TEOREMA DEL PARAMETRO

Consideriamo una funzione di k variabili

$$\Phi^{(k)}(x_1, \dots, x_k, y)$$

(y non è una variabile ordinaria ma il numero di codice del programma)

Esistono delle situazioni nelle quali noi siamo esplicitamente interessati alla dipendenza funzionale da una sola variabile o da un sottogruppo delle k variabili.

In questo caso ci piacerebbe presentare la $\Phi^{(k)}$ come funzione, ad es., di una sola variabile calcolata da un programma opportuno.

Questo programma dovrebbe essere ottenibile, in modo calcolabile, a partire dal programma che calcola $\Phi^{(k)}$ e dalle variabili che non si vogliono considerare.

Il cosiddetto teorema del parametro, o teorema di iterazione o teorema s-m-n, afferma non solo che il desiderio espresso prima è realizzabile ma anche che la funzione che ci fornisce il numero del nuovo programma è ricorsiva primitiva.

TEOREMA

Per ciascun $n, m > 0$ esiste una funzione ricorsiva primitiva $S_m^n(u_1, u_2, \dots, u_n, y)$ tale che

$$\Phi^{(m+n)}(x_1, \dots, x_m, u_1, \dots, u_n, \underbrace{y}_{\text{codice del programma}}) = \Phi^{(m)}(x_1, \dots, x_m, \underbrace{S_m^n(u_1, \dots, u_n, y)}_{\text{codice del programma}}).$$

Dimostrazione.

La dimostrazione è per induzione su n.

Per $n=1$ dobbiamo mostrare che esiste una funzione ricorsiva primitiva $S_m^1(u,y)$ tale che:

$$\Phi^{(m+1)}(x_1,\dots,x_m,u,y) = \Phi^{(m)}(x_1,\dots,x_m, S_m^1(u,y))$$

Ricordando la notazione per le Φ, risulta chiaro che né y né $S_m^1(u,y)$ sono variabili proprie, di ingresso, rispettivamente di $\Phi^{(m+1)}$ e $\Phi^{(m)}$, ma sono i numeri dei programmi che calcolano $\Phi^{(m+1)}$ e $\Phi^{(m)}$.

In particolare, se è $y=\#(\mathcal{P})$ allora possiamo interpretare $S_m^1(u,y)$ come il numero di un programma che per prima cosa assegna il valore u alla variabile X_{m+1} e, successivamente, esegue il programma $\mathcal{P}$.

Un programma che assegna il valore u ad X_{m+1} è il seguente:

$$\left.\begin{array}{l} X_{m+1} \leftarrow X_{m+1}+1 \\ \vdots \\ X_{m+1} \leftarrow X_{m+1}+1 \end{array}\right\} u \text{ volte}$$

N.B.
Nel caso di $\Phi^{(m)}$, X_{m+1} non è una variabile di ingresso in senso stretto; il suo valore di partenza è 0.

Dobbiamo adesso calcolare il numero dell'istruzione non etichettata $X_{m+1} \leftarrow X_{m+1}+1$.

Abbiamo :

$\#(I) = \langle a, \langle b, c \rangle\rangle = \langle 0, \langle 1, 2m+1 \rangle\rangle$

Infatti :

$a = 0$ perché non c'è etichetta.

$b = 1$ perché l'istruzione è di "incremento".

$$c = \#(V) - 1 = \#(X_{m+1}) - 1 = 2(m+1) - 1 = 2m+1$$

($\#(X_{m+1}) = 2(m+1)$ perché la X_{m+1} è la $(m+1)$esima variabile di ingresso e tali variabili occupano sempre posto pari nell'ordinamento dato).

Calcoliamo adesso il valore di $\#(I)$:

Ricordiamo che $\langle x, y \rangle = 2^x(2y+1) - 1$

Si ha che :

$$\langle b, c \rangle = \langle 1, 2m+1 \rangle = 2\big(2(2m+1)+1\big) - 1 = 8m+5$$

e sostituendo :

$$\langle a, \langle b, c \rangle\rangle = \langle 0, 8m+5 \rangle = 2^0\big(2(8m+5)+1\big) - 1 = \\ = 16m + 10.$$

Il numero di codice dell'istruzione non etichettata $X_{m+1} \leftarrow X_{m+1} + 1$ è dunque $\underline{16m + 10}$

$S^1_m(u, y)$ è il numero del nuovo programma mentre y è il numero del programma originario $\mathcal{P}$.

Allora abbiamo che

$$y = \#(\mathcal{P}) = [\#(I_1), \dots, \#(I_q)] - 1 .$$

Vogliamo esprimere y esplicitamente come numero di Gödel, come prodotto di numeri primi elevati ad esponenti opportuni.

Al momento conosciamo y ma non conosciamo esplicitamente né q né tali esponenti;

- q lo ricaviamo mediante l'operatore lunghezza Lt:

$$q = Lt([\#(I_1), \dots, \#(I_q)]) = Lt(y+1)$$

- gli esponenti da dare ai primi $Lt(y+1)$ numeri primi, che sono quelli che compaiono nella scrittura esplicita del numero di Gödel di $\mathcal{P}$, si possono calcolare mediante la funzione $([\ ])_i$

  L'esponente da dare a p_j è:

$$([\#(I_1), \dots, \#(I_q)])_j = (y+1)_j$$

Avremo quindi che:

$$y = \#(\mathcal{P}) = \prod_{j=1}^{Lt(y+1)} p_j^{(y+1)_j} - 1$$

utilizziamo adesso questi dati per calcolare $S_m^1(u, y)$

Come si è già affermato, $S_m^1(u, y)$ è il numero di codice del programma che si ottiene premettendo le u istruzioni:

$$\left.\begin{array}{l} X_{m+1} \leftarrow X_{m+1} + 1 \\ \vdots \\ X_{m+1} \leftarrow X_{m+1} + 1 \end{array}\right. \qquad (*)$$

al programma $\mathcal{P}$.

Ciò che dobbiamo fare per trovare tale numero di codice è quindi traslare di u unità i numeri primi p_j che compaiono in $\mathcal{P}$ e calcolare il codice delle u istruzioni (*)

Il codice delle (*) è dato da:

$$\prod_{i=1}^{u} p_i^{16m+10}$$

(per trovare il codice del "programma" * dobbiamo sottrarne 1)

e quindi si ha che:

$$S_m^1(u, y) = \left[\left(\prod_{i=1}^{u} p_i\right)^{16m+10} \cdot \prod_{j=1}^{Lt(y+1)} p_{u+j}^{(y+1)_j}\right] \dot{-} 1$$

È evidente che S_m^1 è una funzione ricorsiva primitiva.

Completiamo adesso la dimostrazione.

Supponiamo il risultato valido per $n=k$ e vogliamo mostrare che lo è anche per $k+1$.

Si ha che:

$$\Phi^{(m+k+1)}(x_1,\dots,x_m,u_1,\dots,u_k,u_{k+1},y)=$$

$$=\Phi^{(m+k)}(x_1,\dots,x_m,u_1,\dots,u_k,S^1_{m+k}(u_{k+1},y))$$

$$=\Phi^{(m)}(x_1,\dots,x_m,S^k_m(u_1,\dots,u_k,S^1_{m+k}(u_{k+1},y)))$$

Se adesso poniamo

$$S^{k+1}_m(u_1,\dots,u_k,u_{k+1},y)=S^k_m(u_1,\dots,u_k,S^1_{m+k}(u_{k+1},y))$$

il risultato risulta dimostrato.

Consideriamo adesso un'applicazione immediata del teorema del parametro.

Siano $\Phi_u(x)$ e $\Phi_v(x)$ due funzioni di una variabile calcolate dai programmi numero u e v rispettivamente.

Consideriamo adesso la composizione delle due funzioni

$$\Phi_u(\Phi_v(x))$$

che sicuramente sarà anch'essa calcolabile perché la composizione preserva la calcolabilità.

Esisterà quindi un programma che calcola $\Phi_u(\Phi_v(x))$. Vogliamo trovare il numero di tale programma come funzione di u e v; in altre parole vogliamo trovare $g(u,v)$ nell'equazione:

$$\Phi_u(\Phi_v(x)) = \Phi_{g(u,v)}(x)$$

Abbiamo che:

$$\Phi_u(\Phi_v(x)) = \Phi(\Phi(x,v),u) = \Phi^{(3)}(x,u,v;z_0)$$

per qualche z_0 (numero di programma).

Applicando il teorema del parametro otteniamo:

$$\Phi^{(3)}(x,u,v;z_0) = \Phi(x, S_1^2(u,v;z_0)) = \Phi_{S_1^2(u,v;z_0)}(x)$$

attraverso $S_1^2(u,v;z_0)$ possiamo quindi ricavare la dipendenza di z_0 da u e v.

TEOREMA DI RICORSIONE

Sia $g(z, x_1, \dots, x_m)$ una funzione parzialmente calcolabile di $m+1$ variabili. Allora esiste un numero e tale che:

$$\Phi_e^{(m)}(x_1, \dots, x_m) = g(e, x_1, \dots, x_m)$$

Parafrasiamo innanzitutto l'enunciato del teorema:

Data una funzione g parzialmente calcolabile un valore di una delle variabili è il numero di Gödel (il codice) di un programma che calcola g come funzione delle rimanenti variabili.

Dimostrazione

Consideriamo la funzione parzialmente calcolabile $g(S_m^1(v,v), x_1, \dots, x_m)$.

Esisterà allora un programma z_0 tale che:

$$g(S_m^1(v,v), x_1, \dots, x_m) = \Phi^{(m+1)}(x_1, \dots, x_m, v, z_0) =$$

(applicando il teorema del parametro)

$$= \Phi^{(m)}(x_1, \dots, x_m, S_m^1(v, z_0)).$$

Poniamo adesso $v = z_0$, si ha che:

$$g(S_m^1(z_0, z_0), x_1, \dots, x_m) = \Phi^{(m)}(x_1, \dots, x_m, S_m^1(z_0, z_0))$$

ma $S_m^1(z_0, z_0)$ è un numero, sia e; abbiamo:

$$g(e, x_1, \dots, x_m) = \Phi^{(m)}(x_1, \dots, x_m, e)$$
$$= \Phi_e^{(m)}(x_1, \dots, x_m)$$

Proposizione (esistenza del programma che si "auto-riproduce")

Esiste un numero e tale che, per ogni x:

$$\Phi_e(x) = e.$$

Dimostrazione

Consideriamo la funzione calcolabile

$$g(z, x) = U_1^2(z, x) = z$$

applichiamo a g il teorema di ricorsione, si ha che esiste un numero e tale che

$$\Phi_e(x) = g(e, x)$$

ma $g(e, x) = e$, quindi si ottiene

$$\Phi_e(x) = e$$

Osservazione

Il programma numero e fa qualcosa di più che calcolare una funzione costante. Qualunque sia la variabile d'ingresso, il programma numero e usa questa informazione per ricalcolare sé stesso, cioè il suo numero di codice.

Teorema (del punto fisso)

Sia $f(x)$ una funzione calcolabile. Allora esiste un numero e tale che

$$\Phi_{f(e)}(x) = \Phi_e(x)$$

Cioè per qualsiasi funzione calcolabile esiste un numero e tale che sia e che f(e) sono numeri di programmi che calcolano la stessa funzione Φ.

Dimostrazione.

È una applicazione immediata del teorema di ricorsione.

Poniamo $\Phi_{f(z)}(x) = g(z, x)$

applichiamo il teorema di ricorsione a g; si ha che esiste un e tale che

$$g(e, x) = \Phi_e(x)$$

ma $g(e, x) = \Phi_{f(e)}(x)$ per cui

$$\Phi_e(x) = \Phi_{f(e)}(x)$$

Sia T' un insieme di funzioni parzialmente calcolabili di una sola variabile.

Associamo a T' l'insieme di indici $R_{T'}$ così definito:

$$R_{T'} = \{ t \in N \mid \Phi_t \in T' \}$$

Si ha che:

TEOREMA (DI RICE)

Sia T' un insieme di funzioni parzialmente calcolabili di una sola variabile.
Siano $f(x)$ e $g(x)$ due funzioni parzialmente calcolabili tali che $f(x)$ appartiene a T' e $g(x)$ non vi appartiene.
Allora $R_{T'}$ non è ricorsivo.

Dimostrazione

Supponiamo che $R_{T'}$ sia calcolabile.
Consideriamo la funzione caratteristica di $R_{T'}$:

$$P_{T'}(t) = \begin{cases} 1 & \text{se } t \in R_{T'} \\ 0 & \text{altrimenti} \end{cases}$$

introduciamo ora la funzione h così definita:

$$h(t, x) = \begin{cases} g(x) & \text{se } t \in R_{T'} \\ f(x) & \text{altrimenti.} \end{cases}$$

Ci serve mostrare che h è parzialmente calcolabile. Basta osservare che può esprimersi come:

$$h(t,x) = g(x) \cdot P_{T'}(t) + f(x) \cdot \alpha(P_{T'}(t))$$

Dato che è ~~calcolabile~~ parzialmente calcolabile possiamo applicare ad h il teorema di ricorsione.

ip.: $f(x) \in T'$, $g(x) \notin T'$

Abbiamo che:

$$\Phi_e(x) = h(e,x)$$

Riscriviamo questo risultato:

$$\Phi_e(x) = h(e,x) = \begin{cases} g(x) & \text{se } e \in R_{T'} \Leftrightarrow \Phi_e(x) \in T' \\ f(x) & \text{altrimenti.} \end{cases}$$

vogliamo adesso stabilire se $\Phi_e(x)$ è uguale ad $f(x)$ oppure a $g(x)$.

Supponiamo che sia $\Phi_e(x) \in T'$

allora avremmo $\Phi_e(x) = g(x)$ e quindi $g(x) \in T'$

ma, per ipotesi, $g(x) \notin T'$.

Contraddizione.

Supponiamo allora che sia $\Phi_e(x) \notin T'$

allora avremmo $\Phi_e(x) = f(x)$ e quindi che $f(x) \notin T'$.

Ma, per ipotesi, $f(x) \in T'$.

Contraddizione.

Non possiamo quindi assumere che $R_{T'}$ è ricorsivo, ed il teorema è dimostrato.