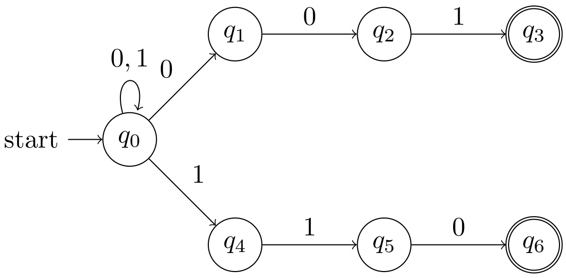
Parte I – Linguaggi Regolari e Linguaggi Liberi da Contesto

1. Dato il seguente NFA

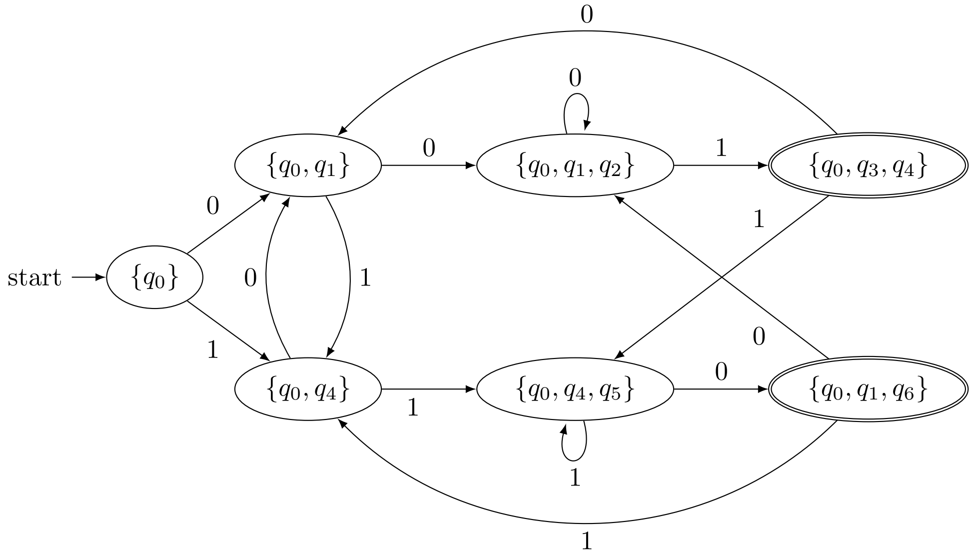


(a) descrivere in italiano il linguaggio riconosciuto dall'automa
(b) costruire un DFA equivalente

Soluzione:

(a) L'automa riconosce il linguaggio di tutte le parole sull'alfabeto $0, 1$ che terminano con 001 oppure con 110.

(b) Applicando la costruzione a sottoinsiemi si ottiene il DFA con il seguente diagramma di transizione (dove gli stati non raggiungibili da $\{q_0\}$ sono omessi):

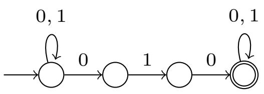


2. Il linguaggio

$$L = \left\{ u010v \mid u, v \in \{0, 1\}^* \right\}$$

è regolare? Motivare la risposta.

Soluzione: Il linguaggio è regolare perché è riconosciuto dall'espressione regolare $(\mathbf{0} + \mathbf{1})^* \mathbf{010} (\mathbf{0} + \mathbf{1})^*$ e dall'automa



Nota: il Pumping Lemma dice che se un linguaggio è regolare, allora rispetta certe proprietà. Quindi dà una condizione *necessaria* perché L sia regolare, ma non *sufficiente*: esistono linguaggi che rispettano il Pumping Lemma ma non sono regolari. Il fatto che lo schema di dimostrazione standard per mostrare che L non è regolare fallisca è una indicazione che L potrebbe essere regolare, ma non una dimostrazione formale. La dimostrazione corretta di regolarità di un linguaggio è quella che dà l'espressione regolare oppure l'automa che riconosce L.

3. Data la seguente grammatica libera da contesto

$$G : S \rightarrow BB$$
$$B \rightarrow 0B0 \mid 1B1 \mid 00 \mid 11$$

rispondere alle seguenti domande:

(a) Dare una definizione del linguaggio L(G) del tipo seguente: L(G) è l'insieme delle stringhe in $\{0, 1\}^*$ che soddisfano la seguente proprietà.....

(b) Dimostrare induttivamente che la vostra definizione di L(G) è corretta.

(c) Descrivere un automa a pila che riconosca L(G) e spiegare perché secondo voi funziona.

(d) Considerate ora la seguente grammatica

$$G' : S \rightarrow BB$$
$$B \rightarrow 0B1 \mid 1B0 \mid 01 \mid 10$$

Definite L(G') in modo simile a quanto fatto in (a) per L(G).

Soluzione:

(a) L(G) è l'insieme delle stringhe w in $\{0, 1\}^*$ tali che $w = w_1 w_2$ dove w_1 e w_2 sono palindromi di lunghezza almeno 2 e pari

(b) E' facile dimostrare che B genera palindromi di lunghezza almeno 2 e pari. Sono necessarie 2 dimostrazioni. La prima serve a dimostrare che se $w \in L(B)$, allora è palindromo di lunghezza almeno 2 e pari. La prova è per induzione sulla lunghezza della derivazione. Per lunghezza=1, le stringhe generabili in 1 passo sono 00 e 11 che soddisfano l'ipotesi. Per il passo induttivo, una derivazione di lunghezza n+1 inizia con $B \Rightarrow 0B0 \Rightarrow^n 0w'0$ (il caso $1B1$ è simile). Per ipotesi induttiva w' è palindromo di lunghezza almeno 2 e pari e quindi anche $0w'0$ lo è. La seconda dimostrazione serve a dimostrare che ogni stringa w che sia palindromo di lunghezza almeno 2 e pari venga generata da B, procede per induzione sulla lunghezza di w. Nella base, w ha lunghezza 2 e, visto che i soli palindromi di questa lunghezza sono 00 e 11, ovviamente B genera tali stringhe. Per stinghe di lunghezza $2 * n$ con $n > 1$, tali stringhe sono necessariamene o $0w'0$ o $1w'1$ con w' pari e di lunghezza pari. Ovviamente G ha le produzioni per produrre $0B0$ e $1B1$ e l'ipotesi induttiva garantisce che B è capace di generare w'. Per concludere la dimostrazione, basta osservare che $S \rightarrow BB$ fa si che le stringhe generabili da S abbiano la forma $w1w2$ desiderata.

(c) È possibile usare la costruzione vista in classe per produrre un automa che accetta per stack vuoto il linguaggio generato da una qualsiasi grammatica libera da contesto. Oppure è possibile definire un automa ad hoc. Vediamo questa seconda strada. Sappiamo già come fare un automa che accetta i palindromi. Dobbiamo solo fare in modo che vengano accettati sono palindromi di lunghezza almeno 2 e pari. Poi dovremo collegare tra loro 2 automi di questo tipo. L'automa finale è il seguente:

$$\begin{array}{ccc}
\delta(q_0, 0, Z) = \{(q_1, 0Z)\} & \delta(q_0, 1, Z) = \{(q_1, 1Z)\} & \delta(q_1, 0, ?) = \{(q_1, 0?)\} \\
\delta(q_1, 1, ?) = \{(q_1, 1?)\} & \delta(q_1, \epsilon, ?) = \{(q_2, ?)\} & \delta(q_2, 0, 0) = \{(q_2, \epsilon)\} \\
\delta(q_2, 1, 1) = \{(q_2, \epsilon)\} & \delta(q_2, \epsilon, Z) = \{(q'_0, Z)\} & \delta(q'_0, 0, Z) = \{(q'_1, 0Z)\} \\
\delta(q'_0, 1, Z) = \{(q'_1, 1Z)\} & \delta(q'_1, 0, ?) = \{(q'_1, 0?)\} & \delta(q'_1, 1, ?) = \{(q'_1, 1?)\} \\
\delta(q'_1, \epsilon, ?) = \{(q'_2, ?)\} & \delta(q'_2, 0, 0) = \{(q'_2, \epsilon)\} & \delta(q'_2, 1, 1) = \{(q'_2, \epsilon)\}
\end{array}$$

lo stato finale è q'_2. Aiuta la comprensione osservare il fatto che si tratta di 2 automi praticamente uguali in cascata. Ciascun automa consiste di tre stati, q_0, q_1 e q_2, (nel secondo automa gli stati sono q'_0, q'_1 e q'_2). q_0 garantisce che il palindromo sia di lunghezza almeno 2, poi q_1 carica l'input nello stack e q_2 matcha la seconda parte dell'input con lo stack.

(d) Data una stringa w in $\{0, 1\}^*$, con $\hat{w}$ indichiamo la stringa ottenuta da w scambiando ciascuno 0 con un 1 e ciascun 1 con uno 0. Insomma $\hat{w}$ è il complemento di w. L(G) contiene stringhe w_1, w_2 tali che sia w_1 che w_2 hanno la forma $q\hat{q}^R$, dove ^{R} indica l'inversa di una stringa e sono di lunghezza almeno 2 e pari.