

Il linguaggio di diagonalizzazione

1 Enumerazioni

Un'enumerazione di un insieme \mathcal{I} è una *funzione suriettiva* dai numeri naturali all'insieme \mathcal{I} , cioè una funzione $\tau : \mathbb{N} \rightarrow \mathcal{I}$ che soddisfa la proprietà

$$\forall e \in \mathcal{I} \exists k \in \mathbb{N} \text{ tale che } \tau(k) = e$$

In pratica, un'enumerazione indicizza gli elementi dell'insieme \mathcal{I} : si può dire che $e_i = \tau(i) \in \mathcal{I}$ è l' i -esimo elemento dell'insieme. Allora, l'insieme \mathcal{I} può essere scritto come

$$\mathcal{I} = \left\{ \underbrace{e_0}_{\tau(0)}, \underbrace{e_1}_{\tau(1)}, \underbrace{e_2}_{\tau(2)}, \dots, \underbrace{e_i}_{\tau(i)}, \dots \right\}$$

e l'enumerazione τ può essere indicata elencando gli elementi in ordine crescente di indice:

$$\tau : e_0, e_1, e_2, \dots, e_i, \dots$$

In seguito, si considereranno enumerazioni *calcolabili* e *invertibili*, cioè che consentano di determinare effettivamente l'elemento dell'insieme a partire dal suo indice, e viceversa.

1.1 Enumerazione delle stringhe su $\{0, 1\}$

Una possibile¹ enumerazione delle stringhe su $\{0, 1\}$ può essere costruita a partire dalla funzione

$$\begin{aligned} f &: \{0, 1\}^* \rightarrow \mathbb{N} \\ f(w) &= \text{bin2dec}(1w) \end{aligned}$$

dove `bin2dec` è l'usuale funzione di conversione da binario a decimale. Alcuni esempi di applicazione della funzione f sono:

¹Per uno stesso insieme si possono definire diverse enumerazioni. I risultati che verranno dimostrati in seguito sono indipendenti dalle specifiche enumerazioni scelte, a patto che tali enumerazioni siano appunto calcolabili e invertibili.

w	$1w$	$f(w)$
ϵ	1	1
0	10	2
1	11	3
00	100	4
01	101	5
10	110	6
...

Essendo intuitivamente calcolabile, per la tesi di Church-Turing f è calcolabile da una MdT.

Si definisce poi la funzione inversa sinistra di f , cioè una funzione g tale che $g(f(w)) = w$:

$$g : \mathbb{N} \rightarrow \{0, 1\}^*$$

$$g(i) = \begin{cases} \epsilon & \text{se } i = 0 \\ \mathbf{tail}(\mathbf{dec2bin}(i)) & \text{se } i > 0 \end{cases}$$

dove $\mathbf{dec2bin}$ è la conversione da decimale a binario, mentre \mathbf{tail} toglie il primo carattere della stringa (restituendo la “coda” rimanente), al fine di invertire l’aggiunta del simbolo 1 effettuata dalla funzione f . Alcuni esempi di applicazione di g sono:

i	$\mathbf{dec2bin}(i)$	$g(i)$
0	0	ϵ
1	1	ϵ
2	10	0
3	11	1
4	100	00
5	101	01
6	110	10
...

g è una funzione dai numeri naturali alle stringhe su $\{0, 1\}$ ed è suriettiva (si può intuire che qualunque stringa su $\{0, 1\}$ compare nell’immagine di g), dunque è un’enumerazione delle stringhe su $\{0, 1\}$. Inoltre, g è intuitivamente calcolabile (quindi è calcolabile da una MdT), ed è invertibile perché f è la sua funzione inversa, quindi la codifica definita da g è appunto calcolabile e invertibile.

Per mettere in evidenza il fatto che sia un’enumerazione, la funzione g viene chiamata e_{01} , e l’ i -esima stringa secondo tale enumerazione viene indicata in astratto con w_i :

$$e_{01}(i) = w_i = \begin{cases} \epsilon & \text{se } i = 0 \\ \mathbf{tail}(\mathbf{dec2bin}(i)) & \text{se } i > 0 \end{cases}$$

i	$\text{dec2bin}(i)$	$e_{01}(i) = w_i$
0	0	$\epsilon = w_0$
1	1	$\epsilon = w_1$
2	10	$0 = w_2$
3	11	$1 = w_3$
4	100	$00 = w_4$
5	101	$01 = w_5$
6	110	$10 = w_6$
...

Allora, usando la notazione introdotta prima, l'enumerazione e_{01} può essere rappresentata come:

$$e_{01} : w_0, w_1, w_2, \dots, w_n, \dots$$

1.2 Enumerazione delle macchine di Turing

Un'enumerazione e_{MdT} delle MdT può essere definita come la composizione dell'enumerazione e_{01} delle stringhe su $\{0, 1\}$ e della funzione $\#_{\text{MdT}}^{-1}$ di decodifica di una stringa in una MdT: $e_{\text{MdT}} = \#_{\text{MdT}}^{-1} \circ e_{01}$, cioè l' i -esima MdT M_i è $e_{\text{MdT}}(i) = \#_{\text{MdT}}^{-1}(e_{01}(i))$. In pratica, dal numero naturale i si ricava l' i -esima stringa binaria (secondo e_{01}), e questa stringa viene poi decodificata (tramite $\#_{\text{MdT}}^{-1}$) per ottenere l' i -esima MdT. Come al solito, l'enumerazione può essere indicata con la notazione

$$e_{\text{MdT}} : M_0, M_1, M_2, \dots, M_n, \dots$$

Avendo già visto che e_{01} e $\#_{\text{MdT}}^{-1}$ sono calcolabili e invertibili, si deduce che l'enumerazione e_{MdT} è a sua volta calcolabile e invertibile.

2 Linguaggio di diagonalizzazione

Date due enumerazioni calcolabili e invertibili, una e_{MdT} delle MdT

$$e_{\text{MdT}} : M_0, M_1, M_2, \dots, M_n, \dots$$

e una e_{01} delle stringhe su $\{0, 1\}$

$$e_{01} : w_0, w_1, w_2, \dots, w_n, \dots$$

(che potrebbero essere le specifiche enumerazioni appena presentate, o altre), si definisce il **linguaggio di diagonalizzazione** L_d come:

$$\begin{aligned} L_d &= \{w_i \in \{0, 1\}^* \mid M_i \text{ non accetta } w_i\} \\ &= \{w_i \in \{0, 1\}^* \mid w_i \notin L(M_i)\} \end{aligned}$$

Detto a parole, l' i -esima stringa nell'enumerazione e_{01} appartiene a L_d se e solo se *non* è accettata dall' i -esima MdT nell'enumerazione e_{MdT} .

L_d è chiamato linguaggio “di diagonalizzazione” perché l'accettazione di tutte le stringhe su $\{0, 1\}$ da parte di tutte le MdT può essere rappresentata in una tabella infinita,

	w_1	w_2	w_3	w_4	\dots
M_1	0	1	1	0	\dots
M_2	1	1	0	0	\dots
M_3	0	0	1	1	\dots
M_4	0	1	0	1	\dots
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots

nella quale gli indici delle righe corrispondono all'enumerazione delle MdT, gli indici delle colonne corrispondono all'enumerazione delle stringhe su $\{0, 1\}$, e una cella alla riga i e alla colonna j ha il valore

$$c_{ij} = \begin{cases} 0 & \text{se } M_i \text{ non accetta } w_j \\ 1 & \text{se } M_i \text{ accetta } w_j \end{cases}$$

Allora, le celle situate sulla diagonale che contengono valori 0 rappresentano il linguaggio L_d :

$$L_d = \{w_i \mid c_{ii} = 0\}$$

Quest'interpretazione del linguaggio di diagonalizzazione si basa su una tecnica simile a quella, chiamata appunto “tecnica di diagonalizzazione”, che fu usata da Cantor per dimostrare che l'insieme dei numeri reali non è in corrispondenza biunivoca con l'insieme dei numeri naturali.

3 Il linguaggio di diagonalizzazione non è ricorsivamente enumerabile

Si vuole dimostrare che:

Teorema: L_d non è ricorsivamente enumerabile.

Ciò significa che L_d è un esempio di un linguaggio che non può essere riconosciuto da una MdT (o, equivalentemente, da un programma in un qualunque linguaggio di programmazione).

3.1 Dimostrazione

Si suppone che L_d sia ricorsivamente enumerabile. Allora, per definizione, esiste una MdT M_d tale che $L(M_d) = L_d$. Come tutte le MdT, M_d ha un certo indice i nell'enumerazione e_{MdT} : $M_i = e_{\text{MdT}}(i) = M_d$.

Considerando ora l' i -esima stringa nell'enumerazione e_{01} , $w_i = e_{01}(i)$, ci si chiede se $w_i \in L(M_i)$.

- Se si suppone che la risposta sia “SÌ”, $w_i \in L(M_i)$, allora:

$$\begin{aligned} & \boxed{w_i \in L(M_i)} \\ & \implies M_i \text{ accetta } w_i \quad [\text{per definizione di linguaggio di una MdT}] \\ & \implies w_i \notin L_d \quad [\text{per definizione di } L_d] \\ & \implies \boxed{w_i \notin L(M_i)} \quad [\text{per } M_d = M_i \implies L_d = L(M_d) = L(M_i)] \end{aligned}$$

Partendo dall'assunzione $w_i \in L(M_i)$, si è arrivati a dedurre $w_i \notin L(M_i)$, il che è assurdo.

- Supponendo invece che la risposta sia “NO”, $w_i \notin L(M_i)$, si giunge ancora a un assurdo:

$$\begin{aligned} & \boxed{w_i \notin L(M_i)} \\ & \implies M_i \text{ non accetta } w_i \quad [\text{per definizione di linguaggio di una MdT}] \\ & \implies w_i \in L_d \quad [\text{per definizione di } L_d] \\ & \implies \boxed{w_i \in L(M_i)} \quad [\text{per } M_d = M_i \implies L_d = L(M_d) = L(M_i)] \end{aligned}$$

Riassumendo, si è partiti dall'ipotesi che L_d sia ricorsivamente enumerabile, e si è mostrato che ciò porta inevitabilmente a una contraddizione, quindi tale ipotesi deve essere scorretta: L_d non può essere ricorsivamente enumerabile, come volevasi dimostrare.