Azzolini Riccardo 2020-12-16

Alcune osservazioni sulle macchine di Turing

1 Esempi di computazioni infinite

Si considerino il linguaggio delle stringhe su $\{0,1\}$ che iniziano con uno 0, $L_{0x} = \{0x \mid x \in \{0,1\}^*\}$, e la MdT

$$M_{0x}^1 = \langle \{q_0, q_A, q_\infty\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_A\} \rangle$$

con funzione di transizione

	0	1	B
$\to q_0$	$(q_A, 0, \mathtt{R})$	$(q_\infty, 1, \mathtt{R})$	$(q_\infty, B, \mathtt{R})$
$*q_A$	$-$	$-$	$-$
q_∞	$(q_\infty, 0, \mathtt{R})$	$(q_\infty, 1, \mathtt{R})$	$(q_\infty, B, \mathtt{R})$

Su un input $w = 0x$, dove $x \in \{0,1\}^*$, la macchina M_{0x}^1 esegue la computazione

$$q_0 0x \Rightarrow 0 q_A x \qquad \delta(q_0, 0) = (q_A, 0, \mathtt{R})$$

che si arresta in uno stato finale, dunque la stringa w è accettata. Allora, M_{0x}^1 accetta tutte le stringhe di L_{0x}: $w \in L_{0x} \implies w \in L(M_{0x}^1)$.

Osservazione: La macchina accetta $0x$ avendo letto solo il primo simbolo della stringa. Questo è ragionevole, perché anche la descrizione di L_{0x} indica che l'appartenenza di una stringa al linguaggio dipende solo dal primo simbolo di tale stringa.

Invece, per le stringhe $w \notin L_{0x}$, la macchina non si arresta mai:

- su $w = 1a_1a_2 \dots a_n$, con $a_i \in \{0,1\}$ e $n \geq 0$ (il caso $n = 0$ corrisponde a $w = 1$), essa esegue la computazione

$$\begin{aligned}
q_0 1 a_1 a_2 \dots a_n &\Rightarrow 1 q_\infty a_1 a_2 \dots a_n && \delta(q_0, 1) = (q_\infty, 1, \mathtt{R}) \\
&\Rightarrow 1 a_1 q_\infty a_2 \dots a_n && \delta(q_\infty, a_1) = (q_\infty, a_1, \mathtt{R}) \\
&\vdots \\
&\Rightarrow 1 a_1 a_2 \dots a_n q_\infty B && \delta(q_\infty, a_n) = (q_\infty, a_n, \mathtt{R}) \\
&\Rightarrow 1 a_1 a_2 \dots a_n B q_\infty B && \delta(q_\infty, B) = (q_\infty, B, \mathtt{R}) \\
&\vdots
\end{aligned}$$

che continua all'infinito a spostare la testina verso destra e riscrivere ogni simbolo (prima di input, poi di blank) che incontra.

- su $w = \epsilon$ essa esegue la computazione

$$\begin{aligned} q_0 B &\Rightarrow q_\infty B \qquad \delta(q_0, B) = (q_\infty, B, \mathtt{R}) \\ &\Rightarrow q_\infty B \qquad \delta(q_\infty, B) = (q_\infty, B, \mathtt{R}) \\ &\vdots \end{aligned}$$

  anch'essa infinita.

Siccome una computazione che prosegue all'infinito non può essere accettante (quando la macchina entra in uno stato finale, si arresta), ogni stringa $w \notin L_{0x}$ non viene accettata da M^1_{0x}: $w \notin L_{0x} \implies w \notin L(M^1_{0x})$.

Bisogna però fare un'osservazione molto importante: per capire che le stringhe $w \notin L_{0x}$ non sono accettate da M^1_{0x} è stato necessario studiare "dall'esterno" le computazioni della macchina. Se invece si considerasse questa macchina come una "scatola nera", che riceve in ingresso una stringa $w \in \{0,1\}^*$ e dà in output, ad esempio, lo stato in cui una computazione è arrivata quando essa termina, allora per gli input $w \notin L_{0x}$ non ci si potrebbe aspettare alcuna risposta, alcun output. Il problema è che anche le computazioni accettanti possono avere una lunghezza arbitraria, per quanto sempre finita: quando si osserva che una computazione non è terminata dopo un certo intervallo di tempo / numero di passi dall'avvio, non si può dedurre niente sul risultato finale della computazione, perché il passo successivo potrebbe sempre essere quello che porta all'accettazione.

Riassumendo le considerazioni precedenti, si ha che $w \in L_{0x} \iff w \in L(M^1_{0x})$, ovvero che $L_{0x} = L(M^1_{0x})$, quindi L_{0x} è per definizione un linguaggio *ricorsivamente enumerabile*. Invece, dall'esistenza di M^1_{0x} non si può dedurre che L_{0x} sia decidibile (anche se lo è, come si vedrà in seguito), perché non è vero che M^1_{0x} termina per ogni input.

Per dimostrare che L_{0x} è decidibile, si considera un'altra MdT

$$M^2_{0x} = \langle \{q_0, q_A, q_R\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_A\} \rangle$$

con funzione di transizione

	0	1	B
$\rightarrow q_0$	$(q_A, 0, \mathtt{R})$	$(q_R, 1, \mathtt{R})$	$(q_R, B, \mathtt{R})$
$*q_A$	–	–	–
q_R	–	–	–

M^2_{0x} è ottenuta da M^1_{0x} sostituendo lo stato q_∞ con uno stato q_R (la R sta per "rifiuto"), nel quale la computazione si blocca invece di proseguire all'infinito. Di conseguenza, per qualunque input $w \in \{0,1\}^*$, questa macchina si arresta sempre dopo un singolo passo, cioè dopo aver letto un solo simbolo:

- la computazione su $w = 0x$, con $x \in \{0,1\}^*$, è uguale a quella di M^1_{0x},

$$q_0 0x \Rightarrow 0 q_A x \qquad \delta(q_0, 0) = (q_A, 0, \mathtt{R})$$

  dunque la stringa viene accettata;

- su $w = \epsilon$, M^2_{0x} esegue il passo

$$q_0 B \Rightarrow q_R B \qquad \delta(q_0, B) = (q_R, B, \mathtt{R})$$

  e poi si blocca, rifiutando la stringa w;

- su $w = 1x$, con $x \in \{0,1\}^*$, M^2_{0x} esegue il passo

$$q_0 1x \Rightarrow 1 q_R x \qquad \delta(q_0, 1) = (q_R, 1, \mathtt{R})$$

  e poi si blocca, rifiutando la stringa w.

Poiché M^2_{0x} si arresta per ogni input e accetta il linguaggio $L(M^2_{0x}) = L_{0x}$, si può affermare che L_{0x} è *decidibile*.

2 Calcolo di funzioni sui numeri naturali

Le macchine di Turing sono qui state introdotte come riconoscitori di linguaggi, ma possono essere utilizzate anche come strumenti per calcolare funzioni sui numeri naturali. In particolare, a una MdT $M = \langle Q, \{0,1\}, \Gamma, \delta, q_0, B, F\rangle$ (avente alfabeto di input $\{0,1\}$, per semplicità) può essere associata una funzione parziale di arità k (a k argomenti) sui numeri naturali,

$$f^{(k)}_M : \mathbb{N}^k \to \mathbb{N} \cup \{\perp\}$$

stabilendo:

- un'opportuna codifica sull'alfabeto $\{0,1\}$ delle k-uple di numeri naturali in input;
- un'opportuna "decodifica" del contenuto del nastro della macchina quando la computazione termina;
- che la funzione ha come risultato $\perp$, *indefinito*, se la computazione non termina (però, come osservato prima, per stabilire che $f^{(k)}_M(n_1, \ldots, n_k) = \perp$ non è sufficiente mandare in esecuzione la macchina e attendere un risultato: serve un ragionamento esterno, al "metalivello", sulle computazioni).

Osservazione: A una stessa MdT M può essere associata una funzione $f^{(k)}_M$ per ogni possibile arità k, a seconda che si fornisca in input la codifica di un singolo numero naturale ($k = 1$), o di una coppia di naturali ($k = 2$), ecc.

2.1 Esempio di codifica

I numeri naturali possono essere rappresentati in notazione *unaria*, scegliendo ad esempio di rappresentare il numero $n \in \mathbb{N}$ tramite la stringa 1^{n+1} (si usa una sequenza di $n+1$ uni, e non di n uni, per fare in modo che la rappresentazione di 0 non sia una stringa vuota, che risulterebbe scomoda da gestire):

$$0 \to 1,\ 1 \to 11,\ 2 \to 111,\ \ldots$$

La k-upla $(n_1, \ldots, n_k)$ di argomenti della funzione viene poi rappresentata usando il simbolo 0 come separatore tra le codifiche unarie dei componenti $n_1, \ldots, n_k$:

$$w = \#(n_1, n_2, \ldots, n_k) = 1^{n_1+1}\, 0\, 1^{n_2+1}\, 0\ \ldots\ 0\, 1^{n_k+1}$$

Se la MdT M si arresta su questo input w, allora l'**output** di M (il valore calcolato da M) è il numero m di simboli 1 presenti sul nastro.

Adottando queste convenzioni, si può definire la funzione a $k \geq 1$ argomenti calcolata da M,

$$f_M^{(k)} : \mathbb{N}^k \to \mathbb{N} \cup \{\perp\}$$

nel modo seguente:

$$f_M^{(k)}(n_1, \ldots, n_k) = \begin{cases} m & \text{se su input } \#(n_1, \ldots, n_k)\ M \text{ si arresta con output } m \\ \perp & \text{se su input } \#(n_1, \ldots, n_k)\ M \text{ non si arresta} \end{cases}$$

2.2 Funzioni calcolabili

Si dice che una funzione $f : \mathbb{N}^k \to \mathbb{N} \cup \{\perp\}$ è (**Turing**) **calcolabile** se esiste una MdT $M = \langle Q, \{0,1\}, \Gamma, \delta, q_0, B, F \rangle$ che la calcola, cioè tale che $f_M^{(k)} = f$.

Una funzione è detta **calcolabile totale** se esiste una MdT che la calcola e che si arresta per ogni input.