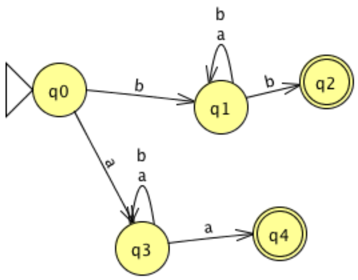
Teoria degli automi e calcolabilità a.a. 2022/23
Prova scritta 13 gennaio 2023

Esercizio 1 Si consideri il seguente automa a stati finiti non deterministico sull'alfabeto $\{a, b\}$.



1. Si spieghi perché la stringa $abab$ non è accettata.
2. Si scriva l'automa in formato tabellare e lo si trasformi in un automa deterministico.
3. Si descriva (in modo preciso) il linguaggio riconosciuto.

Soluzione

1. Le computazioni possibili per la stringa $abab$ sono:

$$\langle q_0, abab\rangle \to \langle q_3, bab\rangle \to \langle q_3, ab\rangle \to \langle q_3, b\rangle \to \langle q_3, \epsilon\rangle$$
$$\langle q_0, abab\rangle \to \langle q_3, bab\rangle \to \langle q_3, ab\rangle \to \langle q_4, b\rangle$$

Entrambe queste computazioni non accettano (la stringa la prima termina in uno stato non finale, la seconda è bloccata), quindi la stringa è rifiutata.

2. Applichiamo la trasformazione che elimina il non determinismo:

	a	b
$\to q_0$	q_3	q_1
q_1	q_1	q_1, q_2
$\star q_2$		
q_3	q_3, q_4	q_3
$\star q_4$		

$\Rightarrow$

	a	b
$\to \{q_0\}$	$\{q_3\}$	$\{q_1\}$
$\{q_3\}$	$\{q_3, q_4\}$	$\{q_3\}$
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\star F_1 \equiv \{q_3, q_4\}$	$\{q_3, q_4\}$	$\{q_3\}$
$\star F_2 \equiv \{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$

3. Il linguaggio riconosciuto è l'insieme delle stringhe (di lunghezza almeno 2) che iniziano e terminano con la stessa lettera.

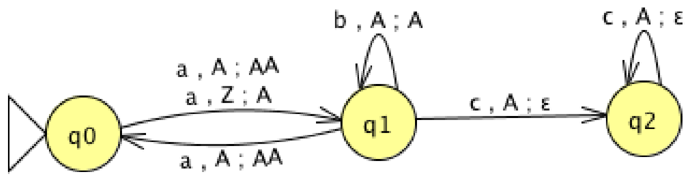
Esercizio 2 Si consideri il linguaggio $L = \{a^n b^m c^n \mid n \text{ dispari}\}$.

1. Si dia un automa a pila, se possibile deterministico, che riconosca il linguaggio (per pila vuota), spiegando su quale idea intuitiva è basato.
2. Il linguaggio è regolare? Si giustifichi formalmente la risposta.

Soluzione

1. Inizialmente, l'automa mette nella pila un simbolo per ogni a letta (almeno una), utilizzando due stati per controllare che il numero sia dispari. Poi legge le b, se ce ne sono, infine legge le c, controllando che siano in numero uguale alle a togliendo ogni volta un simbolo dalla

pila. Questo automa è deterministico.



2. Il linguaggio non è regolare, proviamolo con il pumping lemma. Fissato $n \geq 0$, consideriamo per esempio la stringa $a^n c^n$ se n dispari, altrimenti $a^{n+1}c^{n+1}$, che appartiene al linguaggio. Decomponendo la stringa in uvw, con $|uv| \leq n$ e $|v| > 0$, si ha necessariamente che u e v sono formate di sole a. Quindi, per esempio, la stringa uv^0w ha un numero di a strettamente minore delle c e quindi non appartiene al linguaggio.

Esercizio 3 Si definiscano i predicati primitivi ricorsivi `odd-and`, `odd-or` e `odd-and-leq` tali che $\texttt{odd-and}(\texttt{x}, \texttt{y}) = 1$ se `x` e `y` sono entrambi dispari, $\texttt{odd-or}(\texttt{x}, \texttt{y}) = 1$ se almeno uno dei due è dispari, $\texttt{odd-and-leq}(\texttt{x}, \texttt{y}) = 1$ se sono entrambi dispari e $\texttt{x} \leq \texttt{y}$. Si possono utilizzare le funzioni definite nelle note.

Soluzione

1. $\texttt{odd}(0) = 0$
   $\texttt{odd}(\texttt{x} + 1) = \overline{sg}(\texttt{odd}(\texttt{x}))$
   $\texttt{odd-and}(\texttt{x}, \texttt{y}) = \texttt{odd}(\texttt{x}) \cdot \texttt{odd}(\texttt{y})$

2. $\texttt{odd-or}(\texttt{x}, \texttt{y}) = sg(\texttt{odd}(\texttt{x}) + \texttt{odd}(\texttt{y}))$

3. $\texttt{x} \leq \texttt{y} = \overline{sg}(\texttt{x} \dot{-} \texttt{y})$
   $\texttt{odd-and-leq}(\texttt{x}, \texttt{y}) = \texttt{odd-and}(\texttt{x}, \texttt{y}) \cdot (\texttt{x} \leq \texttt{y})$

Esercizio 4 Per ognuno dei seguenti insiemi di programmi (che calcolano funzioni da $\mathbb{N}$ in $\mathbb{N}$) si dica se è ricorsivamente enumerabile, motivando la risposta.

1. L'insieme dei programmi che su qualche numero pari terminano in (al più) 12 passi.

2. L'insieme dei programmi che su qualche input ≥ 12 terminano in un numero pari di passi.

3. L'insieme dei programmi che sull'input 12 non restituiscono un numero pari.

Soluzione

1. L'insieme è ricorsivamente enumerabile. Infatti, dato un programma, basta eseguirlo successivamente per 12 passi su tutti i numeri pari e se su qualche numero pari il programma termina otterremo risposta positiva.

2. L'insieme è ricorsivamente enumerabile. Infatti, dato un programma, utilizzando la tecnica a zig-zag sugli input ≥ 12, otterremo risposta positiva se questo su qualche input termina in un numero pari di passi.

3. L'insieme non è ricorsivamente enumerabile. Infatti, è non ricorsivo essendo estensionale e non banale per il teorema di Rice, ed il suo complementare è ricorsivamente enumerabile (bsta eseguire il programma sull'input 12), quindi per il teorema di Post non può esserlo.

Esercizio 5 Siano $\exists 5 = \{x \mid \phi_x(y) = 5 \text{ per qualche } y\}$, $\forall 5 = \{x \mid \phi_x(y) = 5 \text{ per ogni } y\}$, $\mathcal{T} = \{x \mid \phi_x \text{ totale}\}$.

1. Si mostri che $\mathcal{T} \leq \forall 5$.
2. Cosa possiamo concludere su $\forall 5$ come conseguenza del punto precedente?
3. Si dica se $\forall 5 \leq \exists 5$, motivando la risposta.

Soluzione

1. L'input del problema $\mathcal{T}$ è (la descrizione di) un algoritmo x, e dobbiamo trasformarlo in un nuovo algoritmo $g(x)$ in modo tale che ϕ_x sia totale se e solo se $\phi_{g(x)}$ è la funzione costante 5.

   È chiaro che questo si può ottenere costruendo l'algoritmo $g(x)$ nel modo seguente:

   > input $y \to$ invoco $\phi_x(y) \to$ se termina restituisco 5, altrimenti si ha non terminazione

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

   > se $x \in \mathcal{T}$, $\phi_{g(x)}(y) = 5$ per ogni y, quindi $g(x) \in \forall 5$.
   > se $x \notin \mathcal{T}$, $\phi_{g(x)}(y)$ non termina per qualche y, quindi $g(x) \notin \forall 5$.

2. Possiamo concludere che $\forall 5$ non è ricorsivamente enumerabile, in quanto se lo fosse lo sarebbe per riduzione anche $\mathcal{T}$.

3. No, perché $\exists 5$ è ricorsivamente enumerabile (infatti si può applicare la tecnica a zig-zag), quindi se fosse $\forall 5 \leq \exists 5$ lo sarebbe anche $\forall 5$.