Esercitazione 2 - Linguaggi e Calcolabilità

05-04-2019

Antonio Cruciani
antonio.cruciani@alumni.uniroma2.eu

Esercizi a lezione

Esercizio 1:
Sia

$$L_{XTM} = \{\langle T\rangle\ : \text{T NON ACCETTA } \langle T\rangle\}$$

Si discuta la decidibilità di tale linguaggio
Esercizio 2 (The Accepting Problem):
Un quesito che ci si pone in modo naturale è il seguente:

"Esiste una macchina di Turing in grado di predire se tutte le altre macchine di Turing terminano nello stato di accettazione?".

Informalmente, ci stiamo chiedendo se esiste una macchina di Turing T che data una qualsiasi altra macchina di Turing M e una parola x, essa riesce a "capire" se la computazione $M(x)$ termina in $\boldsymbol{q_a}$.
Precisiamo che questo quesito è diverso da quello che ci si pone nell' Halting Problem in quanto, in quest'ultimo, ci si chiede:

"Esiste una macchina di Turing in grado di predire se le altre macchine di Turing, data una qualsiasi parola in input, terminano ? ".

La differenza sostanziale tra questi due quesiti è che nel primo siamo interessati a capire, dato un input x, in che **stato finale** terminano le macchine di Turing e nel secondo, invece, si è interessati a capire se, dato un input x, le macchine di Turing **terminano**.

Dopo questa breve precisazione, torniamo all'*Accepting Problem* e verifichiamo se effettivamente esiste una macchina di Turing che permetta di predire l'output di tutte le altre macchine di Turing.

Per rispondere a questo quesito possiamo definire il seguente linguaggio e studiarne l'accettabilità e la decidibilità.
Sia

$$L_{ATM} = \{(\langle T\rangle, x)\ :\ \text{T ACCETTA X}\}$$

Si discuta l'accettabilità e la decidibilità di tale linguaggio.
(**HINT:** Le macchine di Turing possono essere codificate come parole)

Esercizio 3:
Si consideri il seguente linguaggio (versione modificata dell'Halting Problem):

$$L_{BH} = \{(\langle T \rangle, x) : T(x) \text{ Termina in } |x| \text{ passi}\}$$

Si dimostri se L_{BH} è decidibile o non decidibile.

Esercizio 4:
Sia $L_1 \subseteq \Sigma^*$ un linguagggio decidibile e sia $L_2 \subseteq \Sigma^*$ un linguaggio accettabile ma non decidibile. Detta T_1 la macchina di Turing che decide L_1 e detta T_2 la macchina di Turing che accetta L_2, si consideri il linguaggio $L \subseteq \Sigma^* \times \mathbb{N}$ di seguito definito:

$$L = \{(x, k) : x \in \Sigma^* \wedge k \in \mathbb{N} \wedge [x \notin L_1 \vee (x \notin L_2 \wedge T_2(x) \text{ rigetta in k passi })]\}$$

Si dimostri se L è un linguaggio accettabile o decidibile.

<u>Informazioni Notazione:</u>

Data una macchina di Turing T, con $\langle T \rangle$ indichiamo la sua **codifica**.

Con $\mathcal{T}$ indichiamo l'insieme delle macchine di Turing

Soluzioni esercizi a lezione

Esercizio 1:
Mostriamo che L_{XTM} non è decidibile.

Dimostrazione 1:
Assumiamo per assurdo che L_{XTM} sia decidibile $\Rightarrow \exists T_{XTM}, \forall \langle T \rangle \in \mathcal{T}$

$$O_{T_{XTM}}(\langle T \rangle) = \begin{cases} q_a & \text{SE } \langle T \rangle \in L_{XTM} \\ q_r & \text{SE } \langle T \rangle \in L^c_{XTM} \end{cases}$$

Osserviamo esplicitamente che:

1. Se $O_{T_{XTM}}(\langle T \rangle) = q_a \Rightarrow \langle T \rangle \notin L_{XTM}$
2. Se $O_{T_{XTM}}(\langle T \rangle) = q_r \Rightarrow \langle T \rangle \in L_{XTM}$

Abbiamo quindi che:

$$\langle T \rangle \in L_{XTM} \iff O_{T_{XTM}}(\langle T \rangle) = q_r \iff \langle T \rangle \notin L_{XTM}$$

- $O_{T_{XTM}}(\langle T \rangle) = q_r \iff \langle T \rangle \in L_{XTM}$ segue dalla definizione di L_{XTM}.
- $O_{T_{XTM}}(\langle T \rangle) = q_r \iff \langle T \rangle \notin L_{XTM}$ segue dall' ipotesi.

Abbiamo che:

$$\langle T \rangle \in L_{XTM} \iff \langle T \rangle \notin L_{XTM}$$

Assurdo! Quindi L_{XTM} non è decidibile.

Dimostrazione 2:
Dimostriamo la non decidibilità del linguaggio mediante diagonalizzazione.
Si supponga di costruire una matrice infinita $\mathbb{A}$ dove elenchiamo tutte le macchine di Turing e tutte le loro codifiche. Dove, dati i, j generici abbiamo che

$$a_{i,j} = \begin{cases} 1 & \text{SE } T_i(\langle T_j \rangle) \text{ ACCETTA} \\ 0 & \text{SE } T_i(\langle T_j \rangle) \text{ RIGETTA} \end{cases}$$

(La disposizione degli 1 e degli 0 è indifferente, la chiave è lo schema della codifica)

	$\langle T_1\rangle$	$\langle T_2\rangle$	$\langle T_3\rangle$	$\langle T_4\rangle$	$\langle T_5\rangle$	$\langle T_6\rangle$	$\dots$
T_1	0	1	0	0	1	1	$\dots$
T_2	1	1	0	1	0	0	$\dots$
T_3	0	0	0	0	0	0	$\dots$
T_4	1	1	1	1	1	1	$\dots$
T_5	0	1	0	0	1	1	$\dots$
T_6	1	0	0	1	1	1	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Prendiamo la diagonale della matrice $\mathbb{A}$ (ottenendo una sequenza infinita di 1 e 0), invertiamo gli 0 con gli 1 e viceversa. Dall' esempio abbiamo che la diagonale della matrice è $010111\dots$, invertendo gli 1 con gli 0 e viceversa otteniamo $101000\dots$.
Osserviamo che la nuova sequenza (ie: 101000) non compare in nessuna riga della matrice in quanto la nuova sequenza differisce dalla riga i-esima della matrice proprio per l'i-esimo elemento.
Quindi abbiamo ottenuto una sequenza caratteristica di un linguaggio che non è riconoscibile da nessuna macchina di Turing poiché differisce da ogni riga della matrice.
Riflettendo un momento, si può osservare che il linguaggio descritto dalla sequenza ottenuta invertendo gli elementi della diagonale descrive proprio L_{XTM}.
Quindi possiamo concludere che L_{XTM} non è decidibile.

Esercizio 2:
Osserviamo esplicitamente che il linguaggio è accettabile in quanto, dati $\langle T\rangle$ e x possiamo simulare $T(x)$ e verificare se tale computazione è accettante.
Il linguaggio, però, non è decidibile.
Si supponga, per assurdo, che esista una macchina di Turing T_{ATM} in grado di decidere L_{ATM}, allora potremmo derivare da T_{ATM} una nuova macchina, diciamo T_1, in grado di decidere L_{XTM}. Ovvero T_1 è definita come segue:
Essa è una macchina a due nastri:

N_1) Input $\langle P\rangle$

N_2) Simulazione della computazione $T_{ATM}(\langle P\rangle, \langle P\rangle)$

Descriviamo il funzionamento della macchina T_1:

- dato in input $\langle P \rangle$

1) Simula T_{ATM} con input $(\langle P \rangle, \langle P \rangle)$
2) Se T_{ATM} accetta allora **Rigetta**
3) Se T_{ATM} rigetta allora **Accetta**

Chiaramente, T_1 è una macchina di Turing che decide L_{XTM}. Ma sappiamo che tale linguaggio non è decidibile! Quindi, poiché l'esistenza di T_{ATM} implica l'esistenza di T_1 possiamo concludere che T_{ATM} non può esistere e quindi che L_{ATM} è accettabile ma non decidibile.

Esercizio 3:
Mostriamo che L_{BH} è un linguaggio decidibile.
Esso è una modifica dell'Halting Problem il quale è non decidibile, però osserviamo esplicitamente che il linguaggio L_{BH} è l'insieme delle macchine di Turing che terminano su input x in $|x|$ passi.
Possiamo allora definire una macchina di Turing T_{BH} che decide tale linguaggio, descriviamola.
T_{BH} sarà, senza perdita di generalità, una macchina di Turing a tre nastri, dove sul primo nastro sarà presente l'input $(\langle T \rangle, x)$, sul secondo verrà eseguita la simulazione della computazione $T(x)$ e sul terzo nastro ci sarà la lunghezza di x, senza perdita di generalità, assumiamo che $|x|$ sia scritta in unario sul terzo nastro (e che la testina sul terzo nastro sia posizionata sul $\square$ situato prima del primo 1 a sinistra).
Illustriamo, ora, il funzionamento della macchina T_{BH}:
Su input $(\langle T \rangle, x)$,scrivi in unario $|x|$ sul terzo nastro (posizionando la testina di N_3 sul $\square$ situato prima del primo 1 a sinistra), poi simula $T(x)$ e ad ogni passo di tale simulazione sposta a destra la testina sul terzo nastro. Se $T(x)$ termina e sul terzo nastro leggo un 1 allora T_{BH} accetta. Se $T(x)$ non è ancora terminata e sul terzo nastro leggo $\square$ allora T_{BH} rigetta in quanto $T(x)$ non è terminata entro $|x|$ passi.
Osserviamo T_{BH} decide L_{BH} in quanto :

$$O_{T_{BH}}(\langle T \rangle, x) = \begin{cases} q_a & \text{SE } (\langle T \rangle, x) \in L_{BH} \\ q_r & \text{SE } (\langle T \rangle, x) \in L_{BH}^c \end{cases}$$

Quindi L_{BH} è un linguaggio decidibile.

Esercizio 4:
Osserviamo che il linguaggio è decidibile.
ASSUNZIONE: Ogni coppia (x, k) sarà ben formata, ovvero $\forall(x, k)$ avremo sempre che $x \in \Sigma^* \wedge k \in \mathbb{N}$. Quest'assunzione serve per facilitare l'analisi del linguaggio e può essere, chiaramente, rilassata.

Per argomentare tale claim basta osservare che possiamo definire il linguaggio L come l'unione di due linguaggi:

$$L_a = \{(x, k) : x \in \Sigma^* \wedge k \in \mathbb{N} \wedge \ x \in L_1^c\}$$
$$L_b = \{(x, k) : x \in \Sigma^* \wedge k \in \mathbb{N} \wedge (x \in L_2^c \wedge T_2(x) \text{ RIGETTA IN K PASSI}\}$$

Osserviamo che $L = L_a \cup L_b$.
Banalmente L_a è un linguaggio decidibile in quanto può essere deciso definendo una macchina di Turing T_a che esegue le seguenti operazioni:

- Input (x,k)

1) Simula la macchina di Turing T_1 con input x se:
   - $T_1(x)$ Accetta allora T_a **Rigetta**
   - $T_1(x)$ Rigetta allora T_a **Accetta**

Osserviamo esplicitamente che la simulazione a il punto 1) termina sempre in quanto L_1 è un linguaggio decidibile.
Discutiamo ora L_b. Esso è un linguaggio decidibile, argomentiamo:
Mostriamo che esiste una macchina di Turing T_b a 3 nastri che decide L_b. Sia T_b definita come segue:

N_1) Input (x,k)

N_2) Simulazione della computazione $T_2(x)$

N_3) codifica unaria del numero k

Descriviamo, ora, il funzionamento della macchina T_b

- Su Input (x,k)

1) Scrivi k in unario sul nastro N_3.
2) Posiziona la testina del nastro N_3 sul primo $\square$ a sinistra prima del primo 1.
3) Simula $T_2(x)$ e ad ogni passo della simulazione sposta la testina di N_3 di una posizione a destra.

- Se
  a) $T_2(x)$ Accetta e sul nastro N_3 legge 1 allora T_b **Rigetta**
  b) $T_2(x)$ Rigetta e sul nastro N_3 legge un 1 allora T_b **Accetta**
  c) $T_2(x)$ Non è terminata e sul nastro N_3 legge $\square$ allora T_b **Rigetta**

Osserviamo esplicitamente che tale macchina T_b riesce a decidere L_b. Ricordiamo che L_b è il linguaggio composto delle parole che appartengono a L_2^c tali che la computazione di $T_2(x)$ rigetta in k passi. Sappiamo che L_2 è accettabile ma non decidibile, il che significa che esiste una macchina di Turing in grado di accettare $\forall x \in L_2$ e che però per quanto riguarda le parole $x \in L_2^c$ per la computazione $T_2(x)$ non sappiamo cosa accade, la macchina T_2 potrebbe rigettare o non terminare. Quindi se $x \in L^c$ e $T_2(x)$ rigetta (per qualche mistico motivo) entro k passi allora T_b Accetta in quanto $x \in L_b$. Se invece $x \in L_2^c$ ma $T_2(x)$ non rigetta entro k passi allora T_b rigetta in quanto $x \in L_b^c$ (gli altri casi si possono dedurre dalla macchina di Turing descritta sopra).

Dopo questa breve analisi, possiamo studiare il linguaggio L che abbiamo definito come $L = L_a \cup L_b$, abbiamo mostrato che sia L_a *che* L_b sono linguaggi decidibili, quindi possiamo dire che L è un linguaggio decidibile. Definiamo la macchina di Turing che decide L:

- Input (x,k)

1) simula $T_a(x, k)$ se T_a accetta allora **Accetta**, se rigetta esegui passo 2)
2) Simula $T_b(x, k)$ se T_b accetta allora **Accetta**, se T_b rigetta allora **Rigetta**