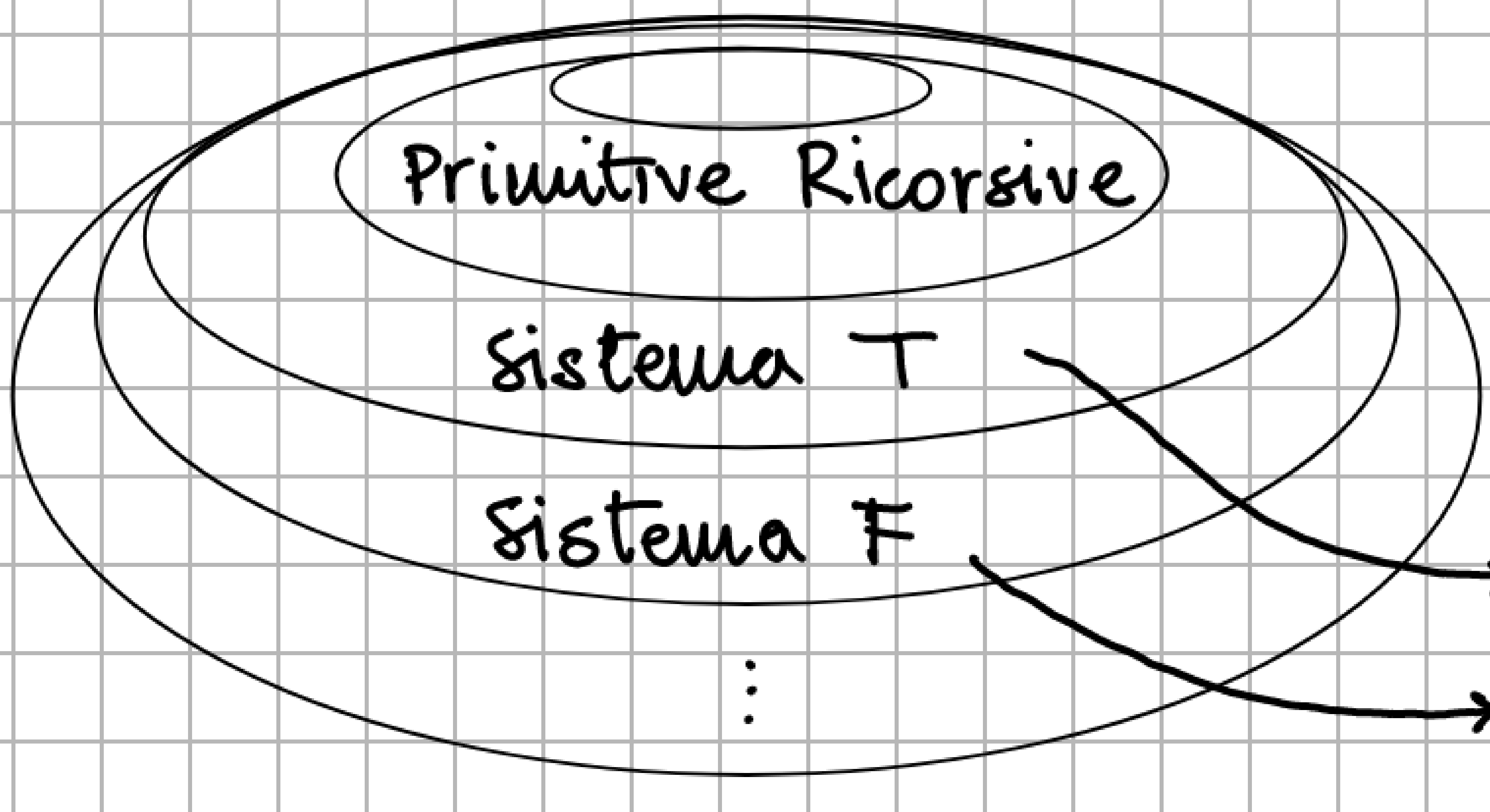
Nessun formalismo totale è in grado di esprimere il proprio interprete

$\hookrightarrow$ se non lo fosse, non sarebbe assurdo (non terminante) $\rightarrow \varphi_i$ non definito

e.g. Java, Fortran, C ? $\rightarrow$ devono ammettere divergenza

linguaggi parziali

GERARCHIE di LINGUAGGI



Verrebbe da aspettarsi una gerarchia infinita, in cui ogni volta l'interprete sfugga dal potere espressivo del proprio linguaggio

aritmetica del primo ordine

" " secondo "

COSTRUTTI potenzialmente DIVERGENTI
- goto
- while
- minimizzazione (μ-ricorsione illimitata)
- ricorsione generale
- costrutti autoreferenziali (auto-applicazione, auto-interpretaz.)
- ...

I ling. parziali ammettono la definizione del proprio interprete

Abbiamo una gerarchia infinita di ling. parziali con potere espressivo crescente?
Probabilmente no... Si sono studiati molti modelli computazionali diversi e si è dimostrato che fossero tutti equivalenti.
Le funzioni esprimibili sono dette funzioni calcolabili

LA TESI di CHURCH
Le funzioni calcolabili sono esattamente quelle intuitivamente calcolabili mediante una procedura effettiva di calcolo.

È una definizione che mostra che il concetto di calcolabilità è indipendente dal formalismo.

NON può essere dimostrata. (si mangia la coda)

Esistono funzioni non calcolabili? Terminazione, correttezza, meta-analisi, ...

MACCHINA di TURING

HW
- nastro di memoria illimitato, diviso in celle di dimensione fissa
- testina mobile
- automa a stati finiti

OP
- r/w della cella sotto la testina
- L/R della testina
- modificare lo stato interno dell'automa

One tape, deterministica

$\langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$

- Q insieme finito di stati
- Γ alfabeto finito del nastro
- $b \in \Gamma$ carattere bianco
- $\Sigma \subseteq \Gamma$ caratteri di I/O
- $q_0 \in Q$ stato iniziale
- $F \subseteq Q$ insieme degli stati finali
- δ: è la funzione di transizione

$Q/F \times \Gamma \to Q \times \Gamma \times \{L, R\}$

↓

ha un grafo finito formato da quintuple $(q, a, q', a', M) \mid \delta(q,a) = (q', a', M)$
≡ insieme delle istruzioni macchina (programma) e la determina univocamente (la macchina di Turing)

CONVENZIONI I/O

Input
- nastro inizializzato con la stringa di input
- testina sul primo blocco
- tutte le altre celle inizializzate a b

Output
- nel momento in cui la macchina si arresta, l'output è la più lunga stringa $\Sigma \setminus b$ alla destra della testina

↳ altrimenti 2 nastri separati per I/O

CONFIGURAZIONI ISTANTANEE

Tutte le informazioni della computazione necessarie (da salvare) per interrompere l'esecuzione e riprenderla in un secondo momento. È una descrizione dello stato della computazione. (! ≠ stato dell'automa, comprende anche la memoria) in un determinato istante

Configurazione, è una tripla:

$$\sigma, q, \tau$$

- σ: stringa a sx della testina non definitivamente bianca
- q: stato dell'automa
- τ: stringa a dx (prossimo carattere da leggere)

Contiene quindi 3 info: memoria, stato interno della macc., posizione della testina (implicita nella divisione sx/dx)

La computazione avviene per passi discreti: transizione tra due configurazioni è una relazione $\vdash$ governata dalla funzione di transizione

$$\sigma, q, a\tau \vdash \sigma a', q', \tau \qquad \text{se } \delta(q,a) = (q', a', R)$$

$$\sigma c, q, \underset{\hat{}}{a}\tau \vdash \sigma, q', \underset{\hat{}}{c}a'\tau \qquad (q', a', L)$$

pos. testina

Siccome le transizioni sono omogenee, sono iterabili: la relazione $\vdash^*$ denota la chiusura transitiva e riflessiva di $\vdash$ (mi assicura che due configurazioni legate da $\vdash^*$ siano riconducibili in un numero arbitrario di passi)

Una funz. $f : \Sigma^* \to \Sigma^*$ è calcolata da una macchina di Turing M, se $\forall \alpha \in \Sigma^*$ (stringa) $\exists q_f \in F \mid$

$$\epsilon, q_0, \alpha\tau \vdash^* \sigma, q_f, \tau$$

$f(\alpha)$ è il più lungo prefisso di $\tau \in \Sigma^*$

L'ESSENZA della MACCHINA di TURING

Agente di calcolo con potenzialità finite, che procede per passi discreti. Ad ogni passo posso:

- prendere visione di una porzione finita dello spazio (discretizzato) circostante (compreso un eventuale stato interno)
- modificare una porzione finita dello spazio
- spostarmi di una distanza finita

LA MACCHINA di TURING UNIVERSALE

Dato che le MdT hanno una forte correlazione hw con la funz. calcolata, la MdT Universale rappresenta il simulatore di una qualsiasi altra MdT (avvicinandosi quindi alla macchina di Von Neumann, caricando le varie MdT/programmi in memoria)

tabelle di quintuple → funzione di transizione
↳ salvabile su nastro
↳ ulteriore nastro per salvare lo stato interno della MdT simulata M

Se leggo l'insieme delle quintuple come la codifica numerica (indice) di M, la MdTU è un interprete per essa.

NUMERAZIONI di GÖDEL

NUMERAZIONE delle FUNZIONI CALCOLABILI

Dato un formalismo Turing-completo, possiamo enumerare i programmi P_i, con funzione φ_i calcolata e quindi enumerare tutte le funz. calcolabili

NUMERAZIONI ACCETTABILI

Sia data un'enumerazione φ_i^k delle funz. parziali calcolabili a k argomenti. Se l'enumerazione è effettiva, mi aspetto che ci sia un interprete.

Proprietà utm (Universal Turing Machine)

$\exists u \in \mathbb{N} \mid \forall x, y$ $\quad \varphi_u(x,y) = \varphi_x(y) \equiv P_x(y)$

φ_u interprete

Inoltre, ci aspettiamo un modo effettivo per determinare i codici dei programmi ottenuti per valutazione parziale di programmi dati

Proprietà smn

Si tratta di istanziare funzioni e poi di far vedere che le istanze si possono ottenere in modo parametrico ed effettivo in funzione del parametro su cui sto istanziando.

CURRYFICAZIONE

Isomorfismo tra funzioni binarie e famiglia di funzioni unarie

$$A \times B \rightarrow C \approx A \rightarrow (B \rightarrow C)$$

Gli isomorfismi a livello di insiemi sono interessanti perché consentono di trasformare i tipi di dato (rappresentazione dell'informazione)

```
def currify(f):
    def fa(a): return (lambda b: f(a,b))
    return fa

def mul(a,b): return a*b

mulc = currify(mul)
mulc4 = mulc(4)
mulc4(9)  # returns 36
```

indice della funzione

$A \times B \rightarrow C \approx A \rightarrow (B \rightarrow C)$

$a, b \mapsto f(a,b)$ $\quad a \mapsto s(a)$

$\varphi_{s(a)}(b) = f(a,b)$

Generalizzando:

Per ogni funzione parziale calcolabile $\overbrace{f}^{m+n}$ (lunghezza degli argomenti) esiste una funzione totale calcolabile s_n^m, tale che, $\forall \vec{x}_m, \vec{y}_n$

$$\varphi_{s_n^m(\vec{x}_m)}(\vec{y}_n) = f(\vec{x}_m, \vec{y}_n)$$

$s_n^m(\vec{x}_m)$ indice della funzione

deve essere totale, la parzialità è ancora presente, ma in φ

Un'enumerazione che gode delle proprietà utm e smn è detta accettabile. Le due proprietà sono "complementari" l'una all'altra: utm mi dice, data la funzione, come calcolarla; smn mi dice come creare delle funzioni calcolabili (se sai calcolare una funz., sai anche calcolare tutte le istanze e ottenerle in modo effettivo). Sarebbero il corrispettivo nel λ-calcolo dell'applicazione e dell'astrazione.

RIDUCIBILITÀ di ENUMERAZIONI

Siano date due funzioni di enumerazione $\varphi, \psi : \mathbb{N} \to A$

1. ψ è riducibile a φ ($\psi \leq \varphi$), se esiste una funz. f totale calcolabile tale che, $\forall n \in \mathbb{N}$, $\psi_n = \varphi_{f(n)}$

2. ψ è equivalente a φ ($\psi \equiv \varphi$), se $\psi \leq \varphi \wedge \varphi \leq \psi$

TEOREMA di EQUIVALENZA di ROGER

Tutte le enumerazioni accettabili di funzioni parziali ricorsive sono equivalenti tra loro

NOTAZIONE

Sia φ_i un'enumerazione accettabile delle funzioni parziali calcolabili.

proprietà "puntuali" → input

$\varphi_i(n)\downarrow$ per indicare che la funz. è definita (converge) su n.

$\varphi_i(n)\uparrow$ indefinita (diverge)

Definiamo dominio (dom) il suo insieme di convergenza

$$dom(\varphi_i) = \{ n \mid \varphi_i(n)\downarrow \}$$

Il codominio è l'insieme dei suoi possibili output

$$cod(\varphi_i) = \{ m \mid \exists n, \varphi_i(n) = m \}$$

Le funzioni parziali sono ordinate parzialmente rispetto all'inclusione insiemistica dei loro grafi

$$\varphi_i \subseteq \varphi_j \Leftrightarrow \forall n \in dom(\varphi_i), \varphi_i(n) = \varphi_j(n)$$

Quando $\varphi_i(n)\uparrow$, $n \notin dom(\varphi_i)$, magari $\varphi_j(n)\downarrow$, quindi

φ_j estende φ_i (vuol dire che φ_i è "meno definita")

Se mi restringo al dominio della prima funzione, mi aspetto che coincidano, in più la 2ª funz. può convergere su elem. $\notin$ dom della prima.

IL PROBLEMA della TERMINAZIONE (generale)

Il test di terminazione è così definito:

$$h(i,x) = \begin{cases} 1 & \text{se } \varphi_i(x)\downarrow \\ 0 & \text{se } \varphi_i(x)\uparrow \end{cases}$$

complemento

Consideriamo la funz. f diagonalizzazione

$$f(x) = \begin{cases} 1 & \text{se } h(x,x)=0 \Leftrightarrow \varphi_x(x)\uparrow \\ \uparrow & \text{se } h(x,x)=1 \Leftrightarrow \varphi_x(x)\downarrow \end{cases}$$

terminazione diagonale: programma sul suo indice

Se h (totale) è calcolabile, anche f (parziale) lo è.
Dovrebbe dunque $\exists m \mid \varphi_m = f$.
Quanto vale $\varphi_m(m)$? $\longrightarrow$ chiusura della diagonalizz.

$$\varphi_m(m) = \begin{cases} 1 & \text{se } h(m,m)=0 \Leftrightarrow \varphi_m(m)\uparrow \\ \uparrow & \text{se } h(m,m)=1 \Leftrightarrow \varphi_m(m)\downarrow \end{cases}$$ ASSURDO

Il test di terminazione non è calcolabile
(il problema della terminazione non è decidibile)

IL PREDICATO T di KLEENE

Esiste un predicato $T(i,n,k,t)$ t.c.

1) $\varphi_i(n) = k \Leftrightarrow \exists t \geq k,\ T(i,n,k,t)$
2) La funz. caratteristica di T è calcolabile

T è il predicato (intuitivamente calcolabile) che afferma che φ_i su input n termina con risorse fissate t (tempo o spazio) restituendo k. Si suppone che le risorse necessarie a produrre k siano almeno pari ad esso.

$\Rightarrow$ È possibile decidere se una computazione si arresta in un tempo dato

Il predicato T fornisce una visione più fine della nozione di interp.

prendo solo k — cerco la minima coppia per cui valga — risorse

$\forall i,n$

$$\varphi_i(n) = fst\left(\mu_{(k,t)}\, T(i,n,k,t)\right)$$

BIG PICTURE

- linguaggi "deboli"
- primitive ricorsive
- sistema T (funz. di Ackermann) → funz. di cui posso dimostrare la totalità dentro aritmetica del 1° ordine
- sistema F
- ⊙ funzioni calcolabili totali
- funzioni calcolabili (parziali)
- ⊙ funzioni "definibili" (enumerabili) (problema della terminaz.)
- funzioni da N in N (cardinalità del continuo, più che enumerab.)

⊙ → non caratterizzabili sintatticamente (dipendono dal formalismo)

IL PROBLEMA della TOTALITÀ

$$total(i) = \begin{cases} 1 & \text{se } \varphi_i \text{ è totale} \\ 0 & \text{altrimenti} \end{cases}$$

totale → converge $\forall$ input

Consideriamo una funz. ausiliare:

unico modo per creare funzioni

$$f(x,y) = \begin{cases} 0 & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$$

f è calcolabile, per s.m.n $\exists h$ <u>totale calcolabile</u>, t.c.

$$\varphi_{h(x)}(y) = f(x,y) = \begin{cases} 0 & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$$

y non viene usato per creare una funzione costante, in questo modo se converge una volta è totale, altrimenti no

$$total(h(x)) = \begin{cases} 1 & \text{se } \varphi_x(x)\downarrow \\ 0 & \text{se } \varphi_x(x)\uparrow \end{cases}$$

⟹ questo è il problema della terminazione
⇓
non calcolabile

Quindi $total \circ h$ ← non calcolabile

total → non è calcolabile

h → è totale calcolabile

IL PROBLEMA dell'EQUIVALENZA ESTENSIONALE dei PROGRAMMI

$$eq(i,j) = \begin{cases} 1 & \text{se } \varphi_i \approx \varphi_j \\ 0 & \text{altrimenti} \end{cases}$$

considero m l'indice della funzione costante 0
considero la funzione h precedente

$$\varphi_{h(x)}(y) = \begin{cases} 0 & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{se } \varphi_x(x)\uparrow \end{cases}$$

Poniamo

$$f(x) = eq(h(x), m) = \begin{cases} 1 & \text{se } \varphi_x(x)\downarrow \\ 0 & \text{altrimenti} \end{cases}$$

problema della terminazione

non calcolabile

L'uguaglianza estensionale (significato) a partire dall'intensione (rappresentazione/descrizione) è sempre un grosso problema

TRE FUNZIONI SIMILI

funzione totale non calcolabile

1) $g(i) \equiv \begin{cases} 1 & \text{se } \exists n, \varphi_i(n)\downarrow \\ 0 & \text{altrimenti} \end{cases}$ c'è un input su cui converge? True/False

funzione parziale → calcolabile

2) $g'(i) \equiv \begin{cases} 1 & \text{se } \exists n, \varphi_i(n)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$ " " " " " " ? Sì / $\uparrow$

3) $g''(i) \equiv \begin{cases} \mu n, \varphi_i(n)\downarrow & \text{se } \exists n, \varphi_i(n)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$ qual è il più piccolo input su cui converge?

1] consideriamo la solita funzione calcolabile h

$$\varphi_{h(x)}(y) = \begin{cases} 0 & \text{se } \varphi_x(x)\downarrow \\ \uparrow & \text{se } \varphi_x(x)\uparrow \end{cases}$$

$$f(x) = g(h(x)) = \begin{cases} 1 & \text{se } \varphi_x(x)\downarrow \\ 0 & \text{altrimenti} \end{cases}$$ terminazione diagonale

↓

g non è calcolabile ← h è calcolabile, f non è calcolabile

2] Accortezza: muoversi con dovetailing tra i due infiniti di possibili valori in input e tempo di esecuzione

Sia $t(i,n,s)$ la funzione caratteristica del predicato di Kleene.

come se fosse un while → appartiene

$$g'(i) = \mu\langle n,s\rangle.\ t(i,n,s) = 1;\ \text{return } 1$$ → algoritmo calcolabile

3] Considero un h totale calcolabile che mi permetta di risolvere g''

$$\varphi_{h(i)}(y) = f(i,y) = \begin{cases} 0 & \text{se } y>0 \vee \varphi_i(i)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$$ mi chiedo se il più piccolo input sia 0 o 1 per $\downarrow$

$\varphi_i(0)\downarrow$ sse $\varphi_i(i)\downarrow$. Inoltre $\varphi_i(1)\downarrow$.

$$g''(h(i)) = \begin{cases} 0 & \text{se } \varphi_i(i)\downarrow \\ 1 & \text{altrimenti} \end{cases}$$ problema della terminazione diagonale

INSIEMI RICORSIVI

Per ragioni storiche, ricorsivo = calcolabile.
In particolare è calcolabile la funzione caratteristica.
(= trovare un'enumerazione effettiva degli elementi dell'insieme)
⚠ → definisce i r.e.

Due approcci possibili:
- GENERATIVO → meccanismo generativo e.g. grammatica
- DECISIONALE → algoritmo discriminante e.g. automa riconoscitore

entrambi i metodi definiscono implicitamente l'insieme infinito dei dati che voglio trasmettere (e.g. linguaggio) (e.g. linguaggio logico / formule dimostrabili dagli assiomi)

Un insieme si dice ricorsivo (decidibile) se la sua funzione caratteristica è calcolabile.

e.g.
- insieme vuoto e $\mathbb{N}$
- insiemi finiti
- insiemi definiti da predicati primitivi ricorsivi
- insiemi definiti da un sistema di calcolo

e.g. contrario
- problema della terminazione diagonale (definito ma non calcolabile)
  $K = \{x \mid \varphi_x(x)\downarrow\}$

PROPRIETÀ di CHIUSURA

Lemma
Gli insiemi ricorsivi sono chiusi rispetto alle operazioni di unione, intersezione e complementazione

Siano A e B insiemi ricorsivi $\Rightarrow \exists\, c_A, c_B$ funzioni caratteristiche
Allora:
- $c_{\overline{A}}(n) = 1 - c_A(n)$
- $c_{A \wedge B}(n) = \min\{c_A(n), c_B(n)\}$
- $c_{A \vee B}(n) = \max\{c_A(n), c_B(n)\}$

Gli insiemi ricorsivi, ordinati rispetto alla relazione di inclusione insiemistica, formano un reticolo booleano (Algebra di Boole)

INSIEMI RICORSIVAMENTE ENUMERABILI

Un insieme si dice r.e. se è vuoto oppure è il codominio di una funzione totale calcolabile (funzione di enumerazione)
$f: \mathbb{N} \to A$

Lemma
$\forall$ insieme ricorsivo è anche r.e.
Sia A ricorsivo e c_A la sua funz. caratteristica
Il caso in cui A è finito è banale.
Suppongo A infinito e scrivo la funzione di enumerazione

$$\begin{cases} f(0) = \mu y, \ c_A(y) = 1 \\ f(x+1) = \mu y, \ c_A(y) = 1 \wedge y > f(x) \end{cases}$$

$\to$ il più piccolo y per cui la funz. carat. mi da True
$\to$ deve essere > dell'elemento precedente

⚠ con A insieme vuoto o finito,
l'enumerazione sopra divergerebbe $\to$ per via della μ (che è appunto un while)
- $\emptyset \Rightarrow c_\emptyset(n) = 0 \ \forall n \in \mathbb{N}$
- insieme finito $\Rightarrow y > f(x)$ sempre falso per $n \geq |A|$

Se l'insieme è r.e., è anche ordinabile rispetto a qualche funzione di ordinamento

Lemma
Un insieme A è r.e. (ricorsivo) sse può essere enumerato in modo crescente
DIM.
- $\Rightarrow$ prova del lemma precedente
- $\Leftarrow$ suppongo A infinito, sia f funz. di enumerazione

$$c_A(x) = [f(\mu_y \ f(y) \geq x) = x]$$ equivalenza logica

(scandisco tutti i dati enumerati da f finché non ne trovo uno non inferiore a x: a questo punto interrompo la ricerca e verifico se il dato coincide)

Dato che f scandisce in maniera ordinata, se non ho trovato x, sicuramente non sarà più avanti nell'enum.

Il fatto (non ovvio) che c_A sia totale è conseguenza dell'infinità di A (e quindi nel trovare sempre un elemento > di $\forall x$)

algoritmi di decisione da algoritmi di enumerazione.

TEOREMA di COMPLEMENTAZIONE

Un insieme A è ricorsivo sse sia A che $\overline{A}$ sono r.e.
Ovvero, un insieme è calcolabile (è possibile trovare un algoritmo di decisione) se anche il suo complementare è enumerabile.
Lanciando in parallelo le due enumerazioni è possibile definire c_A: solo una delle due enumerazione (ed esattamente una) terminerà e determinerà l'appartenenza. Da notare che h "gira" su un solo thread e riassume le due enumerazioni.

DIM.

$\Rightarrow$ ovvio

$\Leftarrow$ suppongo che A e $\overline{A}$ siano enumerati da f e g

$$\begin{cases} h(2x) = f(x) \\ h(2x+1) = g(x) \end{cases}$$

h è suriettiva su $\mathbb{N}$

$$c_A(n) = \text{pari}(\mu y(h(y)=n))$$

sia pari(n) la c_{pari}
c_A è totale e calcolabile

Corollario

Se un insieme non è r.e., non lo è nemmeno il suo complementare.
Ovvero, non ci si può aspettare di poter enumerare gli elementi che non appartengono.

SEMIDICIBILITÀ

- A decidibile

$$f(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

è il dominio di convergenza di

- A semidecidibile

$$f(x) = \begin{cases} \downarrow & \text{se } x \in A \\ \uparrow & \text{se } x \notin A \end{cases}$$

associato all'idea di ricerca

Esistono insiemi semidecidibili (r.e.) ma non decidibili (ricorsivi).
Il complementare di un insieme semidecidibile non è né semi- né decidibile.

Teorema

calcolabile utilizzando Kleene

L'insieme $K = \{x \mid \varphi_x(x)\downarrow\}$ è r.e. (insieme dei programmi che $\downarrow$ sul proprio indice)

$\overline{K}$ non è né r.e. né ricorsivo.
K è un insieme fondamentale per la teoria della calcolabilità, siccome è legato al problema della terminazione.

CARATTERIZZAZIONI EQ. degli insiemi R.E.

Teorema

Sia $A \subseteq \mathbb{N}$. Le seguenti affermazioni sono equivalenti:

1. $A = \emptyset \vee \exists f : A = cod(f)$, f totale calcolabile
2. $\exists g : A = dom(g)$, g parziale calcolabile (funz. di semidecisione)
3. $\exists h : A = cod(h)$, h parziale calcolabile

DIM.

- $1 \Rightarrow 2$

  Sia $A = cod(f)$ per f tot. calc., poniamo:

  $$g(x) = \mu y\,(f(y) = x) \qquad g \text{ è calcolabile}$$

  $g(x)\downarrow$ sse $x \in cod(f)$ c.v.d.p.

- $2 \Rightarrow 3$

  Sia $A = dom(g)$, si considera:

  $$h(x) = x + 0 * g(x)$$ identità rispetto al dom di convergenza

  se $g(x)\downarrow$ pure $h(x)\downarrow$ e dà x

- $3 \Rightarrow 1$

  Sia $A = cod(h)$. Posto $a \in A$ e $h = \varphi_i$ considero:

  $$f(\langle x, k, s\rangle) = \begin{cases} k & \text{se } T(i, x, k, s) = 1 \\ a & \text{altrimenti} \end{cases}$$

  input output tempo/passi

  elemento di default

  mi assicura che $cod(f) = A$

ENUMERAZIONE degli insiemi R.E.

Possiamo definire un'enumerazione $W : \mathbb{N} \to RE$ dell'insieme RE di tutti gli insiemi r.e.

$$W_i = dom(\varphi_i)$$ → dominio di convergenza di una funzione parziale

$$K = \{x \mid \varphi_x(x)\downarrow\} = \{x \mid x \in dom(\varphi_x)\} = \{x \mid x \in W_x\}$$

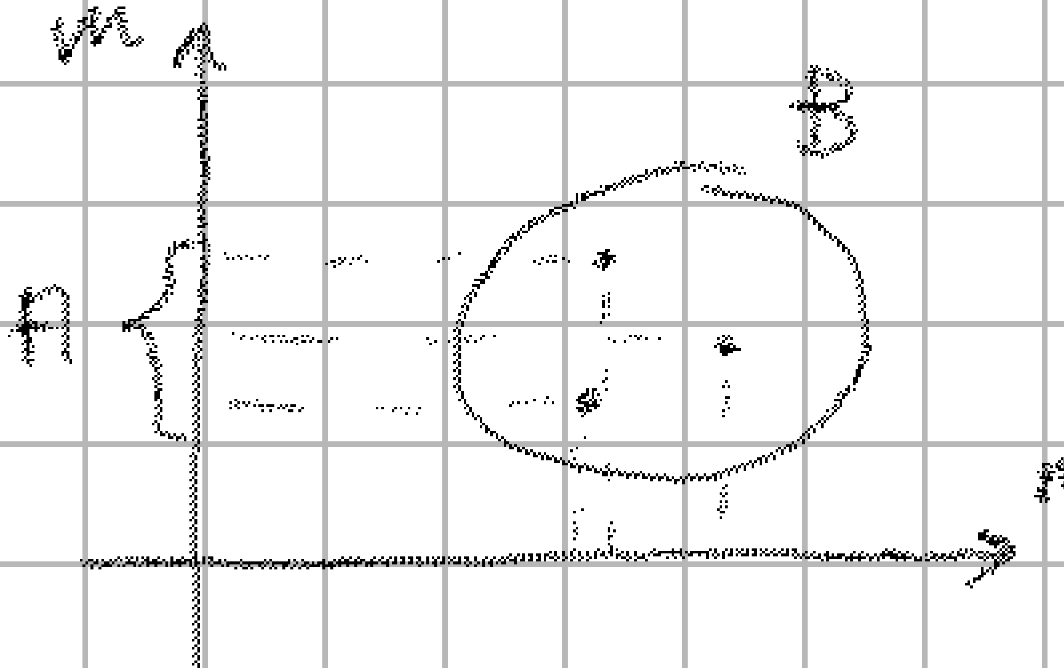
IL TEOREMA di PROIEZIONE

Un insieme A è r.e. sse $\exists B$ ricorsivo tale che

$$A = \{ m \mid \exists n, \langle n, m \rangle \in B \} \rightarrow$$ le proiezioni sono semidecidibili e quindi è possibile fare una ricerca

DIM.

$\Leftarrow$



Sia C_B funz. carat. di B, $A = dom(f)$

$$f(m) = \mu n, C_B(\langle n, m \rangle) = 1$$

$\Rightarrow$

Sia $A = dom(\varphi_i)$. Dunque $m \in A$ sse $\varphi_i(m)\downarrow$ sse $\exists n, T^3(i, n, m)$

tempo

$$B = \{ \langle n, m \rangle \mid T(i, n, m) \}$$

↳ fissato perché φ_i è un algoritmo di semidecisione di A (siccome è r.e.)

(vedremo che)

Un problema sta in NP se può essere visto come una proiezione esistenziale di P. (ricerca in uno spazio su cui ho una tecnica di decisione <u>efficiente</u> → polinomiale)

PROPRIETÀ di CHIUSURA degli insiemi R.E.

Teorema

Gli insiemi r.e. sono chiusi rispetto all'unione e all'intersezione, ma non rispetto alla complementazione.

DIM.

- unione

  funz. di enumerazione

  Sia $A = cod(f')$ e $B = cod(g')$ con f' e g' parziali calcolabili

  $A \vee B = cod(h')$

$$\begin{cases} h'(2x) = f'(x) \\ h'(2x+1) = g'(x) \end{cases}$$

- intersezione

  (algoritmo) funz. di decisione

  Sia $A = dom(f)$ e $B = dom(g)$ per f e g parziali calcolabili

  $A \wedge B = dom(h)$ per $h(x) = f(x) * g(x)$ $\quad h(x)\downarrow$ sse $f(x)\downarrow \wedge g(x)\downarrow$

- complementazione $\bar{K}$ non è r.e. (altrimenti K sarebbe ricorsivo)

UNIONI e INTERSEZIONI infinite

Lemma

1) una unione r.e. di insiemi r.e. è ancora r.e.
2) una intersezione r.e. di insiemi r.e. non è necessariamente r.e.

1) $\forall x \bigcup_{i \in W_x} W_i$ è r.e. 2) $\exists x \bigcap_{i \in W_x} W_i$ non è r.e.

L'intersezione è un'operazione più delicata dell'unione e si vede andando all'infinito.

PROPRIETÀ di CHIUSURA rispetto alle FUNZIONI

⋮

INSIEMI ESTENSIONALI (ciò che è calcolato)

Quali sono e come facciamo a decidere quali sono le proprietà delle funzioni calcolate da un programma?

$\varphi : \mathbb{N} \to PR$ (Partial Recursive $\to$ funzioni parziali $\equiv$ calcol.)
$\quad i \quad\;\; \varphi_i$

quindi devo risaltare φ e parlare di indici

Siamo interessati a scovare delle proprietà di PR, ovvero dei sottoinsiemi (e.g. funzioni costanti, totali, etc.)

$\varphi^{-1}(A)$ è la controimmagine di A, l'insieme di tutti quegli indici che porterebbero in A (tutti i programmi che mi calcolano la funz. corrispondenti ad A)

DEF.
Un insieme (proprietà) $A \subseteq \mathbb{N}$ si dice estensionale (w.r.t φ) se per ogni i, j
with reference to

$$i \in A \wedge \varphi_i \equiv \varphi_j \Rightarrow j \in A$$

Ovvero, se c_A è la funz. carat. di A

$$\varphi_i \equiv \varphi_j \Rightarrow c_A(i) = c_A(j)$$

Una proprietà estensionale di (indici di) programmi è una proprietà relativa alla funzione calcolata (estensione) e non alla forma o al modo (intensione) in cui questa viene calcolata

! Il complementare di un insieme estensionale è estensionale

proprietà di i
$P(i)$ estensionale
- φ_i è totale
- $\varphi_i \equiv f$
- $5 \in cod(\varphi_i)$
- $dom(\varphi_i)$ è finito
- $\varphi_i(0)\uparrow$
- $\exists n, \varphi_i(n)\downarrow \wedge \varphi_i(n+1)\downarrow$

} tutte non decidibili
$\downarrow$
alcune però, semi-decidibili

$\nexists$ un algoritmo che possa decidere se un φ qualsiasi abbia la proprietà

TEOREMA di RICE (1953)

Una proprietà estensionale di programmi è decidibile solo se è banale (insieme vuoto $\equiv \emptyset$ / universo $\equiv \mathbb{N}$)

DIM.

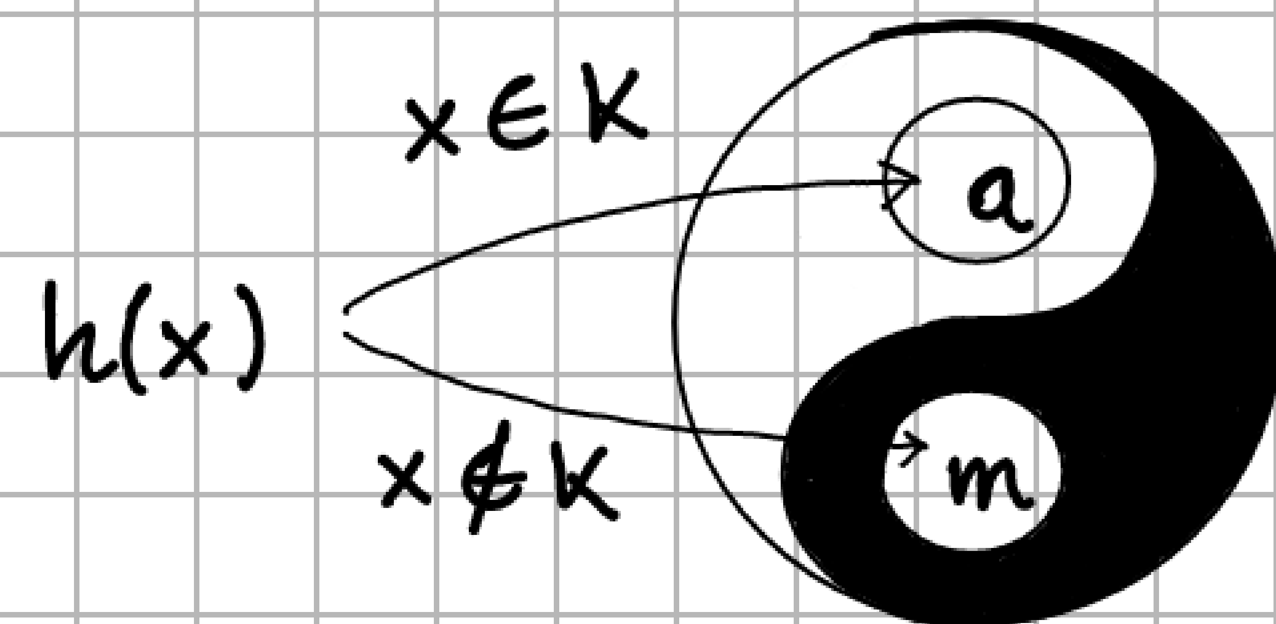
sia c la funz. carat. del predicato. Sia m un indice per la funzione ovunque divergente ($\varphi_m(n)\uparrow \ \forall \ \in \mathbb{N}$) e sia a t.c. $c(a) \neq c(m)$. Cerco h calcolabile t.c. indice che sta nella parte opposta

① $\varphi_{h(x)} \approx \begin{cases} \varphi_a & \text{se } x \in K \\ \varphi_m & \text{se } x \notin K \end{cases}$ riduco il problema a quello della terminazione diagonale

che si comporti come

se $x \in K$

$\hookrightarrow \varphi_x(x)\downarrow$



insieme di tutte le funzioni che hanno lo stesso comportamento (quindi la proprietà è mantenuta)

Consideriamo la funzione

$$\varphi_{h(x)}(y) = \varphi_x(x) \, ; \, \varphi_a(y)$$

composizione sequenziale: se e quando φ_x termina, lancio φ_a (il risultato di φ_x è ignorato)

Per smn, h è totale e calcolabile; è banale verificare ①
Dunque, utilizzando l'ipotesi di estensionalità, avremmo

$$c(h(x)) = \begin{cases} c(a) & \text{se } x \in K \\ c(m) & \text{se } x \notin K \end{cases}$$

sarei capace di risolvere il problema della terminazione

$\downarrow$

Quindi c non è calcolabile $\longleftarrow$ ASSURDO

Lo Yin Yang rappresenta proprio l'impossibilità di definire una divisione netta (e quindi l'impossibilità di decidibilità della proprietà). Due insiemi aperti non possono essere complementari.

USO del TEOREMA di RICE

- uso diretto, per dimostrare che determinate proprietà (essendo estensionali) non sono decidibili (quindi l'insieme non è ricorsivo)
- uso indiretto, per dimostrare che " " non sono nemmeno semidecidibili (dimostrando che il complementare è r.e.)

e.g. • $A = \{ i \mid \varphi_i(0)\downarrow \}$ non è banale → uso diretto di Rice
non è ricorsivo, però è r.e. (semidecidibile)
• $\overline{A} = \{ i \mid \varphi_i(0)\uparrow \}$
non è neppure r.e., altrimenti sia A che $\overline{A}$ sarebbero ricorsivi, contraddicendo Rice (uso indiretto)

MONOTONIA e COMPATTEZZA

Sia un insieme estensionale (rispetto a φ) di numeri naturali.

- A è detto monotono se per ogni i e j

$$i \in A \wedge \underbrace{\varphi_i \subseteq \varphi_j}_{\text{(estende)}} \Rightarrow j \in A$$

se un prog. i soddisfa la propr. allora anche tutte le sue estensioni

- A è detto compatto se $\forall i \in A \ \exists j \in A$ t.c.

(*) 1) il grafo di φ_j è finito → converge su un n° finito di elementi ≡ dom finito
2) $\varphi_j \subseteq \varphi_i$

→ mostra quanto sia debole il testing

Quindi:
- monotonia → tutte le estensioni stanno in A
- compattezza → esiste almeno una restrizione che sta in A

e.g.

insieme	MON.	COMP.
$\{ i \mid \varphi_i(0)\downarrow \}$	✓	✓
$\{ i \mid \varphi_i$ è totale $\}$	✓	✗
$\{ i \mid cod(\varphi_i)$ è finito $\}$	✗	✓
$\{ i \mid dom(\varphi_i)$ e $\overline{dom(\varphi_i)}$ sono infiniti $\}$	✗	✗

↳ convergono e divergono su infiniti punti

(*) $grafo(f) = \{ \langle n,m \rangle \mid m = f(n) \}$ $\quad dom(f) = \{ n \mid \exists \langle n,m \rangle \in grafo(f) \}$

$f \leq g \iff grafo(f) \subseteq grafo(g)$

RICE - SHAPIRO

T. MONOTONIA

ipotesi errata

Ogni insieme estensionale A r.e. è monotono

DIM.

non monotono

Suppongo indici i, j t.c. $\boxed{i \in A, j \notin A} \wedge \varphi_i \subseteq \varphi_j$
(ovvero nego la monotonia). Considero

estensione di

$$\varphi_{f(x)}(y) = \varphi_i(y) \,|\, (\varphi_x(x); \varphi_j(y))$$

composizione parallela: output è del 1° thread term.

È facile vedere che

$\overline{K}$ non è r.e.

ASSURDO

$$\varphi_{f(x)} \approx \begin{cases} \varphi_j & \text{se } x \in K \quad (j \notin A) \\ \varphi_i & \text{se } x \notin K \quad (i \in A) \Leftrightarrow f(x) \in A \Leftrightarrow x \in \overline{K} \end{cases}$$

Nel caso $x \in K$, non è possibile determinare quale dei due thread termini per primo, ma dato che $\varphi_i \subseteq \varphi_j$ se $\varphi_i(x)\downarrow$ anche $\varphi_j(x)\downarrow$. Però, dato che è un'estensione potrebbe darsi che $\varphi_i(x)\uparrow$ e $\varphi_j(x)\downarrow$, quindi si comporta come φ_j

T. COMPATTEZZA

Ogni insieme estensionale A r.e. è compatto

DIM.

A insieme estensionale r.e., suppongo $i \in A$ e $\forall j$ t.c. $\varphi_j \subseteq \varphi_i$, φ_j finito si abbia $j \notin A$. Considero f tot. calc. (per smn)

$$\varphi_{f(x)}(y) = \begin{cases} \uparrow & \text{se } \varphi_x(x)\downarrow \text{ in meno di } y \text{ passi} \\ \varphi_i(y) & \text{altrimenti} \end{cases}$$

restrizione finita di φ_i

Se $x \in \overline{K}$ allora $\varphi_{f(x)} \approx \varphi_i$ e dunque $f(x) \in A$
Se $x \in K$ allora $\varphi_x(x)$ terminerà in un numero finito di t passi e $\varphi_{f(x)}\downarrow$ per $y \leq t$
Dunque $f(x)$ è un indice per una sottofunzione finita di φ_i e per ipotesi $f(x) \notin A$

In conclusione $f(x) \in A \Leftrightarrow x \in \overline{K}$ e $\overline{K}$ sarebbe r.e.

ASSURDO

APPLICAZIONI di RICE - SHAPIRO
Permettono di dimostrare facilmente che determinati insiemi estensionali non sono r.e. (semidecidibili). e.g.

- $\{i \mid \varphi_i$ è totale$\}$ non è r.e. in quanto non è compatto
  l'intuizione è che non può essere testato nel finito
  (totale $\Rightarrow \downarrow \forall y$ in input)
- $\{i \mid cod(\varphi_i)$ è finito$\}$ non è r.e. in quanto non monotona. (mono-tona)

⚠ esistono insiemi monotoni e compatti che non sono r.e.
$\{i \mid dom(\varphi_i) \cap \overline{K} \neq \emptyset\}$
↳ monotona perché se l'$\cap \neq \emptyset$ anche $\forall$ sua estensione
↳ compatta perché basta restringere φ_i sul punto in cui c'è $\cap$

<u>DIM.</u>

...

TEOREMA del PUNTO FISSO di KLEENE (punto fisso "debole") $\approx$ e non $=$

Per ogni funzione f tot. calc. $\exists m$ t.c.

$$\varphi_{f(m)} \approx \varphi_m$$

trasformazione di programmi

prendendo per esempio $f(m) = s(m)$ $\Longrightarrow \varphi_m \approx \varphi_{m+1}$

DIM.

Per smn $\exists h$ tot. calc. t.c. (f è data)

$$\varphi_{h(x)}(y) = \underbrace{g(x,y)}_{\text{parziale calcolabile}} = \varphi_{f(\varphi_x(x))}(y)$$

auto-applicazione

Sia p un indice per h (dato che è tot. calc.) e poniamo

$m = \varphi_p(p) = h(p)$ (che è definito in quanto tot. calc.)

Allora, $\forall y$:

$$\varphi_m(y) = \varphi_{h(p)}(y) = g(p,y) = \varphi_{f(\varphi_p(p))}(y) = \varphi_{f(m)}(y)$$

c.v.d.

PUNTI FISSI e RICORSIONE

Ogni funzione ricorsiva è il punto fisso di un opportuno funzionale (funz. higher order / funz. tra funzioni)

e.g.

$$fact(x) = \text{if } x==0 \text{ then } 1 \text{ else } x * fact(x-1)$$

usando la lambda notazione

$$fact = \underbrace{\text{lambda } x :}_{\text{prendo in input } x} \text{ if } x==0 \text{ then } 1 \text{ else } x * fact(x-1)$$

$$F(g) = \text{lambda } x : \text{ if } x==0 \text{ then } 1 \text{ else } x * g(x-1)$$

↳ prendo g in input

allora:

$$fact = F(fact) \qquad (fact \text{ è un punto fisso di } F)$$

(calcolabile)

Se F è algoritmica, il teorema di Kleene assicura l'esistenza del punto fisso.

Presa una funz. ricors. è sempre possibile trovare un funzionale di cui è punto fisso. Non tutti i funzionali hanno un punto fisso però!

Voglio cercare $\varphi_m = F(\varphi_m)$

Se F è calcolabile, $\exists f$ tot. calc. t.c. $\varphi_{f(i)} = F(\varphi_i)$

Preso dunque un punto fisso di f

$$\varphi_m = \varphi_{f(m)} = F(\varphi_m)$$

! Il teorema del punto fisso richiede solo smn e interprete

⇓

L'interprete permette di simulare la ricorsione

↓

$$u(i, x) \equiv \varphi_i(x)$$

APPLICAZIONI del TEOREMA del PUNTO FISSO

L'uso del t. del punto fisso per simulare ricorsione è particolarmente "pulito", in quanto f che "implementa" F è estensionale, ovvero:

$$\varphi_i \equiv \varphi_j \Rightarrow \varphi_{f(i)} \equiv \varphi_{f(j)}$$

Tuttavia, il teorema è valido per qualunque trasformazione effettiva. Ad es.:

- in ogni enumerazione accettabile di programmi esistono sicuramente due programmi consecutivi con comportamenti identici

  $$\varphi_{i+1} \equiv \varphi_i$$

  DIM. punto fisso del successore

- esiste un programma che stampa se stesso, ovvero $\exists i$ t.c. $\varphi_i(0) = i$

  DIM. per smn $\exists h$ tot. calc. t.c. $\varphi_{h(x)}(y) = x$; se ne prenda un punto fisso

  QUINE (famoso esercizio di programmazione)
  non prende niente in input $\Rightarrow$ come prendere 0 in input
  $\hookrightarrow$ no info

DIMOSTRAZIONE ALTERNATIVA di RICE

Suppongo per assurdo A ricorsivo, ma non banale.
$\exists i, j$ t.c. $i \in A$, $j \notin A$. Considero

$$h(x) = \begin{cases} i & \text{se } x \in \bar{A} \\ j & \text{se } x \in A \end{cases}$$

Per def. $h(x) \in A \Leftrightarrow x \notin A$

Se A è ricorsivo, <u>h è tot. calc.</u> e, per Kleene, $\exists b \mid \varphi_b = \varphi_{h(b)}$

$$b \in A \Leftrightarrow h(b) \in A \Leftrightarrow b \notin A \qquad \text{ASSURDO}$$

SECONDO TEOREMA del PUNTO FISSO

Per ogni funz. binaria totale calcolabile f esiste una funz. calc. s tale che, $\forall y$

$$\varphi_{f(s(y),y)} \approx \varphi_{s(y)}$$

possibile generalizzarlo a funz. n-arie

<u>DIM.</u>

$$\varphi_{f(\varphi_x(x),y)}(z) = g(x,y,z) \quad \text{per smn } \exists r,h \text{ tot. calc.}$$

$$g(x,y,z) = \varphi_{h(x,y)}(z) = \varphi_{\varphi_{r(y)}(x)}(z)$$

Posto $s(y) = \varphi_{r(y)}(r(y))$, abbiamo $\forall z$:

$$\varphi_{s(y)}(z) = \varphi_{\varphi_{r(y)}(r(y))}(z) = \varphi_{h(r(y),y)}(z) =$$

$$= \varphi_{f(\varphi_{r(y)}(r(y)),y)}(z) = \varphi_{f(s(y),y)}(z)$$

NON CA CHIEDE ALL'ORALE

RIDUCIBILITÀ

Siano $A, B \subseteq \mathbb{N}$; A si dice riducibile (m-riducibile) a B (in simboli $A \leq_m B$) se $\exists f$ tot. calc. t.c.

$$x \in A \iff f(x) \in B$$

Due insiemi si dicono equivalenti (m-equivalenti, $A =_m B$) se $A \leq_m B \wedge B \leq_m A$

$A \leq_m B \wedge B \leq_m C$

$A \leq_m A$ $\qquad$ $A \leq_m C$

OSSERVAZIONI:

- la relazione $\leq_m$ è un preordine (i.e. riflessiva $\wedge$ transitiva)
- la relazione $=_m$ è di equivalenza
- $A \leq_m B \iff \overline{A} \leq_m \overline{B}$
- se $A \leq_m B$ e B è ricorsivo (decidibile) (w.r.t. r.e.) (semi-decidibile) allora A è ricorsivo
  ‼ non vale la stessa cosa per l'inclusione insiemistica (e.g. $\mathbb{N}$ è ricorsivo, $K \subseteq \mathbb{N}$ invece no)

$K_0 =_m K$

Sia $K_0 = \{\langle i, n\rangle \mid n \in W_i\}$ → problema della terminaz. generale

$n \in$ al dominio di convergenza di i, ovvero $\varphi_i(n)\downarrow$

- $K \leq_m K_0$. Siccome

$$i \in K \iff i \in W_i \leftrightarrow \langle i, i\rangle \in K_0$$

la funzione $f(x) = \langle x, x\rangle$ permette di ridurre K a K_0

- $K_0 \leq_m K$ Consideriamo la funz. tot. calc. h per cui

$$\varphi_{h(i,x)}(y) = g(i, x, y) = \varphi_i(x) \qquad \text{per smn}$$

Abbiamo

$$\langle i, n\rangle \in K_0 \iff n \in W_i \iff \forall y,\ \varphi_{h(i,n)}(y)\downarrow \iff \varphi_{h(i,n)}h(i,n)\downarrow$$
$$\iff h(i,n) \in K$$

Nonostante K_0 sembri più complicato di K, sono riducibili l'uno all'altro.

M-COMPLETEZZA

Un insieme si dice **m-completo** se è r.e. ed $\forall$ insieme r.e. è riducibile ad esso.

Lemma

K_0 e K sono insiemi completi.

Dato che $K_0 =_m K$ è sufficiente dimostrare la propr. per K_0

Abbiamo già dimostrato che se $A \leq_m K$ allora A è r.e., dunque $\exists i$ (indice) t.c. $A = W_i$. Allora, $\forall n$

↳ dom di convergenza di φ_i

$$\underbrace{n \in A \Leftrightarrow n \in W_i \Leftrightarrow \langle i, n \rangle \in K_0}_{\text{funzione di riduzione}}$$

Lemma

A è completo sse $A =_m K$

Se $A =_m K$ allora A è r.e. e completo perché lo è K

Viceversa, se A è m-completo, allora è r.e. e per la completezza di K, $A \leq_m K$; inoltre, siccome K è r.e., $K \leq_m A$ per la m-completezza di A.

INSIEMI PRODUTTIVI e CREATIVI

Sia $A \subseteq \mathbb{N}$.

1) A si dice **produttivo** se $\exists f$ tot. calc. t.c. $\forall i$

$$W_i \subseteq A \Rightarrow \underbrace{f}_{\text{funz. di produzione}}(i) \in A \setminus W_i$$

se A fosse r.e. $A \setminus W_i = \emptyset$

2) A si dice **creativo** se è r.e. ed il suo complemento $\bar{A}$ è prod.

L'idea è che, dato A non r.e., cercando di approssimarlo con W_i, grazie a una f applicata su W_i trovo un punto che sfugge alla mia approssimazione.

e.g. A = funzioni totali calcolabili W_i = funz. primitive ricorsive per diagonalizzazione possibile trovare un indice che non sta in W_i

TEOREMA
K è creativo (e la funzione di produzione è l'identità)

DIM.

Sappiamo che K è r.e., dimostriamo $\overline{K}$ produttivo, in particolare:

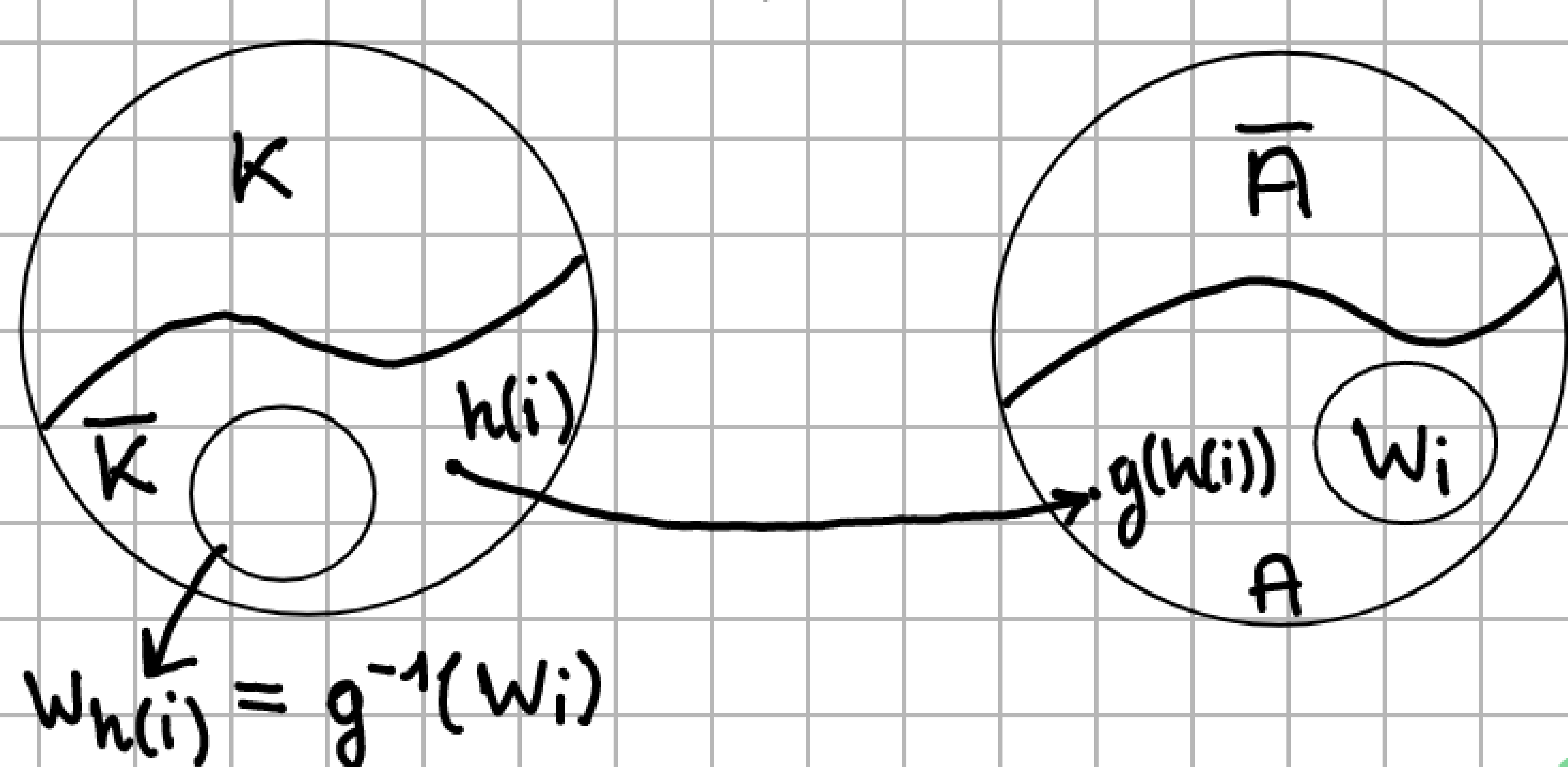
$$W_i \subseteq \overline{K} \Rightarrow i \in \overline{K} \setminus W_i$$

dominio di convergenza di i

- $i \in \overline{K}$. Infatti se $i \in K \Rightarrow i \in W_i$ e siccome $W_i \subseteq \overline{K}$, $i \in \overline{K}$, che è una contraddizione. Quindi $i \in \overline{K}$.
  dovuta all'ipotesi iniziale errata
- $i \notin W_i$. Infatti dovrebbe appartenere a K, ma abbiamo dimostrato il contrario.

CARATTERIZZAZIONE della PRODUTTIVITÀ
Teorema

Sia $A \subseteq \mathbb{N}$. A è produttivo sse $\overline{K} \leq_m A$



creatività ≡ completezza

Teorema

Sia $A \subseteq \mathbb{N}$. A è creativo sse $A =_m K$

DIM.
- Per def. A è creativo sse è r.e. e $\overline{A}$ è produttivo
  Ma A è r.e. sse $A \leq_m K$
- Per il teorema precedente $\overline{A}$ è produttivo sse $\overline{K} \leq_m \overline{A}$, ovvero $K \leq_m A$

DIM.

$\Rightarrow$) $W_{s(y)} = \begin{cases} \{f(s(y))\} & \text{se } y \in K \\ \emptyset \end{cases}$ $\longrightarrow$ che converga solo su questo punto

Voglio dimostrare che $f \circ s$ è una funz. di riduzione da K ad $\bar{A}$.

- se $y \in K$, allora $W_{s(y)} = \{f(s(y))\}$. Se $\{f(s(y))\} \in A$, allora $W_{s(y)} \subseteq A$ e quindi, per la produttività di A, $f(s(y)) \in A \setminus W_{s(y)}$ e in particolare, $f(s(y)) \notin W_{s(y)} = \{f(s(y))\}$ che è ASSURDO. Dunque $f(s(y)) \in \bar{A}$
- se $y \notin K$, allora $W_{s(y)} = \emptyset \subseteq A$ e dunque, per la prod., $f(s(y)) \in A \setminus W_{s(y)}$

Sia f funz. di produzione per A

$$\varphi_{h(z,y)}(n) = \begin{cases} 0 & \text{se } f(z) = n \wedge y \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

per smn tiro fuori famiglia di funzioni indicizzate effettivamente su z e y. h totale calcolabile

Per il secondo teorema di ricorsione $\exists s$ tot. calc. t.c.

$$\varphi_{s(y)}(n) = \varphi_{h(s(y),y)}(n) = \begin{cases} 0 & \text{se } f(s(y)) = n \wedge y \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

Dimostrata la calcolabilità di s.