

Soluzioni Tutorato 6

Giulio Umbrella

Linguaggi Decidibili

Ex 1

Sia $ALL_{DFA} = \{ A \mid A \text{ e' un DFA che } L(A) = \Sigma^* \}$. Mostrare che ALL_{DFA} e' decidibile.

Soluzione

Per dimostrare che un linguaggio e' decidibile dobbiamo costruire un TM che accetti o rifiuti la stringa in input in un numero finito di passi.

Per questo esercizio possiamo usare una TM che simuli il funzionamento del DFA come visto a lezione.

Opz1 Come primo tentativo, possiamo considerare tutte le stringhe in Σ^* e verificare se l'automa le accetta oppure no. Se una stringa non appartiene al linguaggio, l'automa la rifiuta e la computazione si blocca. Il problema e' che questa soluzione potrebbe non terminare mai se l'automa A accetta tutte le stringhe, mentre il testo ci chiede di dimostrare che il linguaggio e' decidibile, ossia la computazione si interrompe sicuramente.

Opz2 In alternativa, possiamo usare i risultati visti a lezione sui linguaggi decidibili.

1. Determinare se un automa accetta il linguaggio vuoto e' un problema decidibile.
2. Il complementare dell'automa che accetta il linguaggio composto da tutte le parole possibile e' automa che accetta il linguaggio vuoto.
3. I linguaggi regolari sono chiusi rispetto all'operazione complemento.

Costruiamo una TM che agisce nel seguente modo:

1. Convertire il DFA nel suo complementare
2. Verificare che il linguaggio accettato dal complementare sia il vuoto

Entrambe le operazioni sono decidibili. Per ottenere il complementare invertiamo stati accettanti e non accettanti del DFA, quindi abbiamo un numero finito di operazioni.

Se la TM accetta il complementare, allora il linguaggio di partenza e' formato l'automa di partenza.

Abbiamo quindi costruito un decisore per il linguaggio e quindi il problema e' decidibile.

Ex 2

Sia $SREX = \{ \langle R, S \rangle \mid R \text{ e } S \text{ sono espressioni regolari tali che } L(R) \text{ e' un sottoinsieme di } L(S) \}$. Mostrare che $SREX$ e' decidibile.

Soluzione

Anche per questo caso potremmo verificare se una stringa del linguaggio appartiene a $L(R)$ ma non $L(S)$, ma andremmo incontro allo stesso problema dell'EX1; se $L(R)$ e' effettivamente un sottoinsieme di $L(S)$, la computazione non terminera' mai, violando la consegna.

Come per EX 1, facciamo riferimento a risultati noti.

1. Per prima cosa ricordiamo la seguente proprieta' degli insiemi: se A e' sotto insieme di B , l'intersezione fra A e B e' uguale all'insieme A , $A \subset B \rightarrow A \cap B = A$.
2. Convertire un'espressione regolare nel suo corrispettivo DFA e' un problema decidibile; piu' precisamente la concatenazione di due operazioni decidibili, il passaggio da RE a NFA e da NFA a RE.
3. Verificare se due DFA riconoscono lo stesso linguaggio e' un problema decidibile.
4. L'intersezione fra due linguaggi regolari e' un linguaggio regolare (proprieta' di chiusura)

Costruiamo quindi un automa che opera nel modo seguente:

1. Dato l'input con le due espressioni regolari, produce per ciascuno di essi i rispettivi DFA, denominati A e B
2. Costruiamo il DFA che riconosce l'intersezione di A e B
3. Verifichiamo che $L(A) \cap L(B) = L(A)$.

Se la TM accetta la stringa, $L(R)$ e' sottoinsieme di $L(S)$.

Abbiamo quindi costruito un decisore per il linguaggio e quindi il problema e' decidibile.

Ex 3

Sia $A_{\epsilon CFG} = \{ G \mid G \text{ e' CFG che genera la parola vuota } \epsilon \}$. Mostrare che $A_{\epsilon CFG}$ e' decidibile.

Soluzione

Dato che G e' una CFG, possiamo convertirla in forma normale di Chomsky. Possiamo quindi usare i due seguenti risultati:

1. G e la sua forma normale riconoscono lo stesso linguaggio.
2. La forma normale ha come produzione $S \rightarrow \epsilon$ per la variabile iniziale.

Quindi costruiamo una TM che opera nel seguente modo:

1. Convertiamo la grammatica G nella sua forma normale
2. Verifichiamo se la forma normale contiene la produzione $S \rightarrow \epsilon$

Entrambe le operazioni sono decidibili.

Se la produzione e' presente, la parola appartiene al linguaggio e la TM la accetta. TM e' quindi un decisore.

Linguaggi Turing Riconoscibili

Ex 4

Sia $ETM = \{ M \mid M \text{ e' una TM tale che } L(M) = \emptyset \}$. Mostrare che ETM^C , il complemento di ETM , e Turing-riconoscibile.

Soluzione

Per prima cosa, cerchiamo di capire che cosa rappresenta l'insieme ETM . Il linguaggio e' composto da TM, quindi possiamo definire $ETM = \{ TM_1, TM_2, TM_3, \dots \}$ dove ciascuna TM_i e' una TM completa, che contiene al suo interno anche la descrizione dell'alfabeto Σ_i .

Ogni elemento dell'insieme ETM e' una stringa che contiene la specifica di una TM. La particolarita' di questo insieme e' che tutte le TM riconoscono il linguaggio vuoto ossia data una TM_i per ciascuna stringa su Σ_i la computazione non termina in nessuno stato accettante.

Ora possiamo definire l'insieme complementare ETM^C che contiene tutte le TM tali che $L(M) \neq \emptyset$. Quindi ciascun TM ha almeno una computazione che termina in uno stato accettante. Anche per ETM^C abbiamo che i suoi componenti sono TM che contengono l'alfabeto su cui sono definite. Per ciascuna TM_i in ETM^C esiste almeno una stringa in Σ_i riconosciuta da TM_i .

Per dimostrare che il linguaggio ETM^C e' Turing-riconoscibile, dobbiamo dimostrare che esiste una TM^* in grado di riconoscere il linguaggio. Quindi TM^* e' una macchina di Turing che prende in input un'altra TM e ne simula il funzionamento. Data una TM_i in ETM^C , TM^* prova tutte le combinazioni sull'alfabeto finche' non trova una stringa accettata.

Notare che la computazione potrebbe entrare in loop ma questa eventualita' e' ammessa nel caso dei linguaggi Turing riconoscibili.

Varianti Macchine di Turing

Ex 5

Turing machine con alfabeto binario: vedi esame scritto del 1 Settembre 2021.