

Introduciamo adesso il seguente predicato:

$$\text{STP}^{(n)}(x_1, \dots, x_n, y, t) \Leftrightarrow$$

Il programma numero y si ferma sulle variabili di ingresso x_1, \dots, x_n dopo al più t passi.

Il teorema del programma universale ci permette di dimostrare immediatamente che

I PREDICATI $\text{STP}^{(n)}(x_1, x_2, \dots, x_n, y, t)$
SONO CALCOLABILI.

Basta infatti fare una modifica al programma universale che conti i "passi" di calcolo compiuti dal programma e ci informi, in corrispondenza ad essi, se il programma si è fermato o meno.

Z ← X_{n+1} + 1

S ← $\prod_{i=1}^n (p_{2i}) X_i$

K ← 1
[C] Q ← Q + 1 (*)
IF Q > X_{n+2} + 1 GOTO E (*)
IF K - Lt(Z) + 1 v K - 0 GOTO F
U ← r((Z)_K)
P ← Pr(U)+1
IF I(U) = 0 GOTO N
IF I(U) = 1 GOTO A
IF v(P | S) GOTO N
IF I(U) = 2 GOTO M
K ← min i ≤ Lt(Z) [I((Z)_i) + 2 = I(U)]
GOTO C
[M] S ← [S/P]
GOTO N
[A] S ← S · P
[N] K ← K + 1
GOTO C
[F] Y ← 1 (*)

Programma che calcola

Y = STP(n) (X₁, ..., X_n, X_{n+1}, X_{n+2}).

INSIEMI RICORSIVAMENTE ENUMERABILI

Come già sappiamo, dire che un insieme B appartiene ad una qualche classe di funzioni è equivalente a dire che il predicato

$$P(x) \leftrightarrow x \in B$$

appartiene alla classe di funzioni detta - la relazione tra B e $P(x)$ è data da

$$B = \{x \in \mathbb{N} \mid P(x)\}$$

Quindi dire che:

- l'insieme B è calcolabile (o ricorsivo) è equivalente a dire che $P(x)$ è una funzione calcolabile.
- l'insieme B è ricorsivo primitivo equivale a dire che $P(x)$ è ricorsivo primitivo.

Diamo adesso la seguente

DEFINIZIONE

Un insieme $B \subseteq \mathbb{N}$ si dice ricorsivamente enumerabile se esiste una funzione parzialmente calcolabile $g(x)$ tale che

$$B = \{x \in \mathbb{N} \mid g(x) \downarrow\}$$

("↓" significa "è definita")

Se P è un programma che calcola la funzione g , allora possiamo dire che B è semplicemente l'unione di tutti i valori di ingresso di P per i quali P prima o poi si ferma.

Esaminiamo adesso tale situazione dal punto di vista che segue:

Noi abbiamo a disposizione il programma P . Di che aiuto è P per verificare se un certo elemento appartiene o no a B ?

La risposta è che P ci aiuta solo parzialmente perché mentre per tutti gli elementi che appartengono a B il programma, prima o poi, si fermerà, negli elementi che non appartengono a B il programma continuerà a girare per sempre (SENZA CHE NOI ABBIAMO MODO DI SAPERLO)

Quindi fino a quando P non si è fermato noi non sapremo mai se ha ancora bisogno di tempo per terminare il calcolo oppure se è entrato in un ciclo infinito e non si fermerà mai.

Abbiamo adesso le due nozioni di

- insieme ricorsivo e
- insieme ricorsivamente enumerabile.

LE RELAZIONI CHE INTERCORRONO TRA LORO SONO STABILITE DALLE DUE PROPOSIZIONI CHE SEGUONO

PROPOSIZIONE

SE L'INSIEME B È RICORSIVO ALLORA
È RICORSIVAMENTE ENUMERABILE

Dimostrazione

Costruiamo il seguente programma P :

[A] IF $\neg(x \in B)$ GOTO A

La prima domanda da porsi è se P è un programma di \mathcal{L} , ossia se esiste una macroespansione di P in termini delle istruzioni base di \mathcal{L} .

La risposta è sì perché, essendo B ricorsivo per ipotesi, allora $x \in B$ è un predicato calcolabile.

Sia h la funzione calcolata da P .

In base alla costruzione del programma h sarà definita solo sugli elementi di B .

Avremo allora che:

$$B = \{x \in \mathbb{N} \mid h(x) \downarrow\} \quad (*)$$

e la proposizione è dimostrata perché la (*) è proprio la definizione di insieme r.e., essendo h parzialmente calcolabile.

Proposizione (Post)

L'INSIEME B È RICORSIVO SE E SOLO SE
SIA B CHE \overline{B} SONO R.E. (RICORSIVAMENTE
ENUMERABILI).

Dimostrazione

- A) Sia B ricorsivo, sappiamo già che allora anche \overline{B} è ricorsivo; applicando quindi la proposizione precedente sia a B che a \overline{B} otteniamo che sia \overline{B} che B sono r.e.
- B) Ammettiamo adesso che sia B che \overline{B} siano r.e.; possiamo allora scrivere, in base alle definizioni, che

$$B = \{x \in \mathbb{N} \mid g(x) \downarrow\}$$

$$\overline{B} = \{x \in \mathbb{N} \mid h(x) \downarrow\}$$

con g ed h funzioni parzialmente calcolabili.

Siano: g calcolata dal programma P
 h calcolata dal programma Q

e siano ancora $p = \#(P)$ e $q = \#(Q)$.

Il programma che segue calcola la funzione caratteristica di B e quindi B è ricorsivo:

```
[A] IF STP(1)(X, p, T) GOTO C
     IF STP(1)(X, q, T) GOTO E
     T ← T+1
     GOTO A
[C] Y ← 1
```

Osservazioni.

1. Come fa il programma?

Usa il predicato contaparsi che è calcolabile per controllare, ricorsivamente, se sulla variabile di ingresso X si ferma il programma P , se sì allora viene assegnato ad Y il valore 1; in caso contrario esegue lo stesso controllo su Q . Se il programma Q si ferma allora si ferma anche il nostro programma ma questa volta senza modificare il valore di Y che quindi rimane 0.

Il processo viene ripetuto per tutti i valori di T .

Esiste in B che \bar{B} sono per ipotesi r. e. il programma globale dovrà fermarsi in qualsiasi ingresso.

2. Tale risultato è intuitivamente più che naturale.

Il predicato contaparsi ci permette però di tradurlo immediatamente in linguaggio formale appropriato.

Il teorema di Post, che abbiamo appena dimostrato, ci dice in che relazione stanno gli insiemi ricorsivi e gli insiemi ricorsivamente enumerabili in base alla loro definizione. Potrebbe però succedere che le due nozioni coincidano.

Osserviamo che dal Teorema di Post discende una conseguenza interessante. Poichè se il complemento di un insieme ricorsivamente enumerabile A è anch'esso ricorsivamente enumerabile, A è ricorsivo, perchè le due definizioni siano realmente distinte devono esistere necessariamente insiemi che non solo non sono ricorsivi, ma non sono neanche ricorsivamente enumerabili.

Quindi, per mostrare che esse sono realmente distinte dobbiamo portare un esempio di insieme ricorsivamente enumerabile ma non ricorsivo. E' quello che faremo immediatamente dopo introducendo e studiando un particolare insieme che chiameremo K .

Una conseguenza di questo fatto, per quanto affermato prima, è che devono esistere anche insiemi che non sono neanche ricorsivamente enumerabili.

Mostreremo, subito dopo, che l'insieme TOT dei numeri di programmi che calcolano funzioni ovunque definite, è proprio un esempio di insieme che non solo non è ricorsivo ma non è neanche ricorsivamente enumerabile.

Ritorniamo un attimo indietro al programma universale.

Per ciascun $n > 0$ abbiamo che la funzione

$$\Phi^{(n)}(x_1, \dots, x_n, y) = \Psi_{\mathcal{P}}^{(n)}(x_1, \dots, x_n) \text{ con } \#(\mathcal{P}) = y$$

è parzialmente calcolabile.

Fissiamo adesso n ,

avremo che le successioni

$$\Phi^{(n)}(x_1, \dots, x_n, 0), \Phi^{(n)}(x_1, \dots, x_n, 1), \dots$$

che può anche scriversi come

$$\Phi_0^{(n)}(x_1, \dots, x_n), \Phi_1^{(n)}(x_1, \dots, x_n), \dots$$

enumera tutte le funzioni parzialmente calcolabili di n variabili.

Limitandoci al caso di una variabile possiamo allora scrivere

$$\Phi_0(x), \Phi_1(x), \dots$$

Giudichiamo adesso:

$$W_n = \{x \in \mathbb{N} \mid \Phi_n(x) \downarrow\}$$

abbiamo ovviamente che un insieme è ricorsivamente enumerabile se e solo se esiste un n per cui $B = W_n$.

Definiamo adesso

$$K = \{n \in \mathbb{N} \mid n \in W_n\}$$

Abbiamo che:

TEOREMA

K È R.E. MA NON È RICORSIVO.

Prima di passare alle dimostrazioni
esplicitiamo il simbolismo.

$$W_n = \{x \in \mathbb{N} \mid \Phi_n(x) \downarrow\}$$

quindi:

$$n \in W_n \iff \Phi_n(n) \downarrow \iff \text{HALT}(n, n)$$

Allora:

K è l'insieme di tutti i numeri n
tali che il programma numero n
si ferma, prima o poi, sull'input n .

Passiamo adesso alla dimostrazione

Teorema. K è r.e. ma non è ricorsivo.

Dimostrazione.

$$K = \{n \in \mathbb{N} \mid n \in W_n\} = \{n \in \mathbb{N} \mid \Phi_n(n) \downarrow\}$$

Poiché $\Phi_n(n)$ è parzialmente calcolabile ne segue che K è r.e.

Assumiamo adesso che lo sia anche \bar{K}

Allora \bar{K} dovrebbe comparire nella numerazione W_i , cioè dovrebbe esistere un indice i tale che

$$\bar{K} = W_i \quad (*)$$

Poniamoci adesso la domanda:

$i \in K$?

Abbiamo che:

$$i \in K \iff i \in W_i$$

ma, in base alla (*), $i \in W_i \iff i \in \bar{K}$
e quindi avremmo che:

$$i \in K \iff i \in \bar{K}$$

— Contraddizione.

\bar{K} non può essere r.e. e quindi K non è ricorsivo.

K può scriversi come:

$$\begin{aligned} K &= \{n \in \mathbb{N} \mid n \in W_n\} \\ &= \{n \in \mathbb{N} \mid \Phi_n(n) \downarrow\} \\ &= \{n \in \mathbb{N} \mid \text{HALT}(n, n)\} \end{aligned}$$

Dire che K è ricorsivo equivale a dire che il predicato di cui esso è l'estensione (HALT) è ricorsivo e noi sappiamo già che HALT non è un predicato calcolabile e quindi ne segue immediatamente che l'insieme K non può essere ricorsivo.

Porteremo adesso l'esempio di un insieme che non solo non è *ricorsivo* ma non è neanche *ricorsivamente enumerabile*.

Questo insieme, che verrà chiamato TOT, è l'insieme di tutti i numeri di codice dei programmi che calcolano *funzioni totali*.

CARATTERIZZAZIONE DELLE FUNZIONI TOTALI

Sia TOT l'insieme di tutti i numeri p tali che p è il numero di un programma che calcola una funzione totale $f(x)$ di una variabile:

$$\text{TOT} = \{z \in \mathbb{N} \mid \forall x (\Phi(x, z) \downarrow)\}$$

Ricordiamo di avere già definito gli insiemi W_z come:

$$W_z = \{x \in \mathbb{N} \mid \Phi(x, z) \downarrow\}$$

$$(\text{cioè } x \in W_z \iff \Phi(x, z) \downarrow)$$

Allora TOT è l'insieme dei numeri z tali che W_z coincide con \mathbb{N} .

Ricordiamo che (come dimostreremo in seguito) gli enunciati seguenti sono equivalenti:

UN INSIEME S È RICORSIVAMENTE ENUMERABILE

S È IL CODOMINIO DI UNA FUNZIONE RICORSIVA PRIMITIVA

S È IL CODOMINIO DI UNA FUNZIONE RICORSIVA (TOTALE)

S È IL CODOMINIO DI UNA FUNZIONE RICORSIVA PARZIALE

TEOREMA

TOT non è ricorsivamente enumerabile.

Dimostrazione

Supponiamo, per assurdo, che sia r.e.

Poiché $TOT \neq \emptyset$, esisterà una funzione calcolabile $g(n)$ tale che:

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

Sia $h(n) = \Phi(n, g(n)) + 1$

Poiché ciascun $g(n)$ è il numero di un programma che calcola una funzione totale, $\Phi(n, g(n))$ deve essere definita per ogni n , e quindi deve essere anche $h(n) \downarrow$ per ogni n .

Quindi anche $h(n)$ è calcolabile (e totale);

Sia calcolata dal programma P e sia

$$p = \#(P).$$

Allora deve essere $p \in TOT$ e, poiché g enumera tutti gli elementi di TOT , sarà $p = g(i)$ per qualche i .

Allora avremo che:

$$h(i) = \Phi(i, g(i)) + 1 = \Phi(i, p) + 1 = h(i) + 1$$

p numero del programma che calcola h , per cui $\Phi(i, p) = h(i)$

CONTRADDIZIONE!

Non possiamo assumere TOT r.e.!