1 Elementi di calcolabilità

1.1 Alcuni risultati classici

Teorema 1.1.1 (Numero funzioni calcolabili, esistenza funzioni non calcolabili)

1. Le funzioni calcolabili sono $\#(\mathbb{N})$; Inoltre le funzioni calcolabili **totali** sono $\#(\mathbb{N})$.
2. Esistono funzioni non calcolabili

....

Intuizione: Le funzioni calcolabili sono tante quante le macchine di Turing, che si possono enumerare. Inoltre, **ci sono più funzioni che funzioni calcolabili**

....

Dimostrazione.

1. Esistono *almeno* $\#(\mathbb{N})$ funzioni calcolabili (totali) poiché possiamo costruire $\#(\mathbb{N})$ MdT M_i che:
   - Svuotano il nastro
   - Scrivono la stringa $|^i$
   - Si arrestano

   Le funzioni calcolabili *non sono più di* $\#(\mathbb{N})$, poiché le MdT si possono enumerare.
2. Con una costruzione analoga a quella di Cantor (argomento diagonale) si vede che $\{f : \mathbb{N} \to \mathbb{N}\}$ ha cardinalità $2^{\aleph_0}$, quindi ci sono più funzioni di quante funzioni calcolabili.

$\square$

Teorema 1.1.2 (Padding Lemma)
Ogni funzione calcolabile φ_x ha $\#(N)$ indici, e l'insieme degli indici:

$$A_x \text{ t.c. } \forall y \in A_x \,.\, \varphi_y = \varphi_x$$

si può costruire mediante una **funzione ricorsiva primitiva**.

....

Intuizione: Posso inserire uno **skip** alla fine di un programma senza alterarne il comportamento; Per le MdT inserisco un nuovo stato ed una quintupla che non fa niente.

....

Dimostrazione. Per ogni MdT M_x, se $Q = \{q_0, \ldots, q_k\}$, si ottiene una nuova MdT M_{x_1}, con $x_1 \in A_x$, aggiungendo uno stato q_{k+1} ed una quintupla "che non fa niente": $(q_{k+1}, \#, q_{k+1}, \#, -)$.
Questo processo può essere ripetuto una quantità numerabile di volte. $\square$

Come si vede intuitivamente dallo schema di un interprete, **bastano un while e dei for** per rappresentare una funzione calcolabile. Detto in maniera più formale:

Si possono rappresentare le funzioni calcolabili come ***composizione di una funzione ricorsiva primitiva ed una funzione ricorsiva generale****.*

Teorema 1.1.3 (Forma Normale - Kleene I)
Esistono un predicato $T(i, x, y)$ ed una funzione $U(y)$ **calcolabili totali** e **ricorsive primitive** tali che:

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

Nota: Dato che T è totale **converge sempre**, quindi $\mu y.T(i, x, y)$ è una **funzione ricorsiva generale**.

. . . .

Intuizione: T è vero iff y è codifica di una computazione terminante della i-esima MdT; U dato l'indice y di una computazione terminante restituisce il risultato della computazione. Componendo le due funz. si ottiene il risultato della computazione terminante di indice minimo, che è risultato di $\varphi_i(x)$

. . . .

Dimostrazione. Si dimostra mostrando due funzioni che soddisfano il requisito:

- $T(i, x, y)$ è **il predicato di Kleene**:

  $T(i, x, y) = tt \iff$ y è la codifica di una computaz. **terminante** di M_i con dato iniziale x

  Il calcolo di T avviene tramite i seguenti passi:

  1. Dato i, recupera la macchina M_i
  2. Decodifica y
  3. Dato x si controlla se il risultato è una computazione terminante della forma $M_i(x) = c_0 \dots c_n$

  Questa sequenza di passi **termina sempre** $\implies$ T **totale**.

- Sia y la codifica della computazione **terminante** $M_i(x) \to^k (h, \triangleright z\#)$, allora U è tale che:

  $$U(y) = z$$

  I passi necessari a questo calcolo sono finiti (decodifica y, ottieni sequenza $c_0 \dots c_n$, accedi a c_n) e terminano tutti, quindi anche U è **totale**.

L'intero procedimento è ***effettivo***, quindi U e T sono **calcolabili** per la tesi di **Church-Turing**
Inoltre le due funzioni sono **ricorsive primitive** perché

- Le codifiche che usiamo lo sono
- I controlli effettuati lo sono

□

Corollario 1.1.3.1
Le funzioni T-calcolabili sono μ-ricorsive (per la **nota** del teorema).

Teorema 1.1.4 (Enumerazione, o della MdTU)

$$\exists z.\forall\ i, x\ .\ \varphi_i(x) = \varphi_z(i, x)$$

Ossia esiste una **funzione calcolabile parziale** φ_z "a due posti" che dati in input l'indice i di una funzione calcolabile parziale φ_i "ad un posto" ed il suo parametro x restituisce $\varphi_i(x)$.

....

Dimostrazione. Si usa $z : \varphi_z(i, x) = U(\mu y.T(i, x, y)) = \varphi_i(x)$, dove T è il predicato di Kleene □

....

Intuitivamente: M_z **recupera la descrizione** di M_i e la applica a x.

Applicazione parziale: data una funzione di due argomenti, posso *fissarne uno* ed ottenere una funzione ad un argomento, e.g.

$$f(x, y) = x + y \underset{\text{fisso } x=5}{\longrightarrow} f(y) = 5 + y$$

Teorema 1.1.5 (Del parametro, S_1^1)
Esiste una funzione s calcolabile totale ed **iniettiva** tale che:

$$\forall\ i, x\ .\ \varphi_{s(i,x)} = \lambda y.\varphi_i(x, y)$$

....

Intuizione: Basta fissare "in memoria" (o ambiente) il **parametro** x. L'indice della funzione con il parametro fissato dipende dalla scelta del parametro e dall'indice della funzione "a due posti".

....

Dimostrazione (sempre molto "intuitiva"). Si "predispone lo stato iniziale", fissando x in anticipo (e.g. si sostituisce l'operazione che legge il parametro x con $x := k$. Questa è una **procedura effettiva**); In questo modo abbiamo che una tale funzione esiste, ma è iniettiva?

Se non lo fosse, si può costruire s' che genera indici (che esistono e sono $\aleph_0$ per il padding lemma) in modo strettamente crescente. La funzione s' è strettamente crescente e totale, perciò è iniettiva. □

....

Generalizzazione: Data una funzione di $m + n$ parametri ne posso fissare m ed ottengo una funzione n-aria.

Teorema 1.1.6 (Del parametro, S_n^m – non dimostrato)
$\forall\ m, n \geq 0\ \exists\ s$ calcolabile totale **iniettiva** ad $m + 1$ posti t.c. $\forall i, (x_1, \ldots, x_m)$:

$$\varphi_s^{(n)}(i, \vec{x}) = \lambda\vec{y}.\varphi_i^{(m+n)}(\vec{x}, \vec{y})$$

Si noti come il teorema del parametro e quello di enumerazione siano "l'uno l'inverso dell'altro", in quanto uno "aumenta" il numero di argomenti e l'altro lo diminuisce.

L'importanza dei teoremi di enum e parametro è sintetizzata nel seguente teorema, non dimostrato:

Teorema 1.1.7 (espressività)
Un formalismo è Turing-equivalente ***se e solo se***:

- Vale il teorema di enumerazione (i.e. ha un algoritmo "universale")
- Vale il teorema del parametro

Teorema 1.1.8 (Di Ricorsione, Kleene II)

$$\forall f \text{ calc. totale } \exists (n).\varphi_n = \varphi_{f(n)}$$

Per ogni funzione calcolabile totale esiste un indice che ne è **punto fisso**, ossia tale che $\varphi_n = \varphi_{f(n)}$

. . . .

Intuizione: La funzione f "trasforma" programmi in altri programmi; quando si considera il punto fisso, la trasformazione operata **non cambia la funzione calcolata**, ovvero trasforma un programma in un programma diverso **con la stessa semantica**.

. . . .

Dimostrazione. Definiamo la seguente funzione calcolabile:

$$\psi(u, z) = \begin{cases} \varphi_{\varphi_u(u)}(z) & \text{se } \varphi_u(u) \downarrow \\ \text{indefinita} & \text{altrimenti} \end{cases} \qquad (1)$$

- $\psi(u, z) = \varphi_i(u, z)$ (Church-Turing, calcolabile $\implies \exists$ indice i)
- $\exists g : \varphi_i(u, z) = \varphi_{g(i,u)}(z)$ (Teorema del parametro)
- Definisco: $\lambda x.g(i, x) = d(x)$ (*Barbatrucco*)
- $\implies \varphi_{g(i,u)}(z) = \varphi_{d(u)}(z)$ (Applico il *Barbatrucco*)
- $f(d(x)) = \varphi_v(x)$ (CT, calcolabile $\implies \exists$ indice v)
- $$f, d \text{ totali} \implies f(d(x)) = \varphi_v(x) \text{ totale} \implies \varphi_v(v) \downarrow \qquad (2)$$
- $\implies \varphi_{d(v)}(z) = \psi(v, z) = \varphi_{\varphi_u(u)}(z)$ (1)
- $$n := d(v) \qquad (3)$$

Adesso possiamo dimostrare che n è punto fisso di f:

$$\varphi_n \overset{(3)}{=} \varphi_{d(v)} \overset{(1)}{=} \varphi_{\varphi_v(v)} \overset{(2)}{=} \varphi_{f(d(v))} \overset{(3)}{=} \varphi_{f(n)}$$

□

1.2 Problemi insolubili e riducibilità

Definizione 1.1 (Insieme ricorsivo)
Un insieme I si dice **ricorsivo** se la sua funzione caratteristica è **calcolabile totale**.

Intuizione Esiste un algoritmo che, dato x in input, determina se $x \in I$ in un numero finito di passi.

Nota: talvolta con "funzione ricorsiva" si intende "funzione ricorsiva generale totale", mentre le funzioni ricorsive generali *parziali* vengono dette semplicemente "ricorsive generali".

Definizione 1.2 (Insieme ricorsivamente enumerabile)
Un insieme I si dice *ricorsivamente enumerabile* se e solo se $\exists i.I = dom(\varphi_i)$, ossia sse è dominio di almeno una funzione calcolabile parziale.

Intuizione S è r.e. se $\exists$ algo che, dato in input x, termina se $x \in S$; alternativamente, r.e. se esiste un algoritmo capace di enumerare gli elementi di S

Teorema 1.2.1
I ricorsivo $\implies$ I ricorsivamente numerabile

Dimostrazione. La funzione φ_i di cui I è dominio converge su x se e solo se $\chi_I(x) = 1$, e.g:

$$\varphi_i = \begin{cases} 1 & \chi_I(x) = 1 \\ \text{indef.} & \text{altrim.} \end{cases}$$

□

Teorema 1.2.2
I (e $\bar{I}$) ricorsivamente enumerabili $\iff$ I (e $\bar{I}$) ricorsivi.

....

Dimostrazione. "Se": vd punto precedente. "Solo se": Siano $\varphi_i(x)$ e $\varphi_{\bar{i}}(x)$ le funzioni i cui domini sono risp. I e $\bar{I}$; Allora si esegue il seguente ciclo:

- Esegui un passo nel calcolo di $\varphi_i(x)$; se $\varphi_i(x) \downarrow$ allora $x \in I$, $\chi_I(x) = 1$
- Altrimenti esegui un passo nel calcolo di $\varphi_{\bar{i}}$; se $\varphi_{\bar{i}} \downarrow$ allora $x \notin I$, $\chi_{\bar{I}} = 0$.

□

Teorema 1.2.3
$I \neq \emptyset$ è ricorsivamente enumerabile $\iff$ $\exists f$ calcolabile tale che $I = \text{imm}(f)$

9	13	18	24
5	8	12	17
2	4	7	11
0	1	3	6

Intuizione ($\Rightarrow$) Devo costruire una funzione la cui immagine è I.
Per ogni $m, n \in \mathbb{N}$, sia $\langle m, n\rangle$ la codifica di (m, n), ossia l'elemento in posizione (m, n) della tabella:

- Se $\varphi_i(n)$ termina in m passi, $f(\langle m, n\rangle) = n$
- Se no $f(\langle m, n\rangle) = \bar{n}$ che si ottiene spostandosi ripetutamente all'intero successivo $\langle m, n\rangle + 1$ nella tabella, finché per qualche $\langle m, \bar{n}\rangle$ vale che $\varphi_i(\bar{n})$ termina in m passi.

....

Dimostrazione.

($\Rightarrow$) $I = \text{dom}(\varphi_i) \neq \emptyset$, costruisco $f : I = \text{imm}(f)$

- Si cerca un elemento di I mediante un procedimento a coda di colomba
- Sia n l'argomento di φ_i ed m il numero di passi nel calcolo di $\varphi_i(n)$.
- Chiamiamo $\langle n, m\rangle$ la codifica di (m, n)
- Si calcolano m passi del calcolo di $\varphi_i(n)$:
  * se il calcolo termina, si pone $f(\langle m, n\rangle) = n$;
  * se il calcolo non termina, si prosegue (si considera $\langle m, n\rangle + 1$ muovendosi "a coda di colomba") finché per qualche $m, \bar{n}$ il calcolo non si arresta, e si pone $f(\langle m, n\rangle) = \bar{n}$.

$\bar{n}$ è sicuramente in I, quindi con questo procedimento si generano tutti gli elementi di I.

($\Leftarrow$) (*non riportata da Degano, ma credo sia questo:*) Se I è immagine di una funzione calcolabile totale f, allora è dominio della funzione g costruita considerando, per ogni $y \in I$, **uno** degli $x : f(x) = y$ e ponendo $g(y) = x$.

□

Definizione 1.3
Chiamiamo *rec* l'insieme delle funzioni "ricorsive", allora:

$$I \text{ è } \begin{cases} \text{ricorsiva} & \iff \chi_I \in rec = \{\varphi_x : \text{ dom } (\varphi_x) = \mathbb{N}\} \\ \text{ric. enumerabile} & \iff I = \text{ dom } (\varphi_x) \end{cases}$$

Inoltre chiamiamo:

- L'insieme degli insiemi ricorsivi $\mathcal{R}$
- L'insieme degli insiemi r.e. $\mathcal{RE}$
- L'insieme degli insiemi non r.e. $non\mathcal{RE}$

E vedremo che $\mathcal{R} \subsetneq \mathcal{RE} \subsetneq non\mathcal{RE}$

Se I è ricorsivamente enumerabile, posso determinare l'appartenenza all'insieme I in tempo finito, ma non posso determinare la non appartenenza. Ad esempio:

Proposizione 1.2.1
L'insieme:

$$K = \{i : \varphi_i(i) \downarrow\}$$

È ricorsivamente enumerabile.

. . . .

Dimostrazione. K è dominio di:

$$\psi(n) = \begin{cases} 1 & \text{se } \varphi_n(n) \downarrow \\ indef & \text{o/w} \end{cases}$$

□

Ma **non tutti gli insiemi r.e. sono ricorsivi!** Infatti:

Proposizione 1.2.2
L'insieme:

$$K = \{i : \varphi_i(i) \downarrow\}$$

Non è ricorsivo.

. . . .

Significato intuitivo Non esiste un algoritmo per decidere se $x \in K$ o no. Il problema è *insolubile.*

. . . .

Dimostrazione. Per assurdo: sia k ricorsivo $\implies \chi_K \in rec$, cioè è calcolabile totale. Sia:

$$\psi(n) = \begin{cases} \varphi_n(n) + 1 & \text{se } n \in K(\iff \chi_K(n) = 1 \iff \varphi_n(n) \downarrow) \\ 0 & \text{o/w} \end{cases}$$

Se χ_K è ricorsiva (i.e. calcolabile totale) allora anche $\psi(n)$ lo è (è definita a partire da χ_K), ma se scegliamo un qualsiasi $\bar{n}$ come indice della funzione:

$$\psi(\bar{n}) \neq \varphi_{\bar{n}}(\bar{n})$$

□

. . . .

Corollario $\bar{K}$ non è ricorsivamente enumerabile, poiché per il teorema su I e $\bar{I}$ abbiamo che K non ric $\implies \neg(K \text{ r. } \wedge \bar{K} \text{ r.}) \implies \neg(K \text{ r. e. } \wedge \bar{K} \text{ r. e.})$, ma dato che K è r.e. allora $\bar{K}$ è non r.e. Potremmo dire che $\bar{K} \in$ co-$\mathcal{RE}$, la classe dei problemi i cui complementi sono r.e.

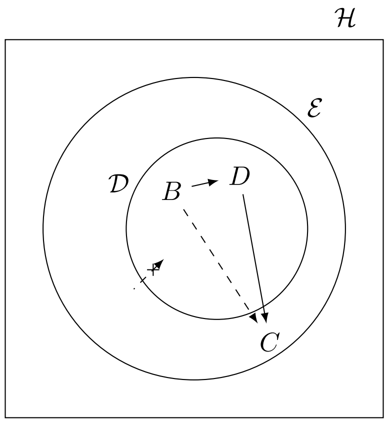
Teorema 1.2.4 (Problema della fermata, in realtà altro **Corollario**)
Sia $K_0 = \{(x, y) : \varphi_y(x) \downarrow\} = \{(x, y) : \exists z.T(y, x, z)\}$, dove T è il pred. di Kleene.
Questo non è ricorsivo.

Dimostrazione. $x \in K \iff (x, x) \in K_0$, quindi se K_0 fosse ricorsivo lo sarebbe anche K. $\square$

Questo è un esempio di **riduzione**.

Definizione 1.4
f è riduzione da A a B se $x \in A \iff f(x) \in B$, e si indica: $a \leq_f B$

Definizione 1.5
Dato un insieme di funzioni $\mathcal{F}$, $A \leq_{\mathcal{F}} B \iff \exists f \in \mathcal{F} : A \leq_f B$



Una relazione di riducibilità è un **preordine**, i.e. una relazione transitiva e riflessiva.
Si dice che $\mathcal{F}$ classifica $\mathcal{D}$ ed $\mathcal{E}$ sse:

1. $A \leq_{\mathcal{F}} A$ (Riflessività)
2. $A \leq B$, $B \leq C \iff A \leq C$ (Transitività)
3. $A \leq B$, $B \in \mathcal{D} \implies A \in \mathcal{D}$ (no "frecce entranti" in $\mathcal{D}$)
4. $A \leq B, B \in \mathcal{E} \implies A \in \mathcal{E}$ (no "frecce entranti" in $\mathcal{E}$)

META-DIGRESSIONE

Le ultime due proprietà derivano dal fatto che $\mathcal{D}$ e $\mathcal{E}$ sono **ideali** del *proset* $(\mathcal{H}, \leq)$, ossia

- $\forall\ B \in \mathcal{D}, \quad A \in \mathcal{H} \quad . \quad A \leq B \implies A \in \mathcal{D}$ (Si dice che $\mathcal{D}$ è **chiuso verso il basso**)

[Se il preordine del *proset* è un ordine parziale, un ideale deve essere anche un **insieme diretto**, i.e. $a, b \in I \implies \exists c : a \leq c, b \leq c$]

Le proprietà 1-4 possono anche essere espresse in modo equivalente:

1. $\mathrm{id} \in \mathcal{F}$
2. $f, g \in \mathcal{F} \implies f \circ g \in \mathcal{F}$
3. $f \in \mathcal{F}$, $B \in \mathcal{D} \implies \{f : f(x) \in B\} \in \mathcal{D}$
4. $f \in \mathcal{F}$, $B \in \mathcal{E} \implies \{f : f(x) \in B\} \in \mathcal{E}$

Supponiamo che $\leq_{\mathcal{F}}$ classifichi $\mathcal{D}$ ed $\mathcal{E}$.

. . . .

Definizione 1.6
H si dice $\leq_{\mathcal{F}}$-arduo per $\mathcal{E}$ se e solo se:

$$\forall A \in \mathcal{E} : A \leq_{\mathcal{F}} H$$

Intuizione: Ogni A è al più *difficile* quanto H. Parallelismo con i problemi NP-ardui, i.e. i problemi "più difficili" della classe NP.

Difficile? Perché un problema A si possa ridurre ad un problema B, B deve essere difficile almeno quanto A. Possiamo quindi interpretare $A \leq B$ come A difficile al più quanto B.

. . . .

Definizione 1.7
C si dice $\leq_{\mathcal{F}}$-completo per $\mathcal{E}$ se e solo se:

$$C \text{ è } \leq_{\mathcal{F}} \text{-arduo per } \mathcal{E}, \quad H \in \mathcal{E}$$

Intuizione: Se C è difficile quanto o più di ogni altro elemento di $\mathcal{E}$ ed è in $\mathcal{E}$, vuol dire che **il più difficile** problema[1] che è interno ad $\mathcal{E}$.

Da ciò si può inoltre intuire che due problemi completi rispetto ad $\mathcal{E}$ saranno equivalenti, i.e. l'uno si può ridurre nell'altro.

. . . .

Definizione 1.8
Con **gradi** di una relazione di riduzione si intendono le classi di equivalenza rispetto alla relazione $\equiv$, dove:

$$A \equiv B \iff A \leq B, B \leq A$$

. . . .

Proposizione 1.2.3
Se $\leq$ classifica $\mathcal{D}$ ed $\mathcal{E}$, $\mathcal{D} \subseteq \mathcal{E}$, C completo per $\mathcal{E}$, allora

$$C \in \mathcal{D} \iff \mathcal{D} = \mathcal{E}$$

. . . .

Intuizione S(s)e un problema NP-completo si trovasse nella classe P, si avrebbe che $P = NP$

. . . .

Dimostrazione.

- Se: ovvia
- Solo se: $C \in \mathcal{D}$, $A \in \mathcal{E}$. Per completezza $A \leq C$, e per chiusura verso il basso $A \in D$; di conseguenza $\mathcal{E} \subseteq \mathcal{D}$, e dato che $\mathcal{D} \subseteq \mathcal{E}$ si ha la tesi.

$\square$

[1]si parla di problema in senso di problema di decisione di appartenenza all'insieme

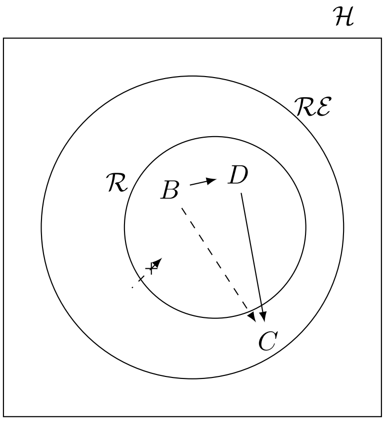
Proposizione 1.2.4
C $\leq$-completo per $\mathcal{E}$, $C \leq E$ allora anche E è $\leq$-completo per $\mathcal{E}$

Dimostrazione. $\forall D \in \mathcal{E}, D \leq A$ per completezza, ma $\leq$ classifica $\mathcal{D}$ ed $\mathcal{E}$, quindi

$$D \leq A, \quad A \leq B \implies D \leq B$$

quindi B è arduo. Ma $B \in \mathcal{E}$, quindi è anche completo. □

Proposizione 1.2.5 □
L'insieme di funzioni $rec = \{\varphi_x : \text{dom}(\varphi_x) = \mathbb{N}\}$ (delle funzioni calcolabili totali) **classifica** $\mathcal{R}$ ed $\mathcal{RE}$.



Dimostrazione. $\mathcal{R} \subseteq \mathcal{RE}$, ok

1. id $\in$ rec, ok
2. $f, g \in$ rec $\implies f \circ g \in rec$
3. Siano $f \in rec$, $B \in \mathcal{R}$.
   La funzione caratteristica di $\{x : f(x) \in B\}$ è $\chi_B \circ f$, calcolabile totale ($\in rec$) perché composizione di funzioni calc tot.
4. Siano $f \in rec$, $B \in \mathcal{RE}$.
   La funzione *semicaratteristica* di $A = \{x : f(x) \in B\}$ è calcolabile, quindi A è dominio di una funzione calcolabile $\implies \in \mathcal{RE}$.

Teorema 1.2.5
K è $\mathcal{RE}$-completo

Dimostrazione. Sappiamo che $K \in \mathcal{RE}$. Dobbiamo mostrare che $A \in \mathcal{RE} \implies A \leq K$
$A \in \mathcal{RE} \implies A$ è il dominio di una qualche funzione calcolabile ψ.
Costruiamo adesso $\psi' = \lambda x, y.\psi(x)$; adesso usiamo *il solito barbatrucco*:

$$\psi'(x, y) = \varphi_i(x, y) = \varphi_{s(i,x)}(y)$$

Allora $A = \{x \mid \varphi_{s(i,x)}(y) \downarrow\}$, e ponendo $y = s(i, x)$ si ottiene:

$$A = \{x \mid \varphi_{s(i,x)}(s(i, x)) \downarrow\} = \{x \mid s(i, x) \in K\}$$

Quindi, sia $f = \lambda x.s(i, x)$:
$x \in A \iff f(x) \in K$, ed f è calcolabile totale per il teorema del parametro (barbatrucco) □

Teorema 1.2.6 (Lemma)
Sia A un iirf tc $\emptyset \neq A \neq \mathbb{N}$, allora:

$$K \leq A \text{ oppure } K \leq \bar{A}$$

Dimostrazione. Sia $i_0 : \varphi_{i_0} = \lambda y.\text{ind}$

- $i_0 \in \bar{A}$: Poiché $A \neq \emptyset$ esiste un $i_1 \in A$, e si ha $\varphi_{i_1} \neq \varphi_{i_0}$ poiché A è iirf. Definiamo ora:

  $$\psi_(x, y) = ... = \varphi_{f(x)}(y) = \begin{cases} \varphi_{i_1}(y) & x \in K \\ \text{ind} = \varphi_{i_0}(y) & \text{altrimenti} \end{cases}$$

  - $x \in K \implies \varphi_{f(x)} = \varphi_{i_1}$, e dato che $i_1 \in A$ (A iirf) vale $f(x) \in A$
  - $x \notin K \implies \varphi_{f(x)} = \varphi_{i_0}$, e dato che $i_0 \notin A$ (A iirf) vale $f(x) \notin A$

  Quindi si ha che $K \leq A$.

- $i_0 \in A$: Poiché $A \neq \mathbb{N}$ esiste $i_1 \notin A$; stessa ψ:

  - $x \in K \implies \varphi_{f(x)} = \varphi_{i_1}$, e dato che $i_1 \notin A$ (A iirf) vale $f(x) \notin A$
  - $x \notin K \implies \varphi_{f(x)} = \varphi_{i_0}$, e dato che $i_0 \in A$ (A iirf) vale $f(x) \in A$

  Quindi si ha che $K \leq \bar{A}$.

□

Teorema 1.2.7 (Rice)
Sia $\mathcal{F}$ una classe di funzioni calcolabili, e $I = \{n | \varphi_n \in \mathcal{F}\}$ (iirf). Questo insieme è ricorsivo se e solo se $\mathcal{A} = \emptyset$ oppure è la classe di tutte le funzioni calcolabili.

Dimostrazione. I è iirf, quindi se diverso da $\emptyset$ e $\mathbb{N}$ non può essere ricorsivo, poiché per il lemma valgono una tra:

- $K \leq A$, K non ricorsivo $\implies$ A non ricorsivo
- $K \leq \bar{A} \implies \bar{K} \leq A$, $\bar{K}$ non ricorsivo $\implies$ A non ricorsivo.

$\emptyset$ e $\mathbb{N}$ sono banalmente ricorsivi.

□