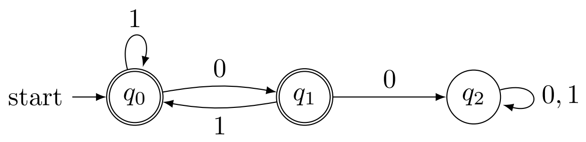
Automi e Linguaggi Formali
Esame del 3 Luglio 2019 – Parte I

Tempo a disposizione: 1 h 30 min

1. Considerate il DFA che riconosce il linguaggio di tutte le stringhe sull'alfabeto $\{0, 1\}$ che *non contengono* la sottostringa 00:



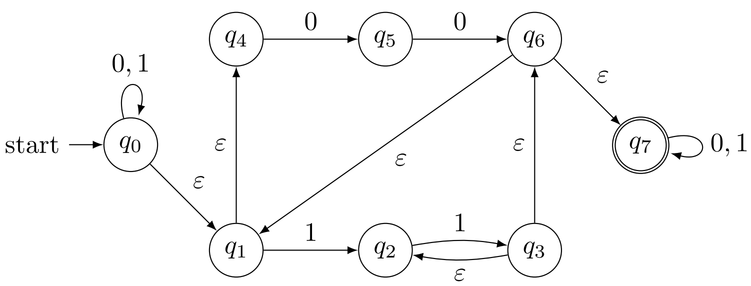
Trasformate il DFA in una Espressione Regolare usando l'algoritmo di eliminazione degli stati.

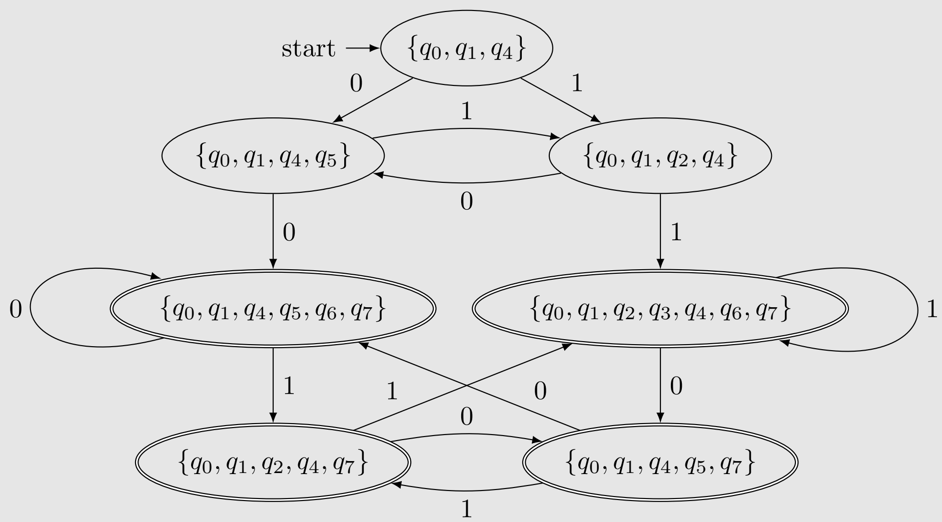
La soluzione ottenuta con l'algoritmo di eliminazione degli stati è $(1+01)^* + (1+01)^*0$, che può essere riscritta in forma più compatta come $(1+01)^*(0+\varepsilon)$

2. Usate la soluzione dell'esercizio precedente per creare una Espressione Regolare che definisca il linguaggio di tutte le stringhe sull'alfabeto $\{0, 1\}$ tali che tutte le occorrenze di 11 appaiono prima di tutte le occorrenze di 00.

La soluzione di questo esercizio si basa sull'osservazione che ogni parola in cui le occorrenze di 11 appaiono prima di tutte le occorrenze di 00 può essere suddivisa in due parti: una prima parte in cui non compare mai 00 (e può comparire 11) seguita da una seconda parte in cui non compare mai 11 (e può comparire 00). Di conseguenza, l'espressione regolare ottenuta nell'esercizio 1 può essere usata per descrivere la prima parte della stringa, mentre la sua versione "duale" in cui si scambiano 0 e 1 si può usare per la seconda parte della parola: $(1+01)^*(0+\varepsilon)(0+10)^*(1+\varepsilon)$, che si può semplificare in $(1+01)^*(0+10)^*(1+\varepsilon)$.

3. Trasformate il seguente ε-NFA in DFA:





4. Sia $\Sigma = \{0, 1\}$ e considerate il linguaggio

$$LMN = \{0^\ell 1^m 0^n \mid \ell < n\}$$

(a) Completate il seguente schema di partita per il Gioco del Pumping Lemma in modo da far vincere il Giocatore 2:

> **Giocatore 1:** sceglie il valore di $h = \mathbf{4}$
> **Giocatore 2:** sceglie la parola $w \in LMN$ di lunghezza maggiore di h
>
> $$w = \mathbf{0000011000000}$$
>
> **Giocatore 1:** suddivide w in
> - $x = \mathbf{0}$
> - $y = \mathbf{00}$
> - $z = \mathbf{0011000000}$
>
> rispettando le condizioni che $|xy| \le h$ e $y \neq \varepsilon$
> **Giocatore 2:** sceglie una potenza $k = \mathbf{2}$
> La parola $xy^k z = \mathbf{000000011000000} \notin LMN$: vince il **Giocatore 2**

(b) Dimostrate che LMN non è un linguaggio regolare usando il Pumping Lemma.

> Supponiamo per assurdo che LMN sia regolare. Di conseguenza, LMN deve rispettare il Pumping Lemma.
> - sia h la lunghezza data dal Pumping Lemma;
> - consideriamo la parola $w = 0^h 1 0^{h+1}$, che appartiene ad LMN ed è di lunghezza maggiore di h;
> - sia $w = xyz$ una suddivisione arbitraria di w tale che $y \neq \varepsilon$ e $|xy| \le h$;
> - poiché $|xy| \le h$, allora xy è completamente contenuta nel prefisso 0^h di w posto prima dell'1 di separazione, e quindi sia x che y sono composte solo da 0. Inoltre, siccome $y \neq \varepsilon$, possiamo dire che $y = 0^p$ per qualche valore $p > 0$. Allora la parola xy^2z è nella forma $0^{h+1}10^{h+1}$, e non appartiene al linguaggio perché la lunghezza della prima sequenza di 0 è uguale alla lunghezza della seconda sequenza di 0.
>
> Abbiamo trovato un assurdo: LMN non è un linguaggio regolare.

5. Si consideri la seguente grammatica CF, G: $\quad S \to aBb\ ,\ B \to aBb \mid BB \mid \varepsilon$

(a) Descrivere il linguaggio $L(G)$ in termini di un linguaggio L composto da "stringhe w con certe proprietà".

> Se sostituiamo a con (e b con) è facile vedere che $L(B)$contiene tutte le stringhe con parentesi bilanciate. Formalmente $w \in L(B)$ sse $w = \varepsilon$ oppure $w = (w')w''$ con w' e $w'' \in L(B)$. $S \to aBb$ semplicemente aggiunge una a all'inizio ed una b alla fine delle stringhe generate da B. In questo modo $L(S)$ ha la proprietà del prefisso.

(b) Dimostrare per induzione che tutte le stringhe in $L(G)$ sono in L, cioè che $L(G) \subseteq L$.

> Dimostriamo innanzitutto la seguente tesi: $L(B)$ contiene solo stringhe ben parentesizzate (se sostituiamo a con (e b con)). Usiamo un'induzione sulla lunghezza della derivazione.
> Base. Lunghezza 1: $B \Rightarrow \epsilon$ ed ϵ è ben parentesizzata.
> Step. Supponiamo che la tesi sia vera per derivazioni di lunghezza minore o uguale a $n \ge 1$ e dimostriamo che allora vale per derivazioni di lunghezza $n+1$. Sia $B \Rightarrow^{n+1} w$, allora il primo passo della derivazione può essere o $B \to (B)$ oppure $B \to BB$. Consideriamo il primo caso: $B \Rightarrow (B) \Rightarrow^n (w')$ e quindi $B \Rightarrow^n w'$, e allora, per ipotesi induttiva, w' è per parentesizzata e quindi (w') è ben parentesizzata. Consideriamo l'altro caso: $B \Rightarrow BB \Rightarrow^n w'w''$, ma allora $B \Rightarrow^k w'$ e $B \Rightarrow^s w''$ con $k + s = n$ ed entrambe maggiori di 0. Quindi, k ed s sono minori di n per cui, per ipotesi induttiva, w' e w'' sono ben parentesizzate, per cui anche la loro concatenazione è ben parentesizzata.