Automi e Linguaggi Formali
Esame del 28 Giugno 2017 – Parte 1

Linguaggi Regolari

1. Scrivere una espressione regolare che definisca il linguaggio

$$L = \{w \in \{0,1\}^* \mid w \text{ contiene un numero pari di } 0\}$$

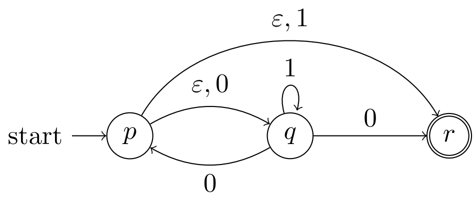
Soluzione. Per definire correttamente il linguaggio occorre tener presente che:

- le coppie di zeri non sono necessariamente vicine; parole come 010, 0110, 01 . . . 10 appartengono a L
- una parola che non contiene 0 è nel linguaggio; quindi 1, 11, . . . sono in L
- anche la parola vuota è nel linguaggio

Le soluzioni elencate qui sotto utilizzano la RE $\mathbf{01^*0}$ per generare coppie di zeri separate da un numero arbitrario (anche nullo) di 1, gestendo poi le parole che non contengono 0 in modo diverso:

$$(\mathbf{1} + \mathbf{01^*0})^* \qquad (\mathbf{1^*01^*0})^*\mathbf{1^*} \qquad \mathbf{1^*}(\mathbf{01^*0})^*\mathbf{1^*} \qquad (\mathbf{1^*01^*01^*})^* + \mathbf{1^*}$$

2. Dato il seguente ε-NFA

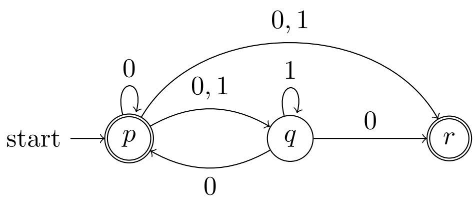


(a) calcolare la ε-chiusura di ogni stato
(b) costruire un DFA equivalente

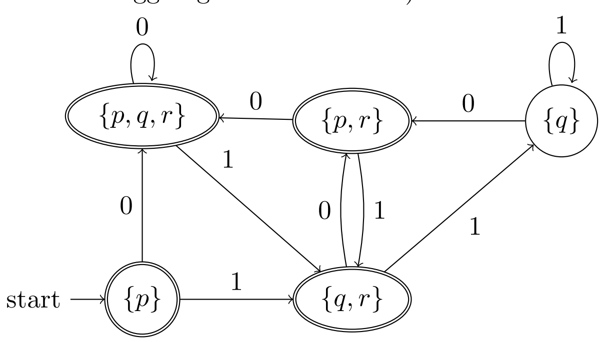
Soluzione.

(a) ECLOSE(p) $= \{p, q, r\}$ ECLOSE(q) $= \{q\}$ ECLOSE(r) $= \{r\}$

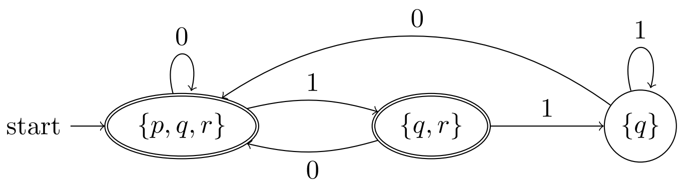
(b) Per costruire un DFA equivalente si può procedere in due modi. Il primo modo utilizza due passaggi: prima si eliminano le ε-transizioni, ottenendo il seguente NFA (dove p diventa uno stato finale)



e poi si usa la costruzione a sottoinsiemi per ottenere l'automa deterministico finale (dove gli stati non raggiungibili sono rimossi)



Il secondo modo è quello di utilizzare la trasformazione diretta da ε-NFA a DFA descritta nel libro di testo (non vista a lezione), che porta al seguente DFA



3. Il linguaggio

$$L = \{v00v \mid v \in \{0,1\}^*\}$$

è regolare? Motivare la risposta.

Soluzione. Il linguaggio contiene tutte le parole formate da due ripetizioni di una stringa v (di lunghezza arbitraria), separate dalla coppia 00. Intuitivamente, non può essere regolare perché per riconoscere le parole nel linguaggio occorre memorizzare tutti i simboli della prima occorrenza di v per poi controllare che si ripetano esattamente uguali dopo il separatore 00.

Per dimostrarlo formalmente, assumiamo per assurdo che L sia regolare:

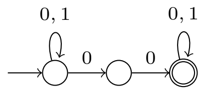
- sia $n > 0$ la lunghezza data dal Pumping Lemma;
- consideriamo la parola $w = 1^n 001^n$, che è di lunghezza maggiore di n ed è nella forma $v00v$ se poniamo $v = 1^n$. Quindi w appartiene ad L;
- sia $w = xyz$ una suddivisione arbitraria di w tale che $y \neq \varepsilon$ e $|xy| \leq n$;
- poiché $|xy| \leq n$, allora xy è completamente contenuta nella prima ripetizione di $v = 1^n$, e quindi sia x che y sono composte solo da 1. Inoltre, siccome $y \neq \varepsilon$, possiamo dire che $y = 1^p$ per qualche valore $p > 0$. Allora la parola xy^0z è nella forma $1^{n-p}001^n$, e non può appartere a L perché il numero di 1 che precedono il separatore 00 è minore del numero di 1 che lo seguono. Quindi non esiste nessuna parola u che ci consenta di scrivere xy^0z come $u00u$.

Abbiamo trovato un assurdo quindi L non può essere regolare.

Note:

(1) In alternativa si può utilizzare xy^2z (o un qualsiasi esponente $k > 1$): in questo caso si ottiene una parola che non appartiene ad L perché il numero di 1 prima del separatore è maggiore del numero di 1 dopo il separatore.

(2) Il ragionamento non funziona quando $v = 0^n$: in questo caso se l'avversario sceglie $y = 0^p$ con p pari allora è sempre possibile trovare un modo per scrivere xy^kz come $u00u$ (spostando la posizione del separatore 00). Quindi non è possibile utilizzare una v qualsiasi nella dimostrazione: occorre sceglie esplicitamente una parola ben precisa. Quella dove la dimostrazione è più semplice è $v = 1^n$.

(3) L'espressione regolare $(\mathbf{0}+\mathbf{1})^*\mathbf{00}(\mathbf{0}+\mathbf{1})^*$ e l'automa 

non riconoscono il linguaggio L perché accettano anche le parole $u00v$ con $u \neq v$.

Linguaggi Liberi dal Contesto

4. Data la seguente grammatica libera da contesto $G : S \to aS \mid Sb \mid a \mid b$, dimostrare, per induzione sulla lunghezza della stringa che nessuna stringa in $L(G)$ contiene "*ba*" come sottostringa. Descrivere $L(G)$ a parole.

Soluzione. Operiamo per induzione sulle stringhe w di L(G).

- Base: $|w| = 1$, le sole stringhe di lunghezza 1 di $L(G)$ sono "*a*" e "*b*" che non contengono "*ab*" ovviamente.
- Passo induttivo: assumiamo che tutte le stringhe w di $L(G)$ tali che $|w| = n \geq 1$ non contengano "*ab*". Consideriamo una stringa w' di lunghezza $n+1$. Visto che è in $L(G)$ esiste una derivazione di w' da S e questa derivazione deve iniziare con $S \Rightarrow aS$ o $S \Rightarrow bS$, dopo di che $S \Rightarrow aS \Rightarrow aw \Rightarrow w'$ oppure $S \Rightarrow Sb \Rightarrow wb \Rightarrow w'$. Visto che w ha lunghezza n, l'ipotesi induttiva dice che w non contiene "*ab*" e quindi neppure aw e wb possono contenere "*ab*".

Quindi nessuna stringa in $L(G)$ contiene "*ab*".

$L(G)$ contiene le stringhe non vuote che iniziano con una sequenza anche vuota di a, seguita da una sequenza anche vuota di b.

5. Dato l'automa a pila $P = (\{q\}, \{a, b\}, \{a, Z\}, \delta, q, Z, \{q\})$ dove δ è come segue:

$$\delta(q, a, Z) = \{(q, aZ)\}, \qquad \delta(q, a, a) = \{(q, aa)\}, \qquad \delta(q, b, a) = \{(q, \epsilon)\}.$$

Descrivere il linguaggio riconosciuto da P. Trasformare P in un PDA P' che accetta per pila vuota lo stesso linguaggio accettato da P per stato finale.

Soluzione. Il linguaggio $L(P)$ è costituito da tutte le stringhe di a e di b t.c. ogni loro prefisso ha un numero di a pari o superiore al numero di b.

Per trasformare P in P' t.c. $N(P') = L(P)$ seguiamo la costruzione generale.

- P' ha per lo stack i simboli $\{a, Z, X_0\}$ di cui X_0 è il fondo dello stack di P'.
- P' ha anche due stati q_0 e q' in più oltre a q di P ed ha le seguenti transizioni aggiuntive: $\delta(q_0, \epsilon, X_0) = \{(q, ZX_0)\}$, $\delta(q, \epsilon, ?) = \{(q', \epsilon)\}$, $\delta(q', \epsilon, ?) = \{(q', \epsilon)\}$, in cui ? sta per ogni simbolo dello stack.

Le spiegazioni sulla necessità (in generale) di X_0 e sulle transizioni aggiuntive le trovate sul libro di testo.

6. In generale gli automi a pila possono accettare per pila vuota o per stato finale. I linguaggi riconosciuti sono gli stessi. Per gli automi a pila deterministici questo non è più vero. Spiegate le ragioni di questa differenza. Cercate di specificare quali linguaggi vengono accettati nelle due modalità di accettazione. La differenza tra le 2 classi di linguaggi vi pare importante?

Soluzione. Nel caso degli automi a pila deterministici (DPDA), le modalità di accettazione fanno grande differenza. Quelli che accettano per stato finale, accettano tutti i linguaggi regolari e anche qualche linguaggio libero da contesto, ma non tutti. I DPDA che accettano per pila vuota accettano i linguaggi accettati dai DPDA che accettano per stato finale che hanno la proprietà del prefisso, cioè quei linguaggi tali che non contengano 2 stringhe di cui una sia prefisso dell'altra.

Questo fatto implica che i DPDA che accettano per pila vuota non accettano neppure tutti i linguaggi regolari, visto che il linguaggio a^* è regolare e non ha la proprietà del prefisso. D'altra parte è facile far si che un linguaggio abbia la proprietà del prefisso: basta aggiungere ad ogni stringa del linguaggio un simbolo speciale (unico) che marchi la fine della stringa. Per esempio la stringa w diventerebbe $w\$$. E' ovvio che in questo modo nessuna stringa del linguaggio può essere prefisso di un'altra.

È importante che i DPDA accettano solo linguaggi liberi da contesto non inerentemente ambigui, ma non tutti questi linguaggi.