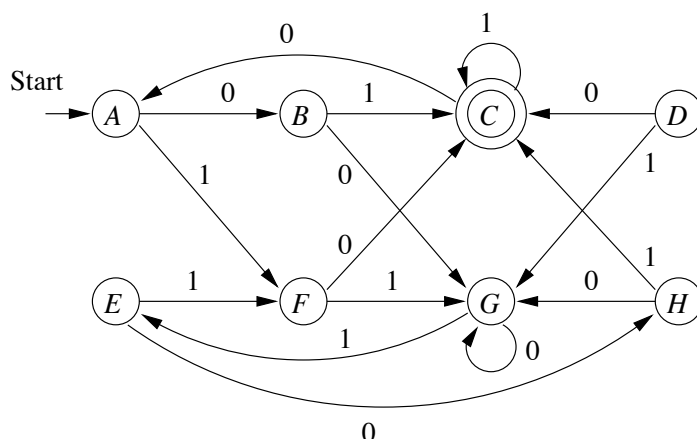


Teoria degli automi e calcolabilità a.a. 2022/23

Prova scritta 3 febbraio 2023

Esercizio 1 Minimizzare il seguente DFA, descrivendo in modo **molto preciso** i passaggi effettuati:



Soluzione Possiamo anzitutto notare che lo stato D non è raggiungibile dallo stato iniziale, quindi possiamo eliminarlo. Inizialmente abbiamo quindi le due classi $\{A, B, E, F, G, H\}$ dei non finali e $\{C\}$ dei finali. Leggendo 0 possiamo discriminare $\{A, B, E, G, H\}$ (che vanno in stati non finali) da $\{F\}$ (che va in stati finali). Abbiamo quindi le classi $\{A, B, E, G, H\}$, $\{F\}$, e $\{C\}$. Leggendo 1 possiamo discriminare $\{A, E\}$ (che vanno in $\{F\}$) da $\{B, H\}$ (che vanno in $\{C\}$) e $\{G\}$ (che va in $\{A, B, E, G, H\}$). Abbiamo quindi le classi $\{A, E\}$, $\{B, H\}$, $\{G\}$, $\{F\}$, e $\{C\}$. Non si può discriminare ulteriormente.

Esercizio 2 Si provi che il linguaggio $\{a^n b^n c^m \mid 0 \leq n \leq m\}$ non è context free.

Soluzione Possiamo dimostrarlo utilizzando il pumping lemma. Preso n arbitrario, consideriamo la stringa $a^n b^n c^n$ che appartiene al linguaggio ed ha lunghezza $\geq n$. Vi sono molti diversi casi di decomposizione di questa stringa come $uvwxy$. Tuttavia, dato che la lunghezza di vw deve essere $\leq n$, è facile capire che tale sottostringa *non* può contenere sia a che c , perché in tal caso dovrebbe contenere tutti i b . Si hanno quindi i seguenti casi:

- contiene a e/o b , ma non c : prendendo uv^2wx^2y , si ottiene una stringa in cui il numero di a e/o di b è maggiore del numero di c
- contiene b e/o c , ma non a : prendendo uv^0wx^0y , si ottiene una stringa in cui il numero di b e/o di c è minore del numero di a .

Esercizio 3 Si dia una macchina di Turing che effettua una copia di una stringa unaria. Più precisamente, a partire da una configurazione iniziale $\dots BB \underbrace{1 \dots 1}_n BB \dots$, con la testina posizionata

sul primo 1 (se $n \geq 1$), la macchina si deve arrestare in una configurazione $\dots BB \underbrace{1 \dots 1}_n B \underbrace{1 \dots 1}_n BB \dots$

È assolutamente necessario dare prima una descrizione a parole dell'algoritmo, e solo successivamente la matrice di transizione, preferibilmente usando nomi significativi per gli stati.

Soluzione L'algoritmo sostituisce via via gli 1 della configurazione iniziale con blank, e li va a duplicare sia a destra del blank alla fine di questi, sia a sinistra dell'inizio. Una configurazione intermedia sarà quindi del tipo $\underbrace{1 \dots 1}_n B \underbrace{1 \dots 1}_m B \underbrace{1 \dots 1}_n$. Quando arrivo all'ultimo 1, ossia ho $\underbrace{1 \dots 1}_n B \underbrace{1 \dots 1}_n$, lo elimino, lo aggiungo a destra e a sinistra e ho finito. In dettaglio:

1. se in q_0 trovo un 1 lo cancello, scorro verso destra la stringa degli 1 ancora da elaborare, un blank, la stringa degli 1 già inseriti (se in q_0 trovo un blank ho finito)
2. trovo un blank e lo sostituisco con 1, poi scorro verso sinistra la stringa degli 1 inseriti, un blank, la stringa degli 1 ancora da elaborare
3. trovo un blank, lo sostituisco con 1, vado a destra e riparto in q_0

Tabella:

	1	B
q_0	q_1, B, R	
$\text{goR} \equiv q_1$	$q_1, 1, R$	q_2, B, R
$\text{goRR} \equiv q_2$	$q_2, 1, R$	$q_3, 1, L$
$\text{goL} \equiv q_3$	$q_3, 1, L$	q_4, B, L
$\text{goLL} \equiv q_4$	$q_4, 1, L$	$q_0, 1, R$

Tabella nel formato del simulatore:

```
0 1 _ r goR ; cancello il primo 1 e vado a destra - se non ci sono 1 ho finito

goR 1 1 r goR ; scorro tutti gli 1 ancora da cancellare verso destra
goR _ _ r goRR ; passo a scorrere tutti gli 1 aggiunti a destra verso destra

goRR 1 1 r goRR ; scorro tutti gli 1 aggiunti a destra verso destra
goRR _ 1 l goL ; aggiungo un 1 a destra e torno indietro

goL 1 1 l goL ; scorro tutti gli 1 aggiunti a destra verso sinistra
goL _ _ l goLL ; passo a scorrere gli 1 da cancellare verso sinistra

goLL 1 1 l goLL ; scorro tutti gli 1 ancora da cancellare verso sinistra
goLL _ 1 r 0 ; trovo un blank, lo sostituisco con 1, vado a destra e ricomincio
```

Notazione Nei prossimi due esercizi indichiamo per brevità con W_x il dominio di definizione di ϕ_x , ossia $W_x = \{y \mid \phi_x(y) \text{ definita}\}$, ossia W_x è l'insieme degli input sui quali il programma \mathcal{M}_x termina. Quindi, per esempio, $\mathcal{K} = \{x \mid x \in W_x\}$.

Esercizio 4 Si dica se le seguenti affermazioni sono vere o false, motivando la risposta.

1. Un insieme A tale che $A = W_x$ per qualche x è ricorsivamente enumerabile?
2. L'insieme $\mathcal{X} = \{x \mid \{1, 2, 3\} \subseteq W_x\}$ è ricorsivamente enumerabile?
3. L'insieme $\mathcal{F} = \{x \mid W_x \text{ finito}\}$ è ricorsivo?

Soluzione

1. Sì, infatti un insieme è r.e. se è il dominio di definizione di una funzione ricorsiva. In altre parole, l'algoritmo che semidecide se $y \in A$ consiste nell'eseguire $\mathcal{M}_x(y)$.
2. Sì, infatti un algoritmo che fornisce la risposta positiva se $x \in \mathcal{X}$ consiste nell'eseguire successivamente il programma \mathcal{M}_x sui tre input.
3. No, per il teorema di Rice, in quanto si tratta di un insieme estensionale e non banale.

Esercizio 5 Si dia una riduzione da $\overline{\mathcal{K}} = \{x \mid x \notin W_x\}$ (l'insieme dei programmi che non terminano su se stessi) in $\mathcal{F} = \{x \mid W_x \text{ finito}\}$ (l'insieme dei programmi che terminano su un insieme finito di input). Cosa possiamo concludere dall'esistenza di tale riduzione?

Soluzione L'input del problema $\overline{\mathcal{K}}$ è (la descrizione di) un algoritmo x , e dobbiamo trasformarlo in un nuovo algoritmo $g(x)$ in modo tale che $x \notin W_x$ se e solo se $\phi_{g(x)}$ è definita su un insieme finito di input. Questo si può ottenere costruendo l'algoritmo $g(x)$ nel modo seguente:

input $y \rightarrow$ restituisco $\mathcal{M}_x(x)$

Allora g è una funzione di riduzione da $\overline{\mathcal{K}}$ in \mathcal{F} , in quanto è calcolabile, totale, e si ha:

se $x \in \overline{\mathcal{K}}$, $\mathcal{M}_x(x)$ non termina, quindi $W_{g(x)} = \emptyset$, quindi $g(x) \in \mathcal{F}$
 se $x \in \mathcal{K}$, $\mathcal{M}_x(x)$ termina, quindi $W_{g(x)} = \mathbb{N}$, quindi $g(x) \notin \mathcal{F}$

Dal fatto che $\overline{\mathcal{K}}$ sia riducibile a \mathcal{F} possiamo concludere che \mathcal{F} non è ricorsivamente enumerabile.

Guida alla correzione

Es.1 -2 a chi non ha diviso in tre classi leggendo 1; -0,5 a chi non ha eliminato lo stato D

Es.2 4 a chi ha detto almeno una stringa sensata; 7 punti a chi non si è accorto che ci volevano due casi; 8-8,5 a chi almeno ha provato a fare una casistica

Es.4.2 8-9 a chi ha capito ma ha detto in interleaving o a zig-zag (non serviva)

Es.5.1 8 a chi ha scritto almeno in modo sensato come doveva essere la riduzione