Fondamenti dell'informatica Modulo 1

Raccolta di prove d'esame

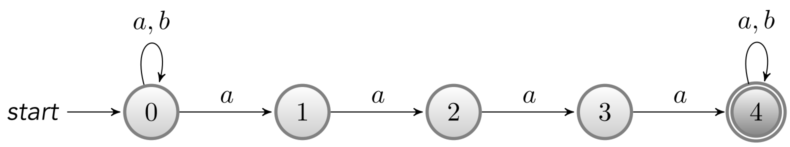
Prof. Giorgio Gambosi

Quesito: Sia dato il linguaggio $L = \{\sigma \in \{a, b, c\}^* \mid \#a(\sigma) = \#b(\sigma) = \#c(\sigma)\}$, dove $\#x(\sigma)$ indica il numero di occorrenze del carattere x nella stringa σ. Il linguaggio L è context free? Dimostrare la risposta data.

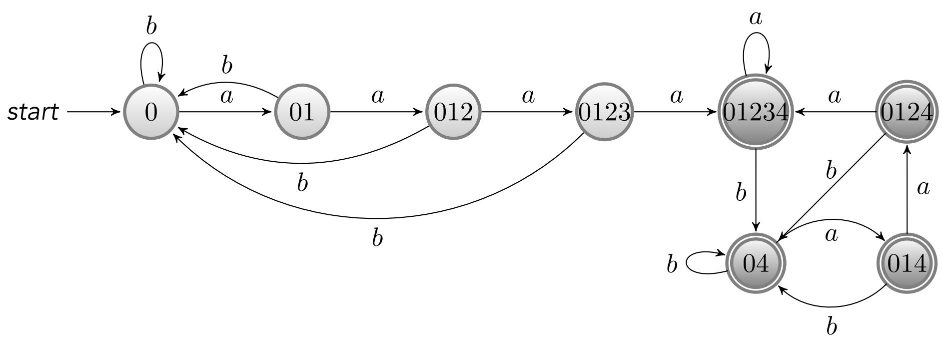
Soluzione: *Il linguaggio non è context free. Per dimostrarlo, utilizziamo il pumping lemma nel modo seguente. Fissato $n > 0$, consideriamo la stringa $\sigma = a^n b^n c^n$. Qualsiasi decomposizione $\sigma = uvwxy$ con $|vwx| \leq n$ e $|vx| \geq 1$ avrà necessariamente che o che vwx è una stringa con tutti simboli uguali (tutti a, tutti b o tutti c), o che vwx è una stringa comprendente due soli tipi di caratteri (del tipo $a^p b^q$ o $b^r c^s$). In entrambi i casi c'è almeno un carattere dell'alfabeto $\{a, b, b\}$ che non compare in vwx , e quindi in v e x. Ne deriva che, considerando la stringa uv^2wx^2y il numero di occorrenze aumentano per almeno uno e al più due caratteri dell'alfabeto, per cui uv^2wx^2y non presenta lo stesso numero di occorrenze di a, b, c.*

Quesito: Definire un ASFD minimo che riconosca il linguaggio $L \subset \{a, b\}^*$ comprendente tutte le stringhe che non contengono sequenze di più di tre a al loro interno.

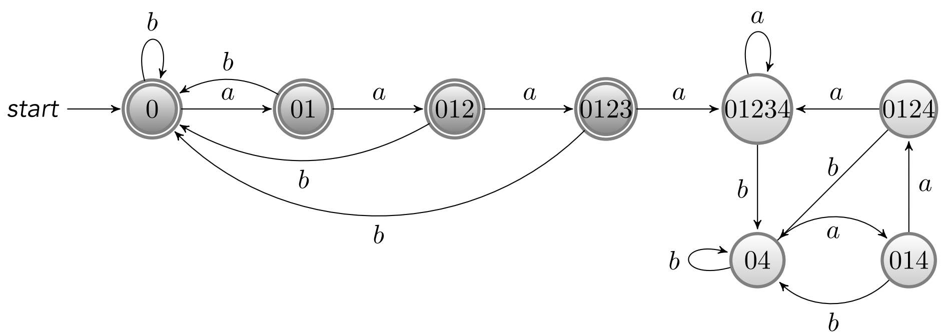
Soluzione: *Definiamo un ASFND che accetta $\overline{L}$.*



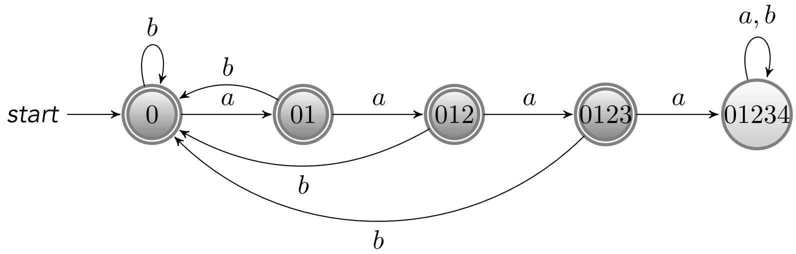
e da questo un ASFD che riconosce lo stesso linguaggio



L'ASFD che riconosce L deriva immediatamente



La minimizzazione dell'automa ci fornisce le classi di equivalenza $\{0\}, \{01\}, \{012\}, \{0123\}, \{04, 014, 0124, 01234\}$, *da cui deriva l'automa minimo*



Quesito: Si definisca una grammatica context free che generi il linguaggio $L = \{a^r b^s c^t a^n c^n | s = r + t, r, t, n \geq 0\}$.

Soluzione: *Una possibile soluzione è la grammatica*

$$
\begin{array}{rcl}
S & \rightarrow & ABC \\
A & \rightarrow & aAb \mid \varepsilon \\
B & \rightarrow & bBc \mid \varepsilon \\
C & \rightarrow & aCc \mid \varepsilon
\end{array}
$$

Quesito: Si definisca un automa a pila (eventualmente non deterministico) che accetti il linguaggio $L = \{a^r b^s c^t a^n c^n | s = r + t, r, t, n \geq 0\}$.

Soluzione: *L'automa si puà² derivare dalla grammatica dell'esercizio precedente, portandola prima in forma ridotta*

$$
\begin{array}{rcl}
S & \rightarrow & ABC \mid BC \mid AB \mid AC \mid \varepsilon \\
A & \rightarrow & aAb \mid ab \\
B & \rightarrow & bBc \mid bc \\
C & \rightarrow & aCc \mid ac
\end{array}
$$

quindi in CNF

$$
\begin{array}{rcl}
S & \rightarrow & TC \mid BC \mid AB \mid AC \mid \varepsilon \\
A & \rightarrow & UY \mid XY \\
B & \rightarrow & VZ \mid YZ \\
C & \rightarrow & WZ \mid XZ \\
T & \rightarrow & AB \\
U & \rightarrow & XA \\
V & \rightarrow & YB \\
W & \rightarrow & XC \\
X & \rightarrow & a \\
Y & \rightarrow & b \\
Z & \rightarrow & c
\end{array}
$$

e in GNF (i non terminali T, U, V, W *risultano inutili nella grammatica in GNF in quanto non raggiungibili)*

$$
\begin{array}{rcl}
S & \to & aAYBC \mid aYBC \mid bBZC \mid bZC \mid aAYB \mid aYB \mid aAYC \mid aYC \mid \varepsilon \\
A & \to & aAY \mid aY \\
B & \to & bBZ \mid bZ \\
X & \to & a \\
Y & \to & b \\
Z & \to & c
\end{array}
$$

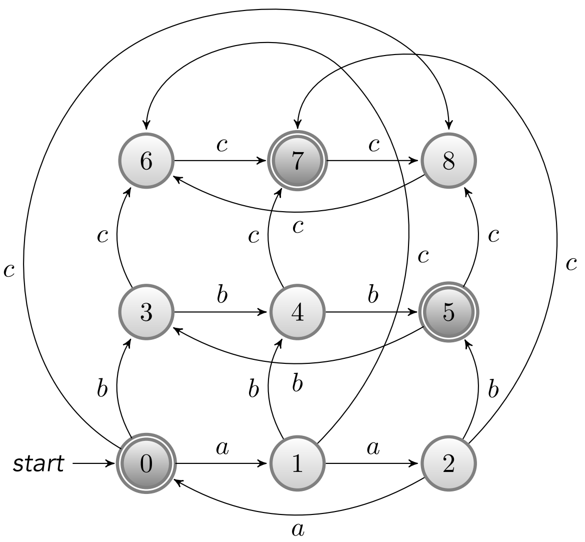
La funzione dei transizione del PDA non deterministico risulta allora:

$$
\begin{array}{l}
\delta(q_0, \varepsilon, S)\{(q_0, \varepsilon)\} \\
\delta(q_0, a, S) = \{(q_0, AYBC), (q_0, YBC), (q_0, AYB), (q_0, YB), (q_0, AYC), (q_0, YC)\} \\
\delta(q_0, b, S) = \{(q_0, BZC), (q_0, ZC)\} \\
\delta(q_0, a, A) = \{(q_0, AY), (q_0, Y)\} \\
\delta(q_0, b, B) = \{(q_0, BZ), (q_0, Z)\} \\
\delta(q_0, a, C) = \{(q_0, CZ), (q_0, Z)\} \\
\delta(q_0, a, X) = \{(q_0, \varepsilon)\} \\
\delta(q_0, b, Y) = \{(q_0, \varepsilon)\} \\
\delta(q_0, c, Z) = \{(q_0, \varepsilon)\}
\end{array}
$$

Quesito: Si definisca una grammatica di tipo 3 che generi il seguente linguaggio

$$L = \{a^n b^m c^k | n + m + k \text{ divisibile per } 3\}$$

Soluzione: *Definiamo un ASFD che riconosce il linguaggio*



da cui deriva immediatamente la grammatica di tipo 3

$$
\begin{array}{rcl}
A_0 & \to & aA_1 \mid bA_3 \mid cA_8 \\
A_1 & \to & aA_2 \mid bA_4 \mid cA_6 \\
A_2 & \to & aA_0 \mid bA_5 \mid cA_7 \mid a \\
A_3 & \to & bA_4 \mid cA_6 \\
A_4 & \to & bA_5 \mid cA_7 \mid b \mid c \\
A_5 & \to & bA_3 \mid cA_8 \\
A_6 & \to & cA_7 \mid c \\
A_7 & \to & cA_8 \\
A_8 & \to & cA_6
\end{array}
$$

Quesito:Definire un automa a stati finiti deterministico che riconosce il linguaggio $L \subset \{0,1\}^*$ composto da tutte le stringhe che non contengono la sequenza 111.

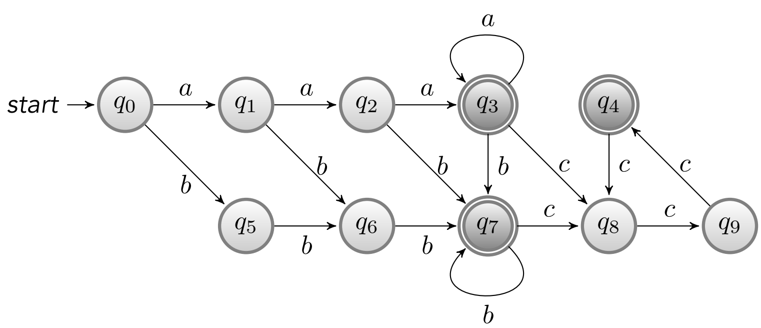
Soluzione:

Quesito: Si consideri il linguaggio

$$L = \{a^i b^j c^k | i + j \geq 3, k \text{mod } 3 = 0\}$$

Il linguaggio è regolare o context free? Dimostrare quale delle due affermazioni è vera. Si definisca inoltre una grammatica (di tipo 3 o di tipo 2, rispettivamente) che generi tutte e sole le stringhe del linguaggio.

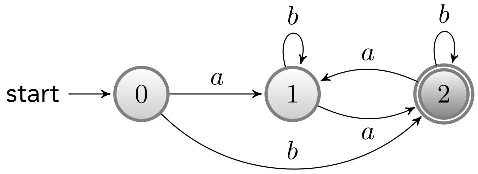
Soluzione: *Il linguaggio è regolare. Per dimostrare ciò, mostriamo un ASFD che lo riconosce.*



La grammatica corrispondente sarà

$$
\begin{array}{rcl}
S & \to & aA_1|bA_5 \\
A_1 & \to & aA_2|bA_6 \\
A_2 & \to & aA_3|bA_7|a|b \\
A_3 & \to & aA_3|bA_7|cA_8|a \\
A_4 & \to & cA_8 \\
A_5 & \to & bA_6 \\
A_6 & \to & bA_7|b \\
A_7 & \to & bA_7|cA_8|b \\
A_8 & \to & cA_9 \\
A_9 & \to & cA_4|c
\end{array}
$$

Quesito: Sia dato l'ASFD seguente



Si mostri come sia possibile ricavare una espressione regolare che descriva il linguaggio riconosciuto dall'automa.

Soluzione:

Quesito: Si definisca una automa a pila che accetta il linguaggio

$$L = \{\heartsuit^n \diamondsuit^{2n} | n > 0\}$$

Soluzione:

Quesito: Sia dato l'ASFD definito come $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$, con

1. $\Sigma = \{a, b\}$
2. $Q = \{1, 2, 3, 4, 5, 6, 7, 8\}$
3. $q_0 = 1$
4. $F = \{2, 4\}$

e funzione di transizione δ:

	a	b
1	3	8
2	3	1
3	8	2
4	5	6
5	6	2
6	7	8
7	6	4
8	5	8

Derivare la grammatica più semplice (con meno simboli) che genera $L(\mathcal{A})$.

Soluzione: *Applicando l'algoritmo di derivazione dell'automa minimo risulta* $1 \equiv 6 \equiv 8$, $2 \equiv 4$ *e* $3 \equiv 5 \equiv 7$.

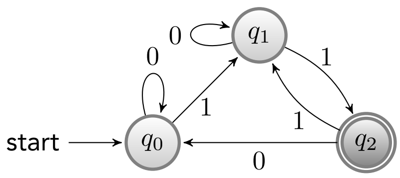
Mantenendo gli stati $1, 2, 3$ *come rappresentanti delle classi di equivalenza, risulta l'automa minimo con stato finale* 2 *e funzione di transizione:*

	a	*b*
1	*3*	*1*
2	*3*	*1*
3	*1*	*2*

Da cui la grammatica, con $S = A_1$,

$$\begin{array}{lcl} A_1 & \rightarrow & aA_3|bA_1 \\ A_2 & \rightarrow & aA_3|bA_1 \\ A_3 & \rightarrow & aA_1|bA_2|b \end{array}$$

Quesito: Sia L il linguaggio riconosciuto dal seguente ASFD,



derivare una espressione regolare che descriva L.

Soluzione: *Una possibile soluzione prevede la derivazione della grammatica regolare equivalente*

$$
\begin{array}{lcl}
A_0 & \to & 0A_0|1A_1 \\
A_1 & \to & 0A_1|1A_2|1 \\
A_2 & \to & 0A_0|1A_1
\end{array}
$$

E da questa, manipolando il sistema di espressioni corrispondente, l'espressione regolare cercata.

$$
\begin{cases}
A_0 = 0A_0 + 1A_1 \\
A_1 = 0A_1 + 1A_2 + 1 \\
A_2 = 0A_0 + 1A_1
\end{cases}
$$

$$
\begin{cases}
A_0 = 0A_0 + 1A_1 \\
A_1 = 0A_1 + 1A_0 + 1 \\
A_2 = A_0
\end{cases}
$$

$$
\begin{cases}
A_0 = 0^*1A_1 \\
A_1 = 0A_1 + 10^*1A_1 + 1 \\
A_2 = A_0
\end{cases}
$$

$$
\begin{cases}
A_0 = 0^*1A_1 \\
A_1 = (0 + 10^*1)A_1 + 1 \\
A_2 = A_0
\end{cases}
$$

$$
\begin{cases}
A_0 = 0^*1A_1 \\
A_1 = (0 + 10^*1)^*1 \\
A_2 = A_0
\end{cases}
$$

$$
\begin{cases}
A_0 = 0^*1(0 + 10^*1)^*1 \\
A_1 = (0 + 10^*1)^*1 \\
A_2 = A_0
\end{cases}
$$

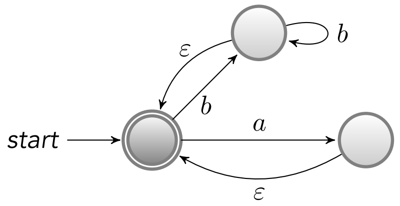
L è descritto dall'espressione associata all'assioma, e quindi da $0^*1(0 + 10^*1)^*1$.

Quesito: Si consideri il linguaggio $L = \{a^h b^k | k > h > 0\}$. Dimostrare se L è regolare o meno.

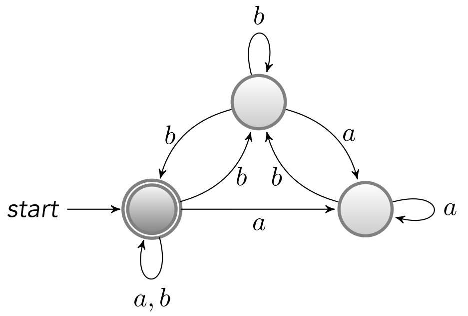
Soluzione: *Il linguaggio non è regolare. Per dimostrarlo, utilizziamo il pumping lemma nel modo seguente. Fissato $n > 0$, consideriamo la stringa $\sigma = a^n b^{n+1}$. Qualsiasi decomposizione $\sigma = uvw$ con $|uv| \leq n$ e $|v| \geq 1$ avrà necessariamente $uv = a^r$ e $w = a^{n-r} b^{n+1}$ con $r \leq n$, e quindi $v = a^s$, $u = a^{r-s}$ per un qualche valore $0 < s \leq r$. Scegliendo ad esempio $i = 2$ abbiamo allora che $uv^2w = a^{r-s} v^s v^s u^{n-r} b^{n+1} = a^{n+s} b^{n+1} \notin L$.*

Quesito: Si consideri l'espressione regolare $r = a(bb^* + a)^*ab$. Derivare un ASFD che riconosce $L(r)$.

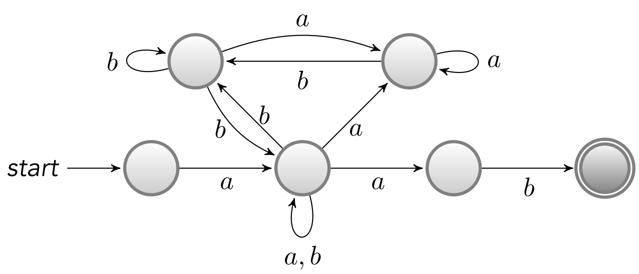
Soluzione: *Deriviamo da r un ASFND con ε-transizioni che riconosca $L(r)$. Possiamo osservare che la sotto-espressione regolare $(bb^* + a)^*$ è accettata per costruzione dall'ASFND con ε-transizioni*



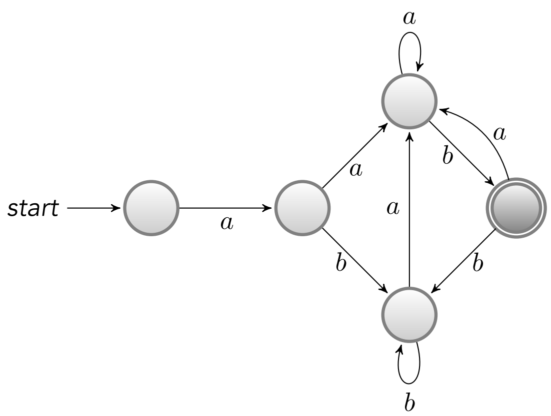
Eliminando le ε-transizioni, si ottiene l'ASFND



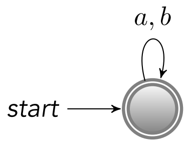
Da cui immediatamente l'ASFND per $L(r)$



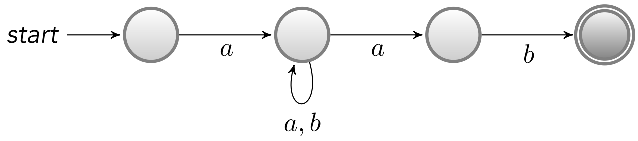
e da questo l'ASFD



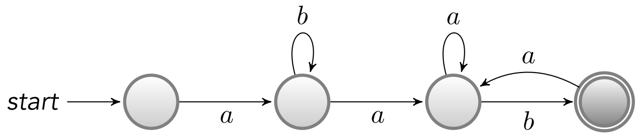
In alternativa, si potrebbe osservare che $(bb^* + a)^*$ comprende tutte le stringhe sull'alfabeto $\{a, b\}$, che sono riconosciute da



Da cui l'ASFND per $L(r)$



e da questo l'ASFD



Quesito: Definire una grammatica CF che generi il linguaggio $L = \{w \in \{a, b\} | w$ contiene almeno $4b\}$

Soluzione: *Osserviamo che possiamo risolvere il problema derivando una grammatica regolare che generi L. A tal fine, definiamo un ASFD che riconosca L.*

	a	b
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_3	q_4
q_4	q_4	q_4

con $F = \{q_4\}$.
La grammatica deriva immediatamente come

$$
\begin{array}{lcl}
A_0 & \rightarrow & aA_0 \mid bA_1 \\
A_1 & \rightarrow & aA_1 \mid bA_2 \\
A_2 & \rightarrow & aA_2 \mid bA_3 \\
A_3 & \rightarrow & aA_3 \mid bA_4 \mid b \\
A_4 & \rightarrow & aA_4 \mid bA_4 \mid a \mid b
\end{array}
$$

Quesito: Ridurre la grammatica seguente in GNF

$$
\begin{array}{lcl}
S & \rightarrow & aEb \mid aaC \mid AA \\
A & \rightarrow & BC \mid bS \mid b \\
B & \rightarrow & aB \mid \varepsilon \\
C & \rightarrow & Ca \mid Cb \\
D & \rightarrow & a \mid c
\end{array}
$$

Soluzione: *Per portare la grammatica in forma ridotta eliminiamo l'ε-produzione, ottenendo*

$$\begin{array}{rcl} S & \rightarrow & aEb \mid aaC \mid AA \\ A & \rightarrow & BC \mid C \mid bS \mid b \\ B & \rightarrow & aB \mid a \\ C & \rightarrow & Ca \mid Cb \\ D & \rightarrow & a \mid c \end{array}$$

Eliminiamo quindi la produzione unitaria

$$\begin{array}{rcl} S & \rightarrow & aEb \mid aaC \mid AA \\ A & \rightarrow & BC \mid Ca \mid Cb \mid bS \mid b \\ B & \rightarrow & aB \mid a \\ C & \rightarrow & Ca \mid Cb \\ D & \rightarrow & a \mid c \end{array}$$

Osserviamo ora che C e E sono simboli non fecondi, per cui eliminandoli otteniamo

$$\begin{array}{rcl} S & \rightarrow & AA \\ A & \rightarrow & bS \mid b \\ B & \rightarrow & aB \mid a \\ D & \rightarrow & a \mid c \end{array}$$

a questo punto, eliminando i simboli non raggiungibili B e D, otteniamo la grammatica equivalente in forma ridotta

$$\begin{array}{rcl} S & \rightarrow & AA \\ A & \rightarrow & bS \mid b \end{array}$$

La corrispondente grammatica in CNF è

$$\begin{array}{rcl} S & \rightarrow & AA \\ A & \rightarrow & BS \mid b \\ B & \rightarrow & b \end{array}$$

e da questa la grammatica in GNF

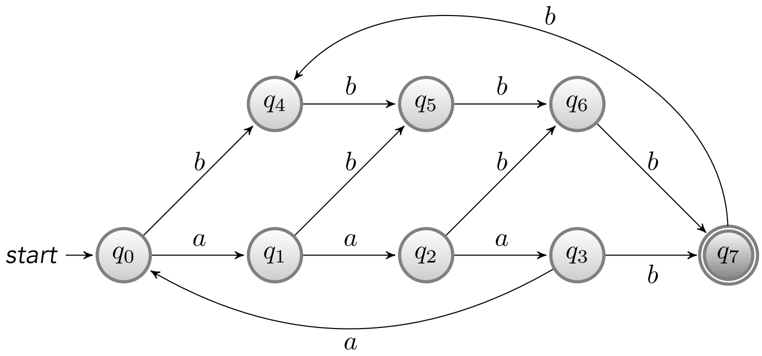
$$\begin{array}{rcl} S & \rightarrow & bSA \mid bA \\ A & \rightarrow & bS \mid b \\ B & \rightarrow & b \end{array}$$

Quesito: Si consideri il linguaggio

$$L = \{w \in \{a, b\}^* | w = a^n b^m, n + m \text{ multiplo di 4}, m \geq 1\}$$

Si definiscano un ASFD che riconosce L e una grammatica regolare che lo genera.

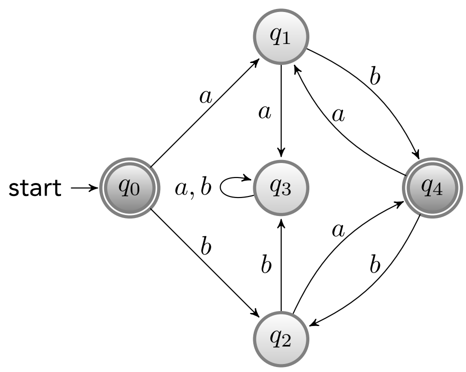
Soluzione: *Possibile soluzione*



La grammatica regolare deriva applicando la trasformazione nota, risultando:

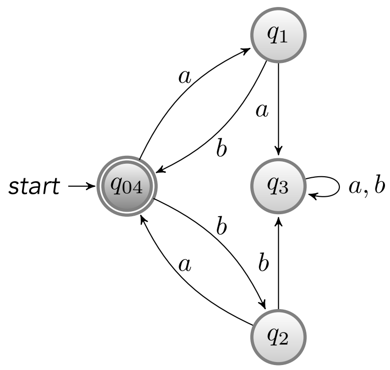
$$
\begin{array}{rcl}
S & \rightarrow & aA_1|bA_4 \\
A_1 & \rightarrow & aA_2|bA_5 \\
A_2 & \rightarrow & aA_3|bA_6 \\
A_3 & \rightarrow & bA_7|aS|b \\
A_4 & \rightarrow & bA_5 \\
A_5 & \rightarrow & bA_6 \\
A_6 & \rightarrow & bA_7|b \\
A_7 & \rightarrow & bA_4
\end{array}
$$

Quesito: Dato l'ASFD seguente



si derivi una ASFD minimo equivalente.

Soluzione: *L'applicazione del metodo studiato indica che i soli stati indistinguibili sono* q_0 *e* q_4. *Ne deriva l'automa minimo seguente*



Quesito: Si dimostri che il linguaggio

$$L = \{a^* b^k c^* a^k b^* | k \geq 4\}$$

non è regolare

Soluzione: *Utilizziamo il pumping lemma per i linguaggi regolari. Dato l'intero n, consideriamo la stringa $b^{n+4}a^{n+4} \in L$: per ogni decomposizione uvw di $a^{n+4}b^{n+4}$ tale che $|uv| \leq n$, $|v| > 0$ si ha che $uv = b^m$, $m \leq n$, e quindi $v = b^r$, $r > 0$. Ne deriva che la stringa $uv^2w = b^{n+r+4}a^{n+4} \notin L$.*

Quesito:Definire un automa a stati finiti deterministico che riconosce il linguaggio $L \subset \{0,1\}^*$ composto da tutte le stringhe che non contengono la sequenza 111.

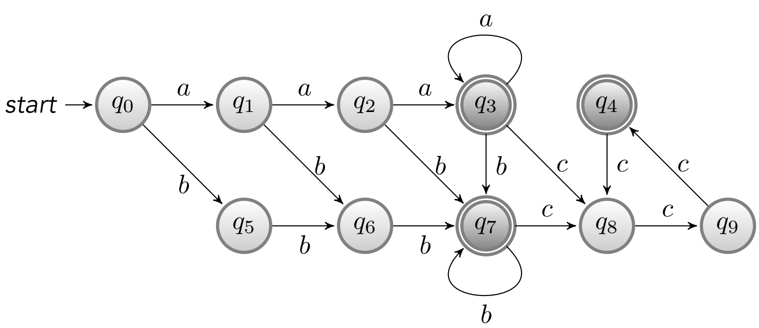
Soluzione:

Quesito: Si consideri il linguaggio

$$L = \{a^i b^j c^k | i + j \geq 3, k \text{mod } 3 = 0\}$$

Il linguaggio è regolare o context free? Dimostrare quale delle due affermazioni è vera. Si definisca inoltre una grammatica (di tipo 3 o di tipo 2, rispettivamente) che generi tutte e sole le stringhe del linguaggio.

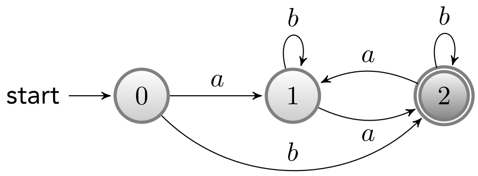
Soluzione: *Il linguaggio è regolare. Per dimostrare ciò, mostriamo un ASFD che lo riconosce.*



La grammatica corrispondente sarà

$$
\begin{array}{rcl}
S & \rightarrow & aA_1|bA_5 \\
A_1 & \rightarrow & aA_2|bA_6 \\
A_2 & \rightarrow & aA_3|bA_7|a|b \\
A_3 & \rightarrow & aA_3|bA_7|cA_8|a \\
A_4 & \rightarrow & cA_8 \\
A_5 & \rightarrow & bA_6 \\
A_6 & \rightarrow & bA_7|b \\
A_7 & \rightarrow & bA_7|cA_8|b \\
A_8 & \rightarrow & cA_9 \\
A_9 & \rightarrow & cA_4|c
\end{array}
$$

Quesito: Sia dato l'ASFD seguente



Si mostri come sia possibile ricavare una espressione regolare che descriva il linguaggio riconosciuto dall'automa.

Soluzione:

Quesito: Sia dato l'ASFD definito come $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$, con

1. $\Sigma = \{a, b\}$
2. $Q = \{1, 2, 3, 4, 5, 6, 7, 8\}$
3. $q_0 = 1$
4. $F = \{2, 4\}$

e funzione di transizione δ:

	a	b
1	3	8
2	3	1
3	8	2
4	5	6
5	6	2
6	7	8
7	6	4
8	5	8

Derivare la grammatica più semplice (con meno simboli) che genera $L(\mathcal{A})$.

Soluzione: *Applicando l'algoritmo di derivazione dell'automa minimo risulta* $1 \equiv 6 \equiv 8$, $2 \equiv 4$ *e* $3 \equiv 5 \equiv 7$.

Mantenendo gli stati $1, 2, 3$ *come rappresentanti delle classi di equivalenza, risulta l'automa minimo con stato finale* 2 *e funzione di transizione:*

	a	b
1	3	1
2	3	1
3	1	2

Da cui la grammatica, con $S = A_1$*,*

$$
\begin{array}{lcl}
A_1 & \rightarrow & aA_3|bA_1 \\
A_2 & \rightarrow & aA_3|bA_1 \\
A_3 & \rightarrow & aA_1|bA_2|b
\end{array}
$$

Quesito: Sia dato il linguaggio

$$L = \{w \in \{a, b\}^*\}$$

tale che:

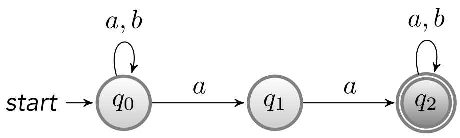
1. w non contiene la stringa aa
2. nessun carattere b in w compare "isolato", vale a dire senza almeno un altro b adiacente (che lo precede o lo segua)
3. $|w|$ è pari

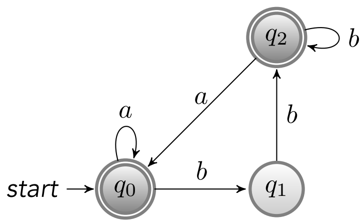
Dimostrare che L è regolare.

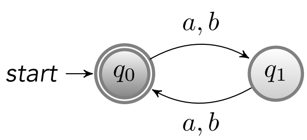
Soluzione: *Possiamo considerare i tre linguaggi:*

- $L_1 = \{w$ *contiene* $aa\}$
- $L_2 = \{w$ *non compaiono* b *isolati*$\}$
- $L_3 = \{w : |w|$ *pari*$\}$

I tre linguaggi sono regolari, in quanto, ad esempio, accettati rispettivamente dagli ASF seguenti







L risulta regolare, per le proprietà di chiusura dei linguaggi regolari, in quanto

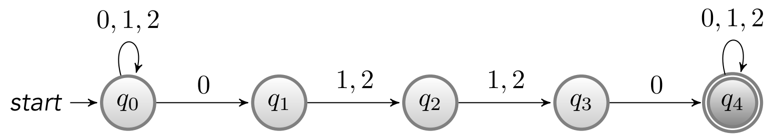
$$L = \overline{L}_1 \cap L_2 \cap L_3$$

Quesito: Si consideri il linguaggio

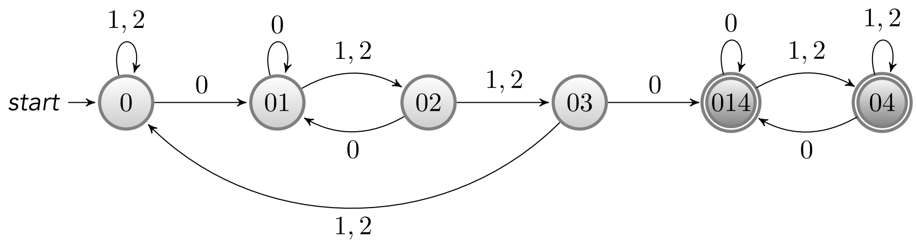
$$L = \{w \in \{0,1,2\}^* | w \text{ contiene una sottostringa } 0x0, \text{ con } x \in \{1,2\}^2\}$$

Si definisca un ASFD minimo che riconosce L

Soluzione: *ASFND che accetta L:*



da cui l'ASFD



L'automa minimo deriva osservando che i soli stati indistinguibili sono 014 e 04, che possono quindi essere unificati.

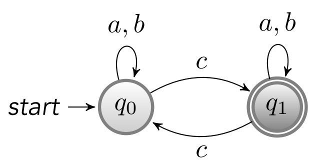
Quesito: Si costruisca un automa (deterministico o non deterministico) che riconosca il linguaggio $L \subseteq \{a,b,c\}^*$ definito come segue

$L = \{w | w$ contiene un numero dispari di c oppure non contiene occorrenze della sottostringa $aba\}$

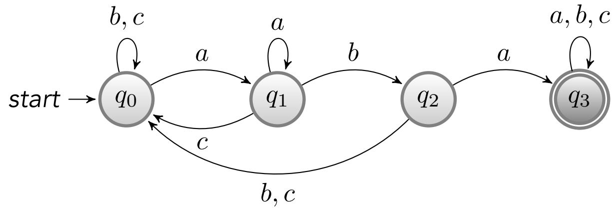
Soluzione: *Si osservi che possiamo scrivere $L = L_1 \cup \overline{L}_2$, con*

$$L_1 = \{w | w \textit{ contiene un numero dispari di } c\}$$
$$L_2 = \{w | w \textit{ contiene la sottostringa } aba\}$$

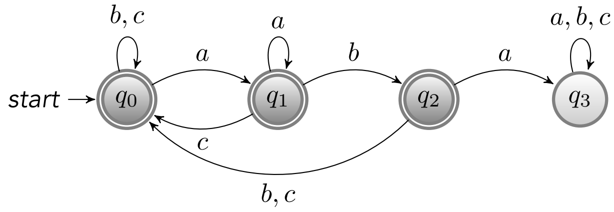
L_1 è riconosciuto dall'automa



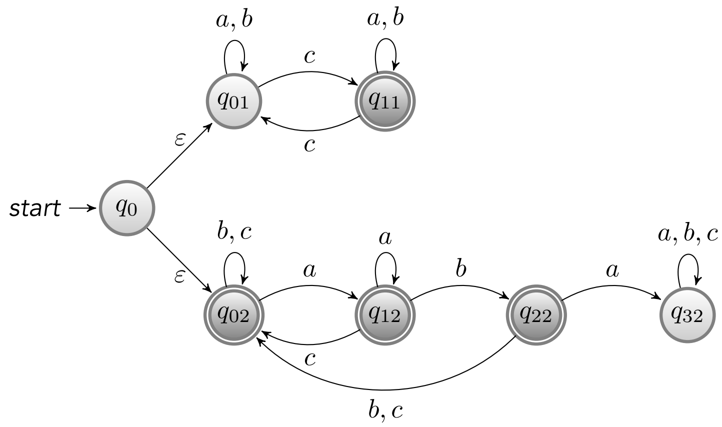
mentre L_2 è riconosciuto da



di conseguenza, $\overline{L}_2$ è riconosciuto da

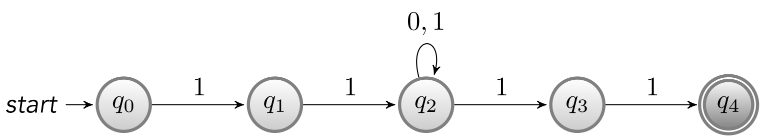


e infine L è riconosciuto da

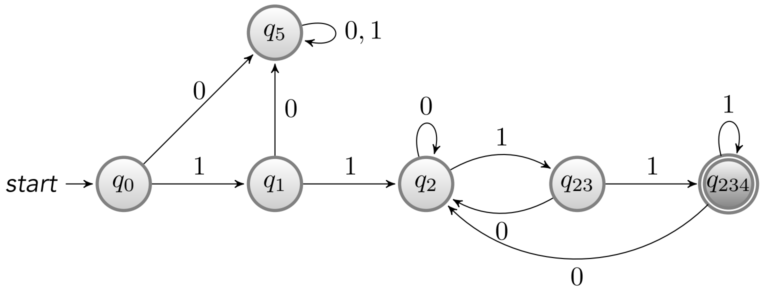


Quesito: Definire un automa deterministico minimo (nel numero di stati) che riconosca il linguaggio $11(0+1)^*11$

Soluzione: *Automa non deterministico che accetta il linguaggio*



Automa deterministico totale equivalente

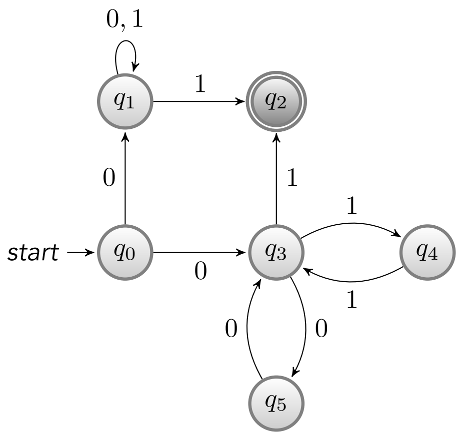


L'automa risulta già minimo, fornendo la matrice di equivalenza seguente (per ogni locazione il carattere che rende i due stati non equivalenti).

	0	*1*	*2*	*23*	*234*	*5*
0						
1	*1*					
2	*1*	*1*				
23	*1*	*1*	*1*			
234	ε	ε	ε	ε		
5	*1*	*1*	*1*	*1*	ε	

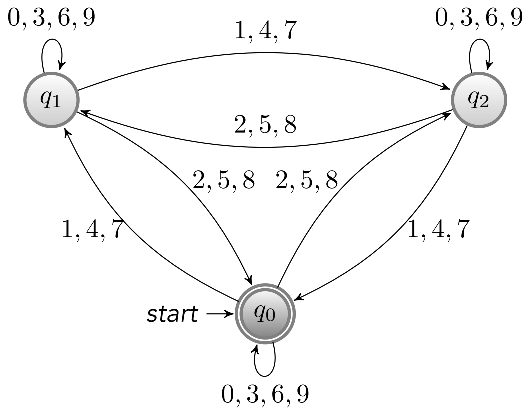
Quesito: Si costruisca un automa a stati finiti non deterministico che accetti il linguaggio L generato dall'espressione regolare $0(1+0)^*1 + 0(11+00)^*1$

Soluzione: *L è riconosciuto dall'automa non deterministico*

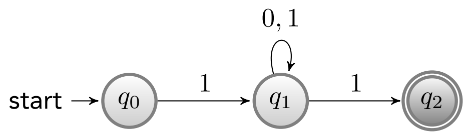


Quesito: Come noto, un numero intero espresso in base 10 è divisibile per 3 se lo è la somma delle sue cifre. Si consideri il linguaggio L comprendente tutte e sole le sequenze di cifre decimali corrispondenti a interi non negativi multipli di 3. Determinare, dimostrandolo, se L è regolare o meno.

Soluzione: *L è regolare e riconosciuto, ad esempio, dall'ASFD seguente, che tiene traccia del resto della divisione per 3 della somma delle cifre decimali lette.*



Quesito: Dato il seguente ASFND A

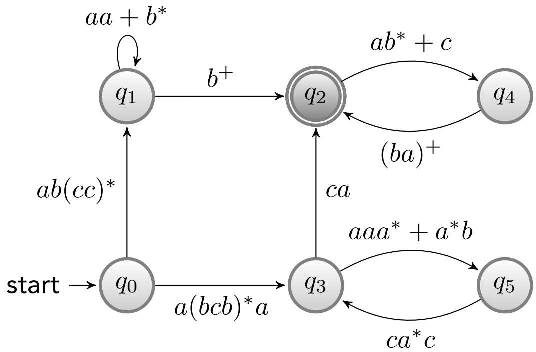


si derivi un grammatica regolare $\mathcal{G}$ tale che $L(\mathcal{G}) = L(A)$

Soluzione: *Applicando la costruzione nota, si ottiene la grammatica*

$$\begin{array}{lcl} S_0 & \rightarrow & 1S_1 \\ S_1 & \rightarrow & 0S_1|1S_1|1 \end{array}$$

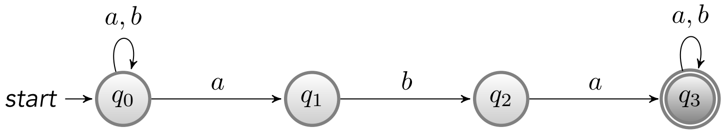
Quesito: Si consideri una estensione dei DFA in cui le transizioni sono associate ad espressioni regolari arbitrarie su Σ. Ad esempio:



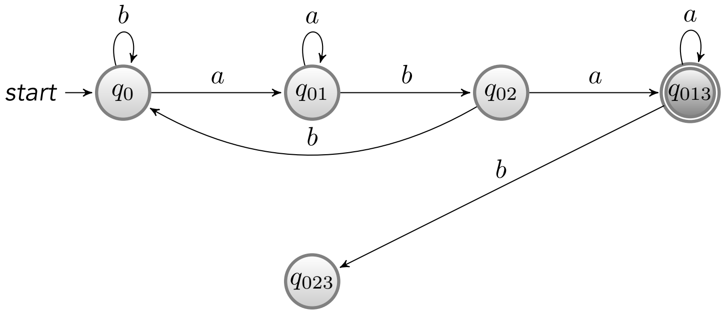
Mostrare che l'insieme dei linguaggi riconoscibili dal modello esteso corrisponde ai linguaggi di tipo 3, mostrando l'equivalenza tra il modello esteso e i DFA.

Quesito: Si definisca una grammatica regolare che generi il linguaggio L composto da tutte le stringhe su $\Sigma = \{a, b\}$ non contenenti la sequenza aba

Soluzione: *Si definisca una automa che accetta il linguaggio complemento $\overline{L}$*



Il corrispondente automa deterministico che riconosce $\overline{L}$

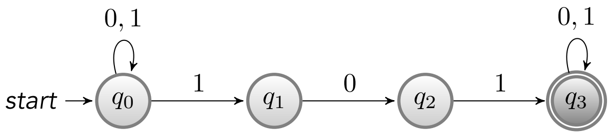


Quesito: Definire una grammatica regolare che generi il seguente linguaggio

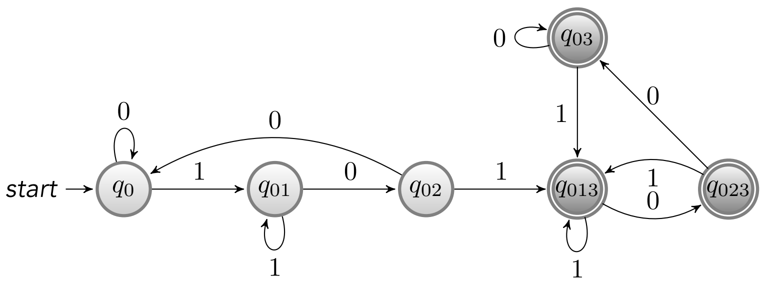
$$L = \{w \in \{0,1\}^* : w \text{ non contiene la sottostringa } 101\}$$

descrivendo e giustificando le scelte effettuate.

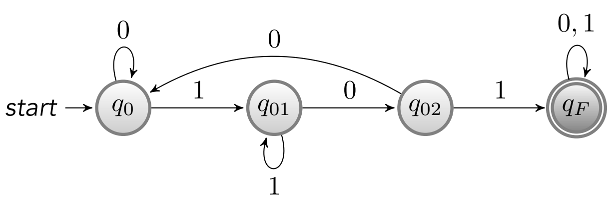
Soluzione: *Definiamo un ASFD $\mathcal{A}$ che riconosca L, derivando poi da esso la grammatica richiesta. A tal fine, definiamo inizialmente un ASFD $\mathcal{A}'$ che riconosca $\overline{L} = \{w \in \{0,1\}^* : w$ contiene la sottostringa $101\}$ a partire dal seguente ASFND $\mathcal{A}'_{\mathcal{N}}$ che riconosce lo stesso linguaggio.*



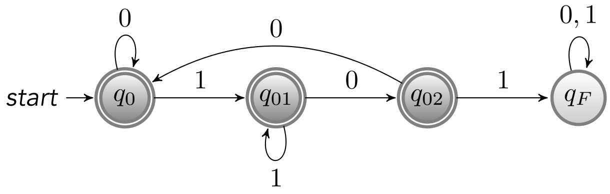
L'ASFD equivalente risulta allora



Gli stati q_{03}, q_{013}, q_{023} risultano immediatamente equivalenti, per cui possiamo assumere $\mathcal{A}'$ come



$\mathcal{A}$ *deriva scambiando stati finali e non:*



Da cui la grammatica:

$$
\begin{aligned}
S &\rightarrow 0S|1A_1|0|1 \\
A_1 &\rightarrow 1S|0A_2|0|1 \\
A_2 &\rightarrow 0S|1A_3|0 \\
A_3 &\rightarrow 0A_3|1A_3
\end{aligned}
$$

Il simbolo A_3 è chiaramente inutile, in quanto non fecondo, per cui la grammatica può essere immediatamente semplificata

$$
\begin{aligned}
S &\rightarrow 0S|1A_1|0|1 \\
A_1 &\rightarrow 1S|0A_2|0|1 \\
A_2 &\rightarrow 0S|0
\end{aligned}
$$

Quesito: Si determini se i linguaggi

$$L = \{a^i b^j c^i | i, j \geq 1\}$$

e

$$L = \{a^i b^j c^k | i, j, k \geq 0\}$$

sono regolari.

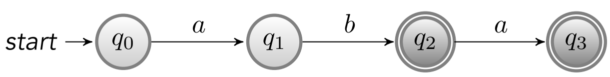
Soluzione: *Il primo linguaggio non è regolare: per dimostrare ciò possiamo utilizzare il pumping lemma. Dato $n > 0$, consideriamo la stringa $\sigma = a^n bc^n$: qualunque decomposizione $\sigma = uvw$ con $|uv| \leq n$ e $|v| > 1$ fa sÃ¬ che $v = a^k$ per un qualche $0 < k < n$. Ne deriva che la stringa $\sigma' = uv^2w = a^{n+k}bc^n \notin L$, da cui la non regolarità di L.*

Il secondo linguaggio è invece regolare: infatti può essere descritto dall'espressione regolare $a^*b^*c^*$.

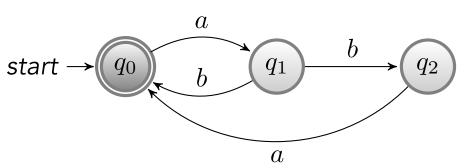
Quesito: Costruire un ASFD che riconosca il linguaggio descritto dall'espressione regolare $a((ab+aba)^*a)^*$

Soluzione: *Deriviamo un ASFND che riconosce il linguaggio in modo graduale.*

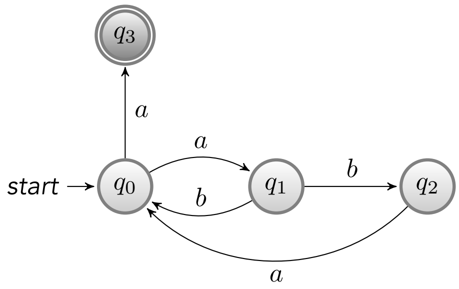
L'ASFND che riconosce $ab + aba$ è



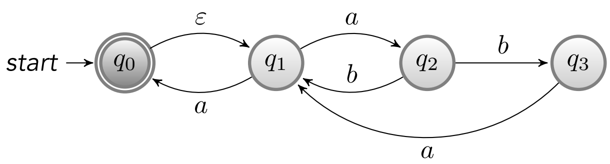
Da cui l'automa che riconosce $(ab + aba)^*$



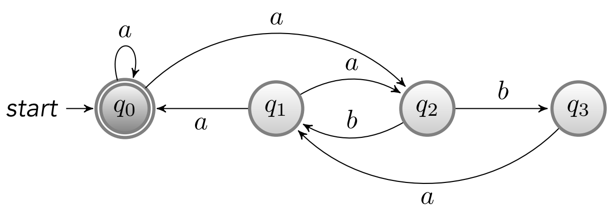
e quello che riconosce $(ab + aba)^*a$



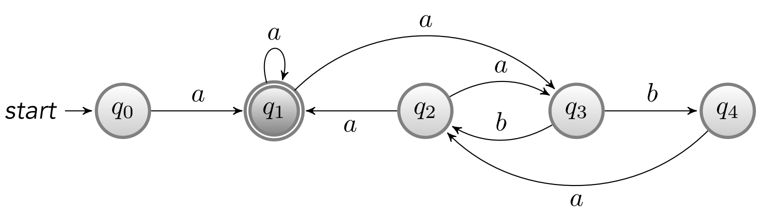
Il linguaggio descritto da $((ab + aba)^*a)^*$ è allora accettato da



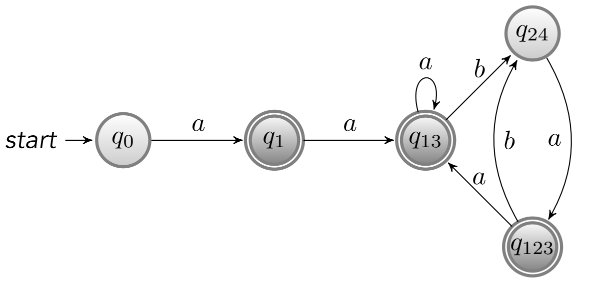
ed eliminando la ε-transizione considerando la ε chiusura,



Ne deriva che l'ASFND che riconosce il linguaggio è

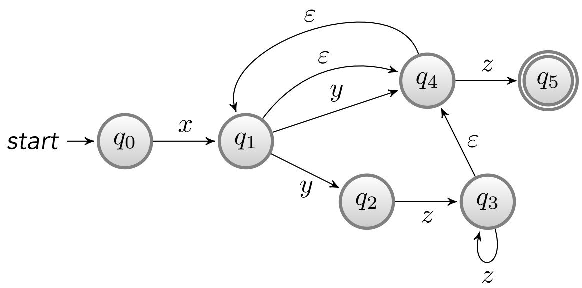


Da cui l'ASFD equivalente

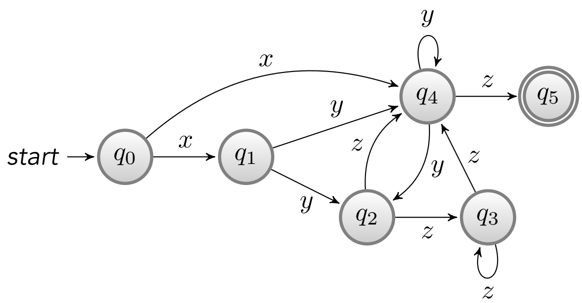


Quesito: Sia data l'espressione regolare $r = x(y|yz^+)^*z$. Derivare un automa deterministico minimo che accetti $L(r)$, il linguaggio descritto da r.

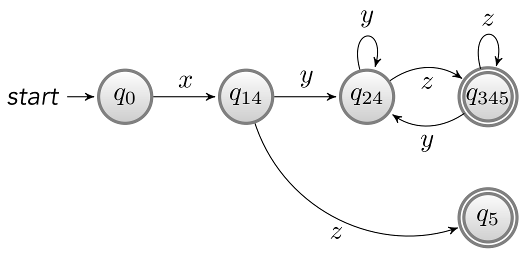
Soluzione: *Dalla espressione regolare possiamo derivare un ASFND con ε-transizioni che accetta $L(r)$*



e da questo un ASFND senza ε-transizioni equivalente



Possiamo quindi derivare un AFSD che riconosce $L(r)$



L'automa risulta minimo, in quanto tutti gli stati risultano distinguibili. Infatti q_{345} e q_5 sono distinguibili da q_0, q_{14} e q_{24} in quanto stati finali. Le uniche coppie di stati che potrebbero risultare indistinguibili sono quindi:

1. *q_0 e q_{14}: sono distinguibili in quanto $\delta(q_0, z)$ è indefinita mentre $\delta(q_{14}, z) = q_5 \in F$*
2. *q_0 e q_{24}: sono distinguibili in quanto $\delta(q_0, z)$ è indefinita mentre $\delta(q_{24}, z) = q_{345} \in F$*
3. *q_{14} e q_{24}: sono indistinguibili rispetto a x e y, mentre lo sono rispetto a z se q_5 e q_{345} risultano indistinguibili anch'essi*
4. *q_5 e q_{345}: sono distinguibili in quanto $\delta(q_5, z)$ è indefinita mentre $\delta(q_{345}, z) = q_{345} \in F$. Da questo consegue che q_{14} e q_{24} sono distinguibili.*

Quesito: Si consideri il seguente linguaggio

$$L = \{0^i 1^j 0^i | i, j > 0\}$$

Il linguaggio è regolare? Si dimostri l'affermazione fatta.

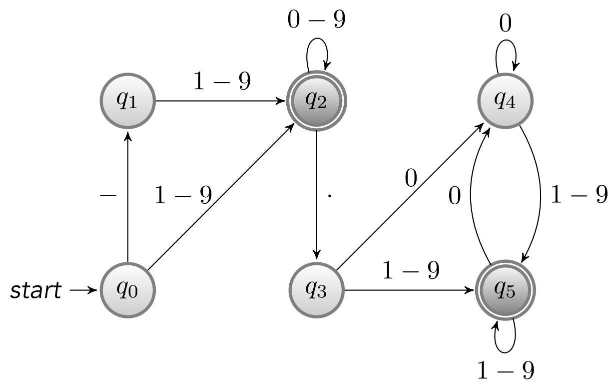
Soluzione: *Il linguaggio non è regolare. Ciò si può dimostrare applicando il pumping lemma nel modo seguente: fissato n, consideriamo la stringa $\sigma = 0^n 1 0^n \in L$: una qualunque decomposizione $\sigma = uvw$ che soddisfi le ipotesi del pumping lemma prevede che $|uv| \leq n$, per cui uv è una sequenza di caratteri 0, per cui anche v è una sequenza di 0, diciamo $v = 0^k$ per qualche $k > 0$. Se consideriamo la stringa $\sigma' = uv^2w$ ne consegue che $\sigma' = 0^{n+k}10^n \notin L$, per cui L non è regolare.*

Quesito: Si considerino un alfabeto $\Sigma = \{a_1, a_2, \ldots, a_5\}$ e il linguaggio L comprendente tutte e sole stringhe $\sigma \in \Sigma^*$ tali che il numero di caratteri diversi che occorrono in σ è maggiore di 3. Definire la struttura di un ASFD che riconosce tale linguaggio.

Soluzione:

Quesito:Definire un automa a stati finiti deterministico che riconosca tutte e sole le stringhe che rappresentano numeri secondo il seguente formato: un segno $-$ opzionale seguito da una sequenza di almeno una cifra decimale, non iniziante per 0, o un segno $-$ opzionale seguito da una sequenza di almeno una cifra decimale, non iniziante per 0, un punto e una sequenza di almeno una cifra decimale, non terminante per 0.

Soluzione: *Il linguaggio è riconosciuto dall'automa deterministico*

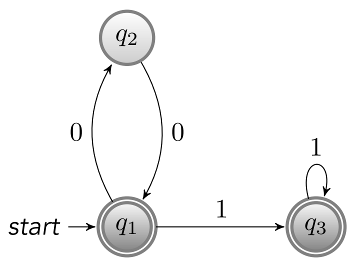


Quesito:Si consideri il linguaggio

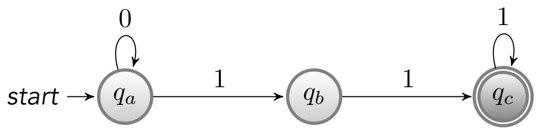
$$L = \{0^i 1^j | i \text{ pari o } 0, \text{ oppure} j \geq 2\}$$

Il linguaggio è regolare o context free? Dimostrare quale delle due affermazioni è vera. Si definisca inoltre una grammatica (di tipo 3 o di tipo 2, rispettivamente) che generi tutte e sole le stringhe del linguaggio.

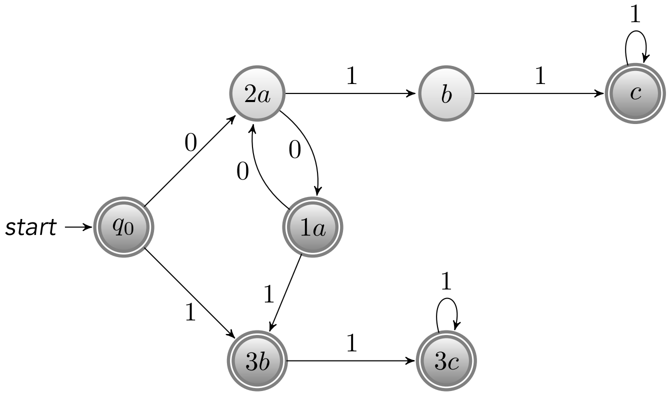
Soluzione: *Il linguaggio è regolare, unione di $L_1 = \{0^i 1^j | i \text{ pari o } 0\}$ e $L_2 = \{0^i 1^j | j \geq 2\}$, regolari in quanto L_1 è riconoscibile da*



e L_2 è riconoscibile da



La composizione dei due automi permette di ottenere l'ASF che riconosce L



da cui la grammatica che genera L

$$
\begin{array}{rcl}
S & \to & 0A_{2a}|1A_{3b}|1|\varepsilon \\
A_{2a} & \to & 0A_{1a}|1A_b|0 \\
A_{3b} & \to & 1A_{3c}|1 \\
A_{1a} & \to & 0A_{2a}|1A_{3b}|0|1 \\
A_b & \to & 1A_c|1 \\
A_c & \to & 1A_c|1 \\
A_{3c} & \to & 1A_{3c}|1
\end{array}
$$

Quesito: Si consideri il linguaggio $L \subset \{0,1\}^*$ tale che $\sigma \in L$ se e solo se $\#_0(\sigma) = \#_1(\sigma)$, dove $\#_a(s)$ indica il numero di occorrenze del carattere a nella stringa s. Si definisca una grammatica CF in GNF che generi L.

Soluzione: *Una possibile grammatica è la seguente:*

$$S \to 0S1S|1S0S|\varepsilon$$

L'eliminazione della ε-produzione porta alla grammatica equivalente

$$S \to 0S1S|1S0S|0S1|01S|1S0|10S|01|10$$

che non presenta produzioni unitarie o simboli inutili.

La grammatica in CNF che ne deriva è

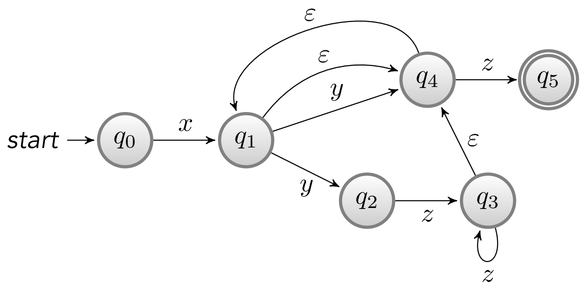
$$
\begin{array}{rcl}
S & \to & XY|YX|XU|ZY|YZ|UX|ZU|UZ \\
X & \to & ZS \\
Y & \to & US \\
Z & \to & 0 \\
U & \to & 1
\end{array}
$$

da cui deriva immediatamente la grammatica in GNF

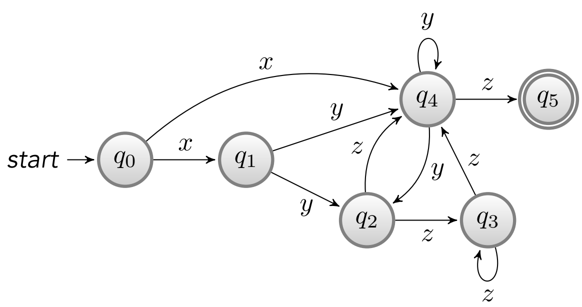
$$\begin{array}{rcl} S & \to & 0SY|1SX|0SU|0Y|1SZ|1X|0U|1Z \\ X & \to & 0S \\ Y & \to & 1S \\ Z & \to & 0 \\ U & \to & 1 \end{array}$$

Quesito: Sia data l'espressione regolare $r = x(y|yz^+)^*z$. Derivare un automa deterministico minimo che accetti $L(r)$, il linguaggio descritto da r.

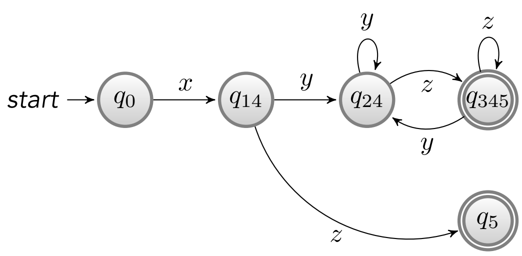
Soluzione: *Dalla espressione regolare possiamo derivare un ASFND con ε-transizioni che accetta* $L(r)$



e da questo un ASFND senza ε-transizioni equivalente



Possiamo quindi derivare un AFSD che riconosce $L(r)$



L'automa risulta minimo, in quanto tutti gli stati risultano distinguibili. Infatti q_{345} e q_5 sono distinguibili da q_0, q_{14} e q_{24} in quanto stati finali. Le uniche coppie di stati che potrebbero risultare indistinguibili sono quindi:

1. q_0 *e* q_{14}*: sono distinguibili in quanto* $\delta(q_0, z)$ *è indefinita mentre* $\delta(q_{14}, z) = q_5 \in F$

2. q_0 e q_{24}: *sono distinguibili in quanto* $\delta(q_0, z)$ *è indefinita mentre* $\delta(q_{24}, z) = q_{345} \in F$

3. q_{14} e q_{24}: *sono indistinguibili rispetto a* x *e* y*, mentre lo sono rispetto a* z *se* q_5 *e* q_{345} *risultano indistinguibili anch'essi*

4. q_5 e q_{345}: *sono distinguibili in quanto* $\delta(q_5, z)$ *è indefinita mentre* $\delta(q_{345}, z) = q_{345} \in F$*. Da questo consegue che* q_{14} *e* q_{24} *sono distinguibili.*

Quesito: Si consideri il seguente linguaggio

$$L = \{0^i 1^j 0^i | i, j > 0\}$$

Il linguaggio è regolare? Si dimostri l'affermazione fatta.

Soluzione: *Il linguaggio non è regolare. Ciò si può dimostrare applicando il pumping lemma nel modo seguente: fissato* n*, consideriamo la stringa* $\sigma = 0^n 1 0^n \in L$*: una qualunque decomposizione* $\sigma = uvw$ *che soddisfi le ipotesi del pumping lemma prevede che* $|uv| \leq n$*, per cui* uv *è una sequenza di caratteri* 0*, per cui anche* v *è una sequenza di* 0*, diciamo* $v = 0^k$ *per qualche* $k > 0$*. Se consideriamo la stringa* $\sigma' = uv^2w$ *ne consegue che* $\sigma' = 0^{n+k} 1 0^n \notin L$*, per cui* L *non è regolare.*

Quesito: Si considerino un alfabeto $\Sigma = \{a_1, a_2, \ldots, a_5\}$ e il linguaggio L comprendente tutte e sole stringhe $\sigma \in \Sigma^*$ tali che il numero di caratteri diversi che occorrono in σ è maggiore di 3. Definire la struttura di un ASFD che riconosce tale linguaggio.

Soluzione:

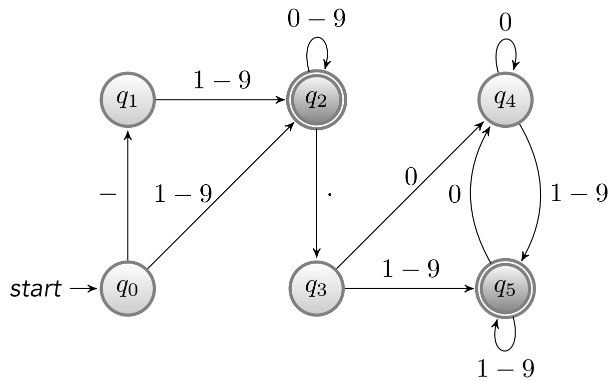
Quesito: Si definisca una grammatica CF che generi il linguaggio $L = \{a^r b^s c^t | s = r + t; r, s, t > 0\}$

Soluzione: *Una possibile grammatica che generi* L *è ad esempio:*

$$\begin{array}{rcl} S & \rightarrow & S_1 S_2 \\ S_1 & \rightarrow & aS_1b|ab \\ S_2 & \rightarrow & bS_2c|bc \end{array}$$

Quesito:Definire un automa a stati finiti deterministico che riconosca tutte e sole le stringhe che rappresentano numeri secondo il seguente formato: un segno $-$ opzionale seguito da una sequenza di almeno una cifra decimale, non iniziante per 0, o un segno $-$ opzionale seguito da una sequenza di almeno una cifra decimale, non iniziante per 0, un punto e una sequenza di almeno una cifra decimale, non terminante per 0.

Soluzione: *Il linguaggio è riconosciuto dall'automa deterministico*

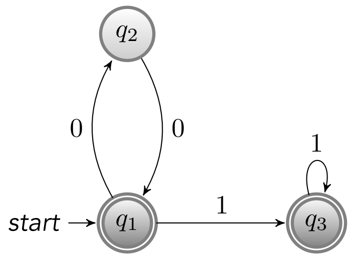


Quesito:Si consideri il linguaggio

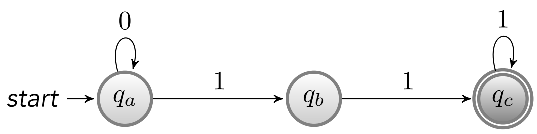
$$L = \{0^i 1^j | i \text{ pari o } 0, \text{ oppure} j \geq 2\}$$

Il linguaggio è regolare o context free? Dimostrare quale delle due affermazioni è vera. Si definisca inoltre una grammatica (di tipo 3 o di tipo 2, rispettivamente) che generi tutte e sole le stringhe del linguaggio.

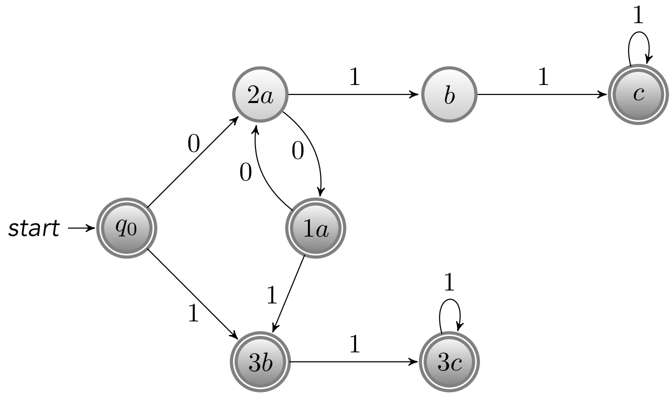
Soluzione: *Il linguaggio è regolare, unione di* $L_1 = \{0^i 1^j | i \text{ pari o } 0\}$ *e* $L_2 = \{0^i 1^j | j \geq 2\}$*, regolari in quanto* L_1 *è riconoscibile da*



e L_2 *è riconoscibile da*



La composizione dei due automi permette di ottenere l'ASF che riconosce L



da cui la grammatica che genera L

$$\begin{array}{rcl} S & \rightarrow & 0A_{2a}|1A_{3b}|1|\varepsilon \\ A_{2a} & \rightarrow & 0A_{1a}|1A_b|0 \\ A_{3b} & \rightarrow & 1A_{3c}|1 \\ A_{1a} & \rightarrow & 0A_{2a}|1A_{3b}|0|1 \\ A_b & \rightarrow & 1A_c|1 \\ A_c & \rightarrow & 1A_c|1 \\ A_{3c} & \rightarrow & 1A_{3c}|1 \end{array}$$

Quesito: Si consideri la seguente operazione $\mathcal{I}()$ definita come:

$$\mathcal{I}(L) = \{x_1 x_2 \cdots x_k | k \geq 1, x_i \in L \text{ per } i = 1, \ldots, k\}$$

Mostrare che la classe dei linguaggi regolari è chiusa rispetto a $\mathcal{I}()$.

Soluzione: *Si considerino i linguaggi, per $k \geq 1$*

$$L_k = \{x_1 x_2 \cdots x_k | x_i \in L \text{ per } i = 1, \ldots, k\}$$

Se L è regolare allora ogni L_k è regolare in quanto $L_k = L^k$, potenza k-esima di L, e i linguaggi regolari sono chiusi rispetto alla concatenazione (e quindi alla potenza).

Ma $\mathcal{I}(L) = \cup_{k \geq 1} L_k$, per cui se L è regolare $\mathcal{I}(L)$ è l'unione di linguaggi regolari: per la chiusura dei linguaggi regolari rispetto all'unione, ne deriva che $\mathcal{I}(L)$ è regolare se lo è L.

Quesito:Si consideri il linguaggio

$$L = \{0^i 1^j | i \geq j\}$$

Il linguaggio è regolare o context free? Dimostrare quale delle due affermazioni è vera. Si definisca inoltre una grammatica (di tipo 3 o di tipo 2, rispettivamente) che generi tutte e sole le stringhe del linguaggio.

Soluzione: *Il linguaggio non è regolare, ma è context free. Per verificare che non è regolare si può utilizzare il pumping lemma applicato (fissato n)alla stringa $0^n 1^n \in L$. Dato che per ogni $uvx = 0^n 1^n$ con $|uv| \leq n$ e $|v| \geq 1$ si deve avere necessariamente che $v = 0^k$ per un qualche $k > 0$, si che $uv^0 w = uv = 0^{n-k} 1^k \notin L$, per cui L non è regolare.*

Una grammatica CF che genera L è ad esempio

$$\begin{array}{rcl} S & \rightarrow & 0S1|0T1|\varepsilon \\ T & \rightarrow & 0T|0 \end{array}$$

Quesito: Si definiscano una grammatica in CNF e una grammatica in GNF che generino il linguaggio L composto da tutte le stringhe su $\Sigma = \{0, 1\}$ che iniziano e terminano per lo stesso carattere.

Soluzione: *Una grammatica CF che genera L è ad esempio*

$$\begin{array}{rcl} S & \rightarrow & 0T0|1T1|00|11 \\ T & \rightarrow & 0T|1T|0|1 \end{array}$$

La grammatica è già in forma ridotta.

Una grammatica in CNF risultante è allora

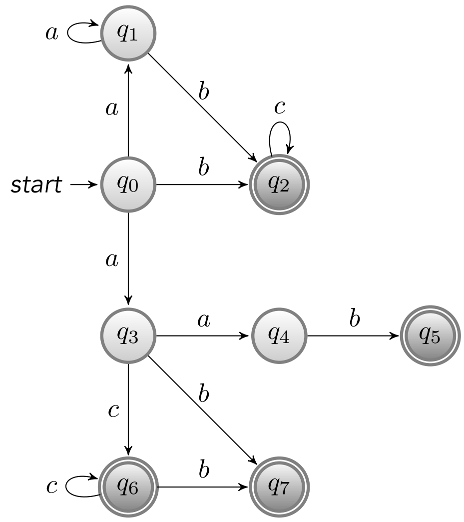
$$\begin{array}{rcl} S & \rightarrow & XZ|YU|ZZ|UU \\ T & \rightarrow & ZT|UT|0|1 \\ X & \rightarrow & ZT \\ Y & \rightarrow & UT \\ Z & \rightarrow & 0 \\ U & \rightarrow & 1 \end{array}$$

e una grammatica in GNF è

$$\begin{array}{rcl} S & \rightarrow & 0TZ|1TU|0Z|1U \\ T & \rightarrow & 0T|1T|0|1 \\ X & \rightarrow & 0T \\ Y & \rightarrow & 1T \\ Z & \rightarrow & 0 \\ U & \rightarrow & 1 \end{array}$$

Quesito: Definire una grammatica di tipo 3, priva di simboli inutili, che generi il linguaggio descritto dall'espressione regolare $a^*bc^* + a(ab + c^*b)$

Soluzione: *Il linguaggio è riconosciuto dall'automa*



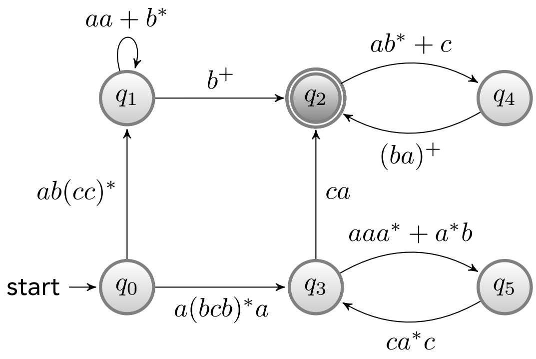
Da cui deriva la grammatica:

$$\begin{array}{rcl}
S & \rightarrow & aS|aA_1|bA_2|aA_3|b \\
A_1 & \rightarrow & aA_1|cbA_2|b \\
A_2 & \rightarrow & cA_2|c \\
A_3 & \rightarrow & aA_4|bA_7|cA_6|b|c \\
A_4 & \rightarrow & bA_5|b \\
A_6 & \rightarrow & cA_6|bA_7|c|b
\end{array}$$

ed eliminando i simboli inutili (non fecondi) A_5, A_7

$$\begin{array}{rcl}
S & \rightarrow & aS|aA_1|bA_2|aA_3|b \\
A_1 & \rightarrow & aA_1|cbA_2|b \\
A_2 & \rightarrow & cA_2|c \\
A_3 & \rightarrow & aA_4|cA_6|b|c \\
A_4 & \rightarrow & b \\
A_6 & \rightarrow & cA_6|c|b
\end{array}$$

Quesito: Si consideri una estensione dei DFA in cui le transizioni sono associate ad espressioni regolari arbitrarie su Σ. Ad esempio:



Mostrare che l'insieme dei linguaggi riconoscibili dal modello esteso corrisponde ai linguaggi di tipo 3, mostrando l'equivalenza tra il modello esteso e i DFA.

Soluzione:

Quesito: Definire un automa a pila che accetta per stato finale il linguaggio composto dalle stringhe $w \in \{0,1\}^+$ contenenti uno stesso numero di 0 e di 1.

Soluzione: *Un possibile automa ha 2 soli stati q_0, q_F e un alfabeto di pila Z_0, Z, U. Ad ogni istante la pila contiene, al di sopra di Z_0, una sequenza di Z di dimensione pari a $\#(0)-\#(1)$ se $\#(0)-\#(1) > 0$ o una sequenza di U di dimensione pari a $\#(1) - \#(0)$ se $\#(0) - \#(1) < 0$.*

	(q_0, 0)	(q_0, 1)	(q_0,ε)
Z_0	(q_0,ZZ_0)	(q_0,UZ_0)	(q_F,ε)
Z	(q_0,ZZ)	(q_0,ε)	-
U	(q_0,ε)	(q_0,UU)	-

Quesito: Verificare se il linguaggio

$$L = \{a^i b^j c^k | i < j \wedge i < k\}$$

è context free o meno.

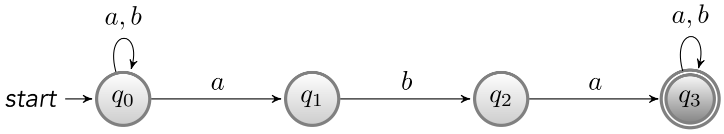
Soluzione: *Applicando il pumping lemma per i CFL, abbiamo che se L è regolare esiste un n tale che per $i + j + k > n$ possiamo scrivere $z = uvwxy$ con $|vx| > 1$ e $|vwx| <\leq n$, e che $uv^iwx^iy \in L$ per ogni $i \geq 0$.*

Consideriamo la stringa $a^m b^{[}m + 1]c^{[}m + 1]$, con $n = 3m + 2$, e osserviamo che per qualunque decomposizione $z = uvwxy$:

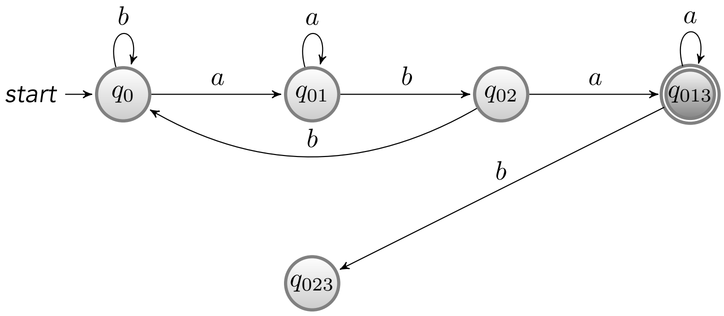
- *se v o x corrispondono a sequenze non omogenee (a^r, b^s, c^t), allora $uv^2wx^2y \notin L$*
- *altrimenti, se $v = a^r$ e $x = b^s$, se $r > 0$ la stringa $uv^2wx^2y \notin L$ in quanto il numero di a è maggiore del numero di c; se $r = 0$ la stringa $uwy \notin L$ in quanto il numero di b è minore o uguale del numero di a. Le stesse considerazioni valgono se $v = a^r$ e $x = c^s$.*
- *infine, se $v = b^r$ e $x = c^s$, la stringa $uvwxy \notin L$ in quanto il numero di a è maggiore o uguale di almeno uno tra il numero di b e il numero di c;*

Quesito: Si definisca una grammatica regolare che generi il linguaggio L composto da tutte le stringhe su $\Sigma = \{a, b\}$ non contenenti la sequenza aba

Soluzione: *Si definisca una automa che accetta il linguaggio complemento $\overline{L}$*



Il corrispondente automa deterministico che riconosce $\overline{L}$



Quesito: Definire un automa a pila che accetti il seguente linguaggio

$$L = \{a^p b^{p+2q} a^q ; p, q > 0\}$$

descrivendo e giustificando le scelte effettuate.

Soluzione: *Un possibile NPDA che accetta il linguaggio è il seguente.*

	(q_0, Z_0)	(q_0, A)	(q_1, Z_0)	(q_1, A)	(q_2, B)	(q_3, B)	(q_4, Z_0)	(q_4, B)
a	(q_0, AZ_0)	(q_0, AA)	-	-	(q_3, ε)	-	-	(q_3, ε)
b	-	(q_1, ε)	(q_2, BZ_0)	(q_1, ε)	(q_2, BB)	-	-	-
ε	-	-	-	-	-	(q_4, ε)	(q_4, ε)	-

Nello stato q_0 vengono posti nella pila tanti simboli A quanti simboli a sono letti. Lo stato diventa q_1 al primo simbolo b letto: in tale stato, un simbolo A viene tolto dalla pila per ogni b letto, fino a giungere al fondo della pila e passare in q_2. In questo stato, per ogni simbolo b letto viene posto sulla pila un simbolo B. L'automa passa in q_3 quando legge un nuovo simbolo a: a questo punto, per ogni simbolo a letto dovrà togliere due simboli B: per far ciò, passerà ciclicamente in q_3, in cui toglierà la prima B dalla pila avendo letto a, e in q_4, in cui toglierà la seconda B con una ε-transizione. Infine, se l'automa si trova in q_4, ed ha quindi tolto BB dalla pila avendo letto a, può eliminare Z_0 dalla pila con una ε. La stringa è accettata per pila vuota.

Un approccio alternativo è basato sulla definizione di una CFG per il linguaggio e sulla derivazione da essa di un NPDA, secondo il metodo visto a lezione.

Grammatica:

$$\begin{array}{rcl} S & \rightarrow & XY \\ X & \rightarrow & aXb|ab \\ Y & \rightarrow & bbYa|bba \end{array}$$

La grammatica non ha ε-produzioni, produzioni unitarie o simboli inutili, per cui è già in forma ridotta.

In CNF:

$$
\begin{array}{lcl}
S & \to & XY \\
X & \to & AZ|AB \\
Y & \to & VW|VA \\
Z & \to & XB \\
V & \to & BB \\
W & \to & YA \\
A & \to & a \\
B & \to & b
\end{array}
$$

In GNF:

$$
\begin{array}{lcl}
S & \to & aZY|aBY \\
X & \to & aZ|aB \\
Y & \to & bBW|bBA \\
Z & \to & aZB|aBB \\
V & \to & bB \\
W & \to & bBWA|bBAA \\
A & \to & a \\
B & \to & b
\end{array}
$$

NPDA: L'automa ha un solo stato, che per brevità non viene riportato.

	S	X	Y	Z	V	W	A	B
a	ZY, BY	Z, B	-	ZB, BB	-	-	ε	-
b	-	-	BW, BA	-	B	BWA, BAA	-	ε

Quesito: Si consideri il linguaggio

$$L = \{w \in \{a, b, c\}^+, \#_a(w) = \#_b(w) = \#_c(w)\}$$

dove con $\#_c(x)$ indichiamo il numero di occorrenze del carattere c nella stringa x.

Si mostri che L non è context free.

Soluzione: *Possiamo applicare il pumping lemma, considerando ad esempio, dato $n > 0$, la stringa $\sigma = a^n b^n c^n$.*

Per qualunque decomposizione $\sigma = uvwxy$ con $|vwx| \leq n$ si deve necessariamente avere che vwx (e quindi vx) non può contenere sia caratteri a che caratteri b che caratteri c. Inoltre, per costruzione, $|vx| \geq 1$.

Consideriamo ad esempio il caso in cui $\#_a(vx) = 0$: allora avremo, relativamente a $\sigma' = uv^2wx^2y$, che $\#_a(\sigma') = \#_a(\sigma)$, $\#_b(\sigma') = \#_b(\sigma) + \#_b(vx)$ e $\#_c(\sigma') = \#_c(\sigma) + \#_c(vx)$, in cui $\#_b(vx) + \#_c(vx) > 0$. Ne deriva che $\sigma' \notin L$, e quindi che L non è context free. Lo stesso chiaramente vale se assumiamo $\#_b(vx) = 0$ o $\#_c(vx) = 0$.

Quesito: Sia dato il linguaggio

$$L = \{w \in \{a, b\}^*\}$$

tale che:

1. w non contiene la stringa aa

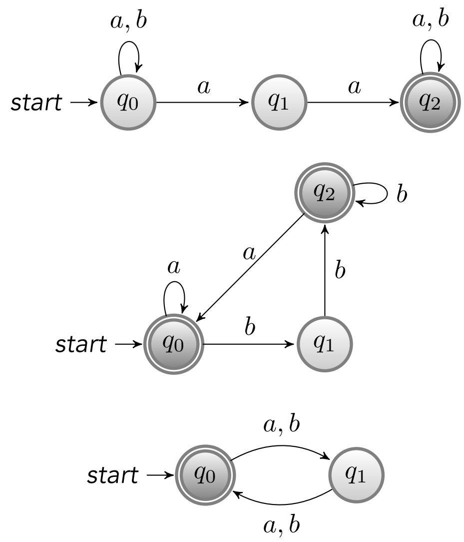
2. nessun carattere b in w compare "isolato", vale a dire senza almeno un altro b adiacente (che lo precede o lo segua)
3. $|w|$ è pari

Dimostrare che L è regolare.

Soluzione: *Possiamo considerare i tre linguaggi:*

- $L_1 = \{w$ *contiene* $aa\}$
- $L_2 = \{w$ *non compaiono* b *isolati*$\}$
- $L_3 = \{w : |w|$ *pari*$\}$

I tre linguaggi sono regolari, in quanto, ad esempio, accettati rispettivamente dagli ASF seguenti



L risulta regolare, per le proprietà di chiusura dei linguaggi regolari, in quanto

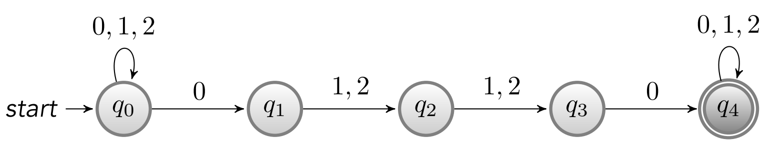
$$L = \overline{L}_1 \cap L_2 \cap L_3$$

Quesito: Si consideri il linguaggio

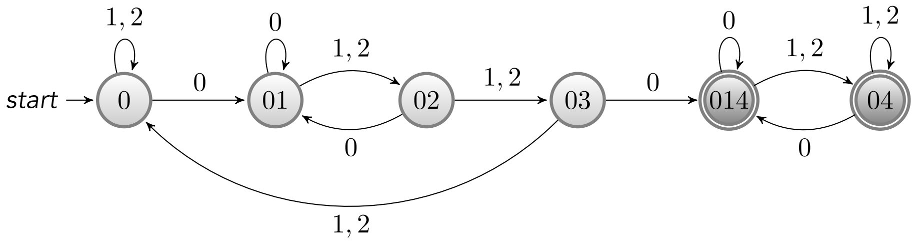
$$L = \{w \in \{0,1,2\}^* | w \text{ contiene una sottostringa } 0x0\text{, con } x \in \{1,2\}^2\}$$

Si definisca un ASFD minimo che riconosce L

Soluzione: *ASFND che accetta L:*



da cui l'ASFD



L'automa minimo deriva osservando che i soli stati indistinguibili sono 014 *e* 04*, che possono quindi essere unificati.*

Quesito: Definire una CFG in CNF che generi il linguaggio

$$L = \{a^n b^m c^k : k = 2(n+m)\}$$

Soluzione:

$$\begin{array}{rcl} S & \to & aScc|X|\varepsilon \\ X & \to & bXcc|\varepsilon \end{array}$$

Eliminazione delle ε*-produzioni (tenendo conto che* $\varepsilon \in L$*)*

$$\begin{array}{rcl} S' & \to & S|\varepsilon \\ S & \to & aScc|X|acc \\ X & \to & bXcc|bcc \end{array}$$

Eliminazione delle produzioni unitarie

$$\begin{array}{rcl} S' & \to & aScc|bXcc|bcc|acc|\varepsilon \\ S & \to & aScc|bXcc|bcc|acc \\ X & \to & bXcc|bcc \end{array}$$

Non ci sono simboli inutili.
CNF

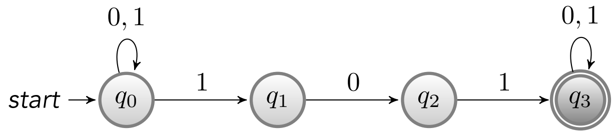
$$\begin{array}{rcl} S' & \to & ASY|BXY|BY|AY|\varepsilon \\ S & \to & CSY|BXY|BY|AY \\ X & \to & BXY|BY \\ Y & \to & CC \\ A & \to & a \\ B & \to & b \\ C & \to & c \end{array}$$

Quesito: Definire una grammatica regolare che generi il seguente linguaggio

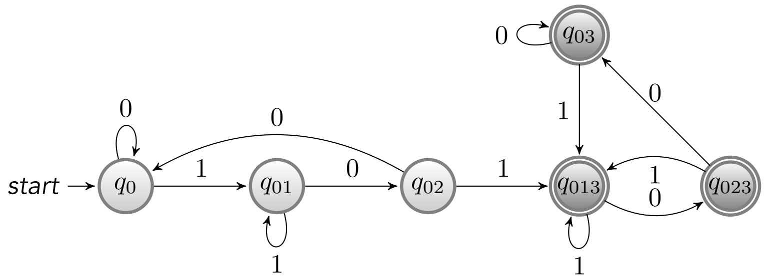
$$L = \{w \in \{0,1\}^* : w \text{ non contiene la sottostringa } 101\}$$

descrivendo e giustificando le scelte effettuate.

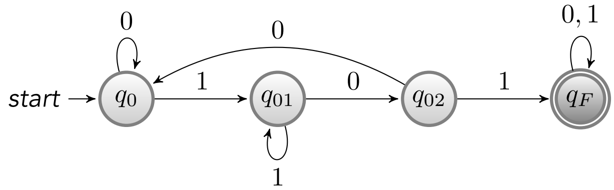
Soluzione: *Definiamo un ASFD* $\mathcal{A}$ *che riconosca* L*, derivando poi da esso la grammatica richiesta.*
A tal fine, definiamo inizialmente un ASFD $\mathcal{A}'$ *che riconosca* $\overline{L} = \{w \in \{0,1\}^* : w$ *contiene la sottostringa* $101\}$ *a partire dal seguente ASFND* $\mathcal{A}'_{\mathcal{N}}$ *che riconosce lo stesso linguaggio.*



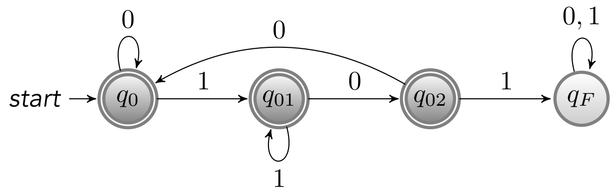
L'ASFD equivalente risulta allora



Gli stati q_{03}, q_{013}, q_{023} risultano immediatamente equivalenti, per cui possiamo assumere $\mathcal{A}'$ come



$\mathcal{A}$ deriva scambiando stati finali e non:



Da cui la grammatica:

$$\begin{array}{rcl} S & \rightarrow & 0S|1A_1|0|1 \\ A_1 & \rightarrow & 1S|0A_2|0|1 \\ A_2 & \rightarrow & 0S|1A_3|0 \\ A_3 & \rightarrow & 0A_3|1A_3 \end{array}$$

Il simbolo A_3 è chiaramente inutile, in quanto non fecondo, per cui la grammatica può essere immediatamente semplificata

$$\begin{array}{rcl} S & \rightarrow & 0S|1A_1|0|1 \\ A_1 & \rightarrow & 1S|0A_2|0|1 \\ A_2 & \rightarrow & 0S|0 \end{array}$$

Quesito: Si consideri il linguaggio

$$L = \{w\#x | w, x \in \{0,1\}^+, w^R \text{ è suffisso di } x\}$$

Si verifichi che L è context free definendo un automa a pila che lo accetta.

Soluzione: *Un possibile NPDA che accetta il linguaggio è il seguente.*

	(q_0, Z_0)	(q_0, Z)	(q_0, U)	(q_1, Z)	(q_1, U)	(q_2, Z)	(q_2, U)	(q_2, Z_0)
0	(q_0, ZZ_0)	(q_0, ZZ)	(q_0, UZ)	$\{(q_1, Z), (q_2, \varepsilon)\}$	(q_1, U)	(q_2, ε)	-	-
1	(q_0, UZ_0)	(q_0, UZ)	(q_0, UU)	(q_1, Z)	$\{(q_1, U), (q_2, \varepsilon)\}$	-	(q_2, ε)	-
#	-	(q_1, Z)	(q_1, U)	-	-	-	-	-
ε	-	-	-	-	-	-	-	(q_2, ε)

L'automa dapprima (nello stato q_0) legge w e la trascrive sulla pila in ordine inverso. Alla lettura del carattere # l'automa passa nello stato q_1 di lettura di x: in qualunque passo in cui il carattere letto corrisponde a quello in cima alla pila l'automa effettua una scelta non deterministica tra due opzioni:

1. *assumere che w^R compaia in x a partire da questo carattere, in tal caso passa nello stato q_2 ed elimina il primo carattere dalla pila*
2. *assumere che w^R non compaia in x a partire da questo carattere, e continuare a leggere caratteri, nello stato q_1*

Nello stato q_2, l'automa procede nella computazione fin tanto che i caratteri letti corrispondono a quelli via via estratti dalla pila. Nel caso positivo, la stringa termina con Z_0 sulla pila: questo carattere viene quindi estratto con una ε-transizione.

Quesito: Sia dato il linguaggio

$$L = \{a^n b^m c^k | k = |n - m|\}$$

Definire una grammatica context free che generi il linguaggio. Discutere se la grammatica risultante è ambigua.

Soluzione: *Una possibile grammatica è la seguente:*

$$\begin{array}{rcl} S & \rightarrow & S_1 | S_3 \\ S_1 & \rightarrow & aS_1b | S_2 \\ S_2 & \rightarrow & aS_2c | \varepsilon \\ S_3 & \rightarrow & S_4S_5 \\ S_4 & \rightarrow & aS_4b | \varepsilon \\ S_5 & \rightarrow & bS_5c | \varepsilon \end{array}$$

S_1 corrisponde al caso $n \geq m$, mentre S_3 al caso $m \geq n$. La grammatica in questo caso risulta ambigua, in quanto ad esempio la stringa $aabb$ può essere generata sia come $S \Rightarrow S_1 \Rightarrow aS_1b \Rightarrow aaS_1bb \Rightarrow aabb$ che come $S \Rightarrow S_3 \Rightarrow S_4S_5 \Rightarrow aS_4bS_5 \Rightarrow aaS_4bbS_5 \Rightarrow aabbS_5 \Rightarrow aabb$

Quesito: Si determini se i linguaggi

$$L = \{a^i b^j c^i | i, j \geq 1\}$$

e

$$L = \{a^i b^j c^k | i, j, k \geq 0\}$$

sono regolari.

Soluzione: *Il primo linguaggio non è regolare: per dimostrare ciò possiamo utilizzare il pumping lemma. Dato $n > 0$, consideriamo la stringa $\sigma = a^n bc^n$: qualunque decomposizione $\sigma = uvw$*

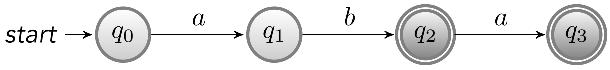
con $|uv| \le n$ e $|v| > 1$ fa sÃ¬ che $v = a^k$ per un qualche $0 < k < n$. Ne deriva che la stringa $\sigma' = uv^2w = a^{n+k}bc^n \notin L$, da cui la non regolarità di L.

Il secondo linguaggio è invece regolare: infatti può essere descritto dall'espressione regolare $a^*b^*c^*$.

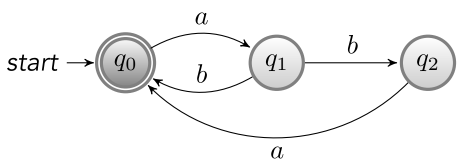
Quesito: Costruire un ASFD che riconosca il linguaggio descritto dall'espressione regolare $a((ab+aba)^*a)^*$

Soluzione: *Deriviamo un ASFND che riconosce il linguaggio in modo graduale.*

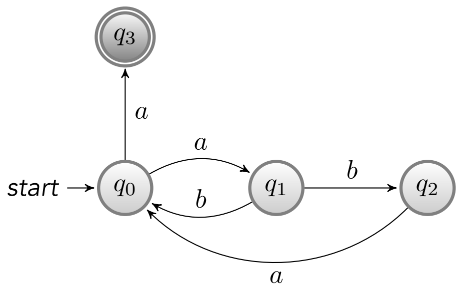
L'ASFND che riconosce $ab + aba$ è



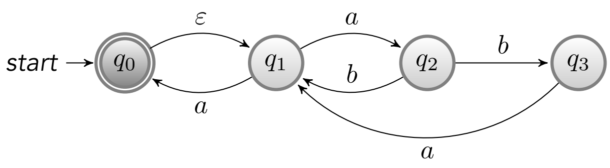
Da cui l'automa che riconosce $(ab + aba)^*$



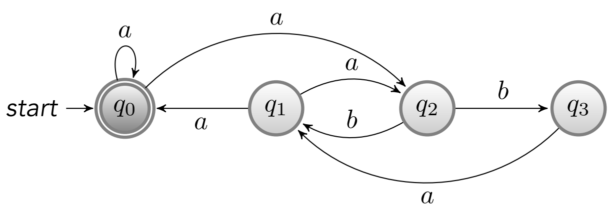
e quello che riconosce $(ab + aba)^*a$



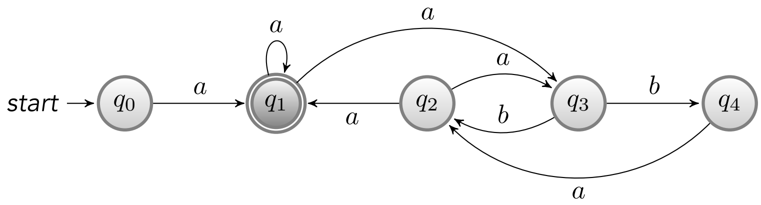
Il linguaggio descritto da $((ab + aba)^*a)^*$ è allora accettato da



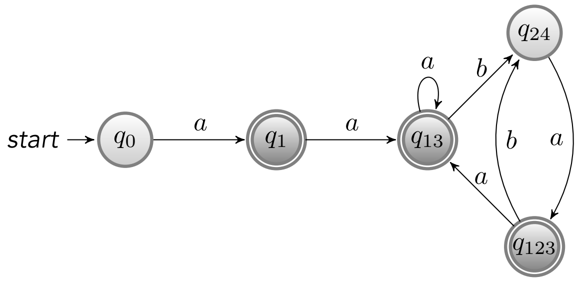
ed eliminando la ε-transizione considerando la ε chiusura,



Ne deriva che l'ASFND che riconosce il linguaggio è



Da cui l'ASFD equivalente



Quesito: Dimostrare che il seguente linguaggio.

$$L = \{w \in \{a, b\}^+ : \#_w(a) = 2\#_w(b)\}$$

è context free, dove $\#_w(x)$ indica il numero di occorrenze del carattere x nella stringa w

Soluzione: *Una possibile soluzione è quella di definire un PDA che accetta il linguaggio.*

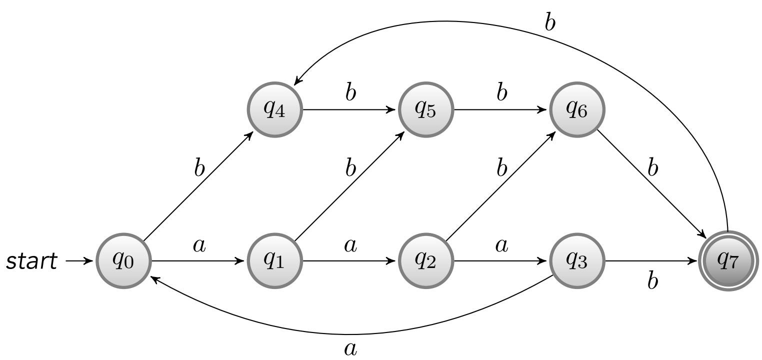
	(q_0, Z_0)	(q_0, X)	(q_0, Y)	(q_1, Z_0)	(q_1, Y)
a	(q_0, XXZ_0)	(q_0, XXX)	(q_1, ε)	-	-
b	(q_0, YZ_0)	(q_0, ε)	(q_0, YY)	-	-
ε	(q_0, ε)	-	-	(q_0, X)	(q_0, ε)

Quesito: Si consideri il linguaggio

$$L = \{w \in \{a, b\}^* | w = a^n b^m, n + m \text{ multiplo di 4}, m \geq 1\}$$

Si definiscano un ASFD che riconosce L e una grammatica regolare che lo genera.

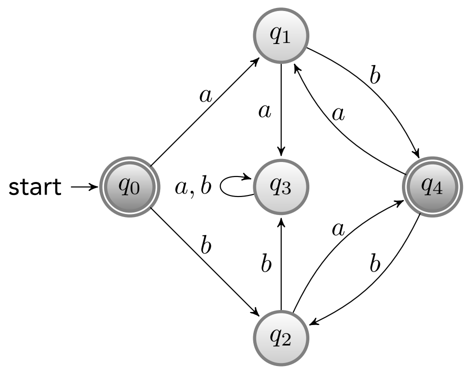
Soluzione: *Possibile soluzione*



La grammatica regolare deriva applicando la trasformazione nota, risultando:

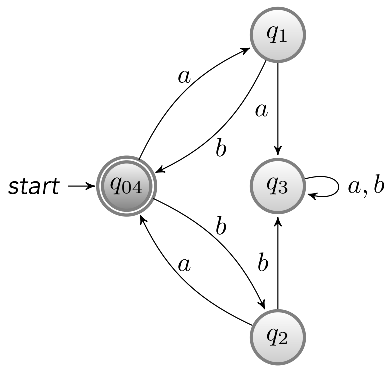
$$\begin{array}{rcl} S & \rightarrow & aA_1|bA_4 \\ A_1 & \rightarrow & aA_2|bA_5 \\ A_2 & \rightarrow & aA_3|bA_6 \\ A_3 & \rightarrow & bA_7|aS|b \\ A_4 & \rightarrow & bA_5 \\ A_5 & \rightarrow & bA_6 \\ A_6 & \rightarrow & bA_7|b \\ A_7 & \rightarrow & bA_4 \end{array}$$

Quesito: Dato l'ASFD seguente



si derivi una ASFD minimo equivalente.

Soluzione: *L'applicazione del metodo studiato indica che i soli stati indistinguibili sono q_0 e q_4. Ne deriva l'automa minimo seguente*



Quesito: Si dimostri che il linguaggio

$$L = \{a^*b^kc^*a^kb^*|k \geq 4\}$$

non è regolare

Soluzione: *Utilizziamo il pumping lemma per i linguaggi regolari. Dato l'intero n, consideriamo la stringa $b^{n+4}a^{n+4} \in L$: per ogni decomposizione uvw di $a^{n+4}b^{n+4}$ tale che $|uv| \leq n$, $|v| > 0$ si ha che $uv = b^m$, $m \leq n$, e quindi $v = b^r$, $r > 0$. Ne deriva che la stringa $uv^2w = b^{n+r+4}a^{n+4} \notin L$.*

Quesito: Si definisca una grammatica in CNF equivalente alla seguente

$$\begin{array}{rcl} S & \rightarrow & ABa \\ A & \rightarrow & aAbb|\varepsilon \\ B & \rightarrow & bB|A|b \end{array}$$

Soluzione: *A e B sono simboli annullabili, per cui l'eliminazione delle ε-produzioni fornisce*

$$\begin{array}{rcl} S & \rightarrow & ABa|Aa|Ba|a \\ A & \rightarrow & aAbb|abb \\ B & \rightarrow & bB|A|b \end{array}$$

L'eliminazione della produzione unitaria $B \rightarrow A$ fornisce

$$\begin{array}{rcl} S & \rightarrow & ABa|Aa|Ba|a \\ A & \rightarrow & aAbb|abb \\ B & \rightarrow & bB|aAbb|abb|b \end{array}$$

Tutti i simboli sono fecondi e raggiungibili, per cui non ci sono simboli inutili.

Una grammatica CNF equivalente è allora ottenuta dapprima eliminando i simboli terminali nelle parti destre delle produzioni non unitarie, ottenenendo

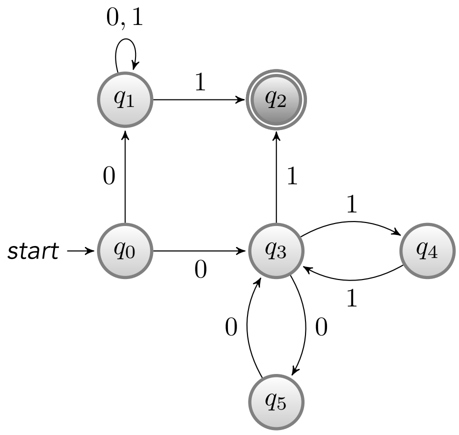
$$\begin{array}{rcl} S & \rightarrow & ABX|AX|BX|a \\ A & \rightarrow & XAYY|XYY \\ B & \rightarrow & YB|AXAYY|XYY|b \\ X & \rightarrow & a \\ Y & \rightarrow & b \end{array}$$

ed eliminando poi le produzioni con parti destre di lunghezza maggiore di 2, da cui

$$\begin{array}{rcl} S & \rightarrow & UX|AX|BX|a \\ A & \rightarrow & WZ|XZ \\ B & \rightarrow & YB|VZ|XZ|b \\ Z & \rightarrow & YY \\ W & \rightarrow & XA \\ U & \rightarrow & AB \\ V & \rightarrow & AW \\ X & \rightarrow & a \\ Y & \rightarrow & b \end{array}$$

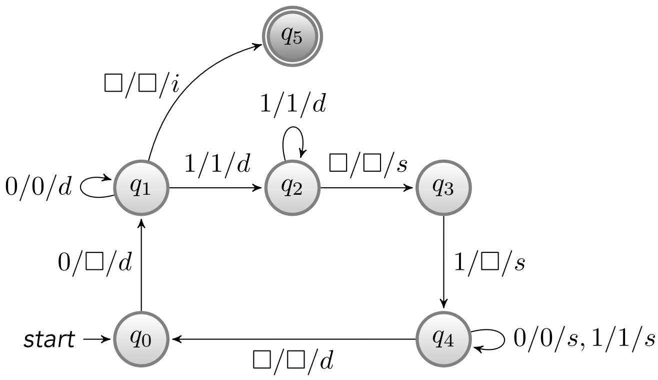
Quesito: Si costruisca un automa a stati finiti non deterministico che accetti il linguaggio L generato dall'espressione regolare $0(1+0)^*1 + 0(11+00)^*1$

Soluzione: *L è riconosciuto dall'automa non deterministico*



Quesito: Si definisca una macchina di Turing deterministica che accetti il linguaggio $L = \{0^i 1^j | i > j\}$

Soluzione: *Possibile soluzione*



La MdT elimina alternativamente un carattere 0 dall'inizio e un carattere 1 dalla fine della stringa: se rimane con soli 0 la stringa è accettata, altrimenti no.

La MdT inizia in q_0 sul primo carattere della stringa. Cancella il carattere se è 0 passando in q_1 e poi scorre la stringa fino a superare l'ultimo carattere, leggendo prima i caratteri 0 (in q_1) e poi i caratteri 1 (in q_2). Se legge soltanto caratteri 0, seguiti da una cella vuota, allora il numero di 0 era maggiore del numero di 1 e la stringa è accettata (stato q_5). Altrimenti, superato l'ultimo carattere 1, torna indietro per posizionarci la testina (stato q_3) ed eliminarlo passando in q_4 e scorrendo poi la stringa da destra verso sinistra. Quando viene superato il primo carattere, la testina viene spostata a destra per posizionarsi sul primo carattere (stato q_0).

Quesito: Si consideri la grammatica $\mathcal{G}$ con assioma S e produzioni

$$
\begin{array}{lcl}
S & \rightarrow & aAB|F \\
A & \rightarrow & aA|C \\
B & \rightarrow & bB|D \\
C & \rightarrow & cC|\varepsilon \\
D & \rightarrow & dD|\varepsilon \\
E & \rightarrow & eE|\varepsilon \\
F & \rightarrow & eF
\end{array}
$$

Costruire una grammatica in CNF equivalente a $\mathcal{G}$

Soluzione: *Eliminazione ε-produzioni: A, B, C, D, E sono simboli annullabili.*

$$
\begin{array}{lcl}
S & \to & aAB|aA|aB|a|F \\
A & \to & aA|a|C \\
B & \to & bB|b|D \\
C & \to & cC|c \\
D & \to & dD|d \\
E & \to & eE|e \\
F & \to & eF
\end{array}
$$

Eliminazione produzioni unitarie.

$$
\begin{array}{lcl}
S & \to & aAB|aA|aB|a|eF \\
A & \to & aA|a|cC|c \\
B & \to & bB|b|dD|d \\
C & \to & cC|c \\
D & \to & dD|d \\
E & \to & eE|e \\
F & \to & eF
\end{array}
$$

Eliminazione simboli inutili. Il simbolo F risulta raggiungibile ma non fecondo, il simbolo E è non raggiungibile.

$$
\begin{array}{lcl}
S & \to & aAB|aA|aB|a \\
A & \to & aA|a|cC|c \\
B & \to & bB|b|dD|d \\
C & \to & cC|c \\
D & \to & dD|d
\end{array}
$$

Forma normale di Chomsky.

$$
\begin{array}{lcl}
S & \to & VU|VA|VB|a \\
A & \to & VA|a|YC|c \\
B & \to & XB|b|ZD|d \\
C & \to & YC|c \\
D & \to & ZD|d \\
U & \to & AB \\
V & \to & a \\
X & \to & b \\
Y & \to & c \\
Z & \to & d
\end{array}
$$

Quesito: Si definisca una grammatica CF che generi il linguaggio $L = \{a^n b^m c^k | k = |n - m|\}$. (Suggerimento: si considerino separatamente i casi $n \geq m$ e $m > n$.)

Soluzione:

$$
\begin{array}{rcl}
S_1 & \rightarrow & S_1|S_2 \\
S_1 & \rightarrow & aS_1c|ac|X \\
X & \rightarrow & aXb|ab \\
S_2 & \rightarrow & YZ \\
Y & \rightarrow & aYb|ab \\
Z & \rightarrow & bZc|bc
\end{array}
$$

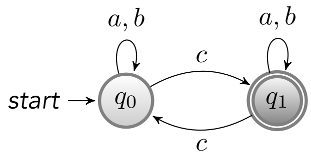
Quesito: Si costruisca un automa (deterministico o non deterministico) che riconosca il linguaggio $L \subseteq \{a,b,c\}^*$ definito come segue

$L = \{w|w$ contiene un numero dispari di c oppure non contiene occorrenze della sottostringa $aba\}$

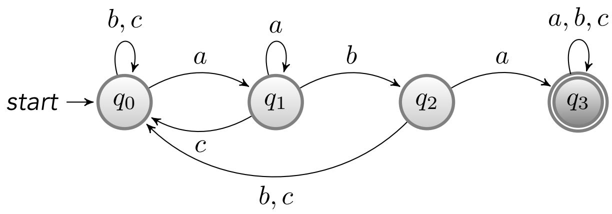
Soluzione: *Si osservi che possiamo scrivere $L = L_1 \cup \overline{L}_2$, con*

$L_1 = \{w|w$ *contiene un numero dispari di* $c\}$
$L_2 = \{w|w$ *contiene la sottostringa* $aba\}$

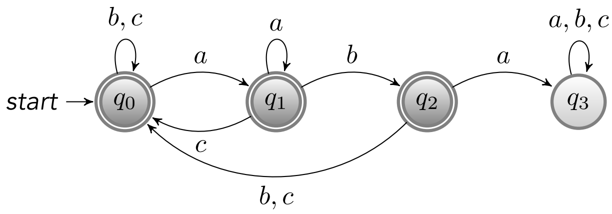
L_1 *è riconosciuto dall'automa*



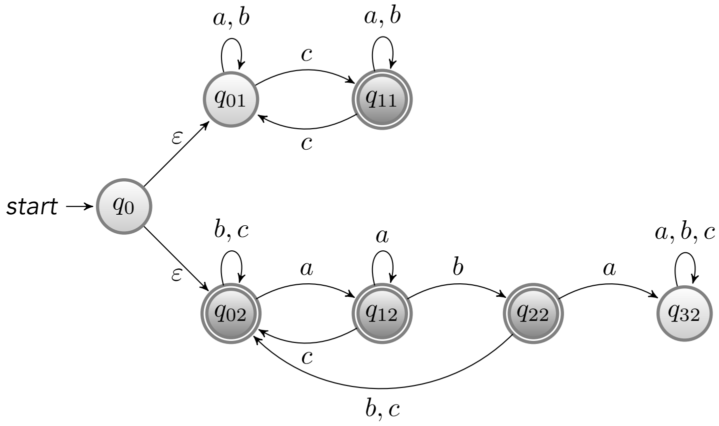
mentre L_2 è riconosciuto da



di conseguenza, $\overline{L}_2$ è riconosciuto da



e infine L è riconosciuto da



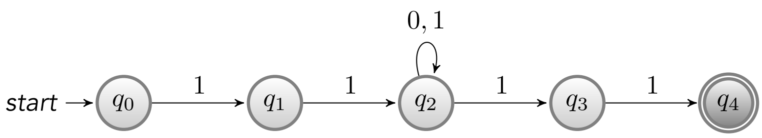
Quesito: Sia dato il linguaggio $L \subseteq \{a, b, c\}^*$ tale che $w \in L$ se e solo se $\#_w(a) = \#_w(c)$, dove $\#_w(x)$ indica il numero di occorrenze di $x \in \{a, b, c\}$ in w. Tale linguaggio è context free? Motivare la propria risposta o mediante applicazione del pumping lemma o fornendo una grammatica CF che lo generi .

Soluzione: *Il linguaggio è context-free: per motivare tale risposta definiamo una grammatica CF che lo generi*

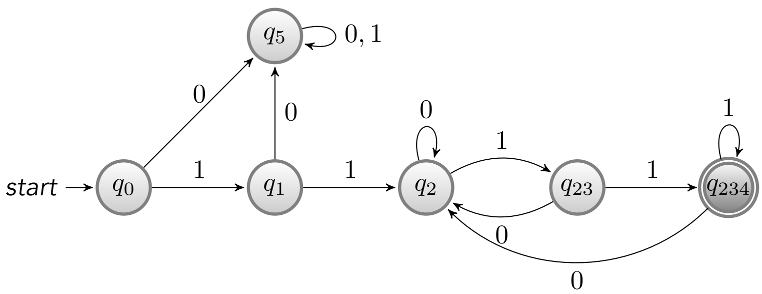
$$S \rightarrow \varepsilon | bS | Sb | aScS | cSaS$$

Quesito: Definire un automa deterministico minimo (nel numero di stati) che riconosca il linguaggio $11(0+1)^*11$

Soluzione: *Automa non deterministico che accetta il linguaggio*



Automa deterministico totale equivalente



L'automa risulta già minimo, fornendo la matrice di equivalenza seguente (per ogni locazione il carattere che rende i due stati non equivalenti).

	0	1	2	23	234	5
0						
1	1					
2	1	1				
23	1	1	1			
234	ε	ε	ε	ε		
5	1	1	1	1	ε	

Quesito: Si definisca una grammatica context free che generi il linguaggio

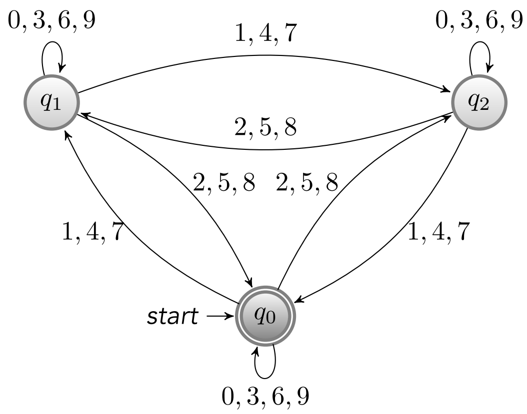
$$L = \{a^m b^n + a^r b^s a^t | 1 \leq m \leq n \leq 3m; s \geq 1, 1 \leq r \leq t \leq 2r\}$$

Soluzione:

$$\begin{array}{rcl} S & \rightarrow & S_1 | S_2 \\ S_1 & \rightarrow & ab|abb|abbb|aS_1b|aS_1bb|aS_1bbb \\ S_2 & \rightarrow & aBa|aBaa|aS_2a|aS_2aa \\ B & \rightarrow & bB|b \end{array}$$

Quesito: Come noto, un numero intero espresso in base 10 è divisibile per 3 se lo è la somma delle sue cifre. Si consideri il linguaggio L comprendente tutte e sole le sequenze di cifre decimali corrispondenti a interi non negativi multipli di 3. Determinare, dimostrandolo, se L è regolare o meno.

Soluzione: *L è regolare e riconosciuto, ad esempio, dall'ASFD seguente, che tiene traccia del resto della divisione per 3 della somma delle cifre decimali lette.*



Quesito: Si consideri il seguente linguaggio su $\Sigma = \{0, 1, \sharp, \varepsilon, +, \cdot, *, (,)\}$

$$L = \{r\sharp s | r, s \text{ sono espressioni regolari su } \{0,1\}, L(r) \subseteq L(s)\}$$

Dimostrare che L è decidibile, definendo (in modo informale) un metodo per il suo riconoscimento.

Soluzione: *Si noti che $L(r) \subseteq L(s)$ è equivalente a $L(r) \cap \overline{L}(s) = \emptyset$ e quindi a $\overline{\overline{L}(r) \cup L(s)} = \emptyset$: date r e s è allora possibile derivare due ASFD A_r, A_s che riconoscono, rispettivamente, $L(r)$ e*

$L(s)$. Per le proprietà di chiusura dei linguaggi regolari, da A_r è quindi possibile derivare l'automa $\overline{A}_r$ che riconosce $\overline{L}(r)$. Infine, è possibile comporre $\overline{A}_r$ e A_s costruendo l'automa A che riconosce $\overline{L}(r) \cup L(s)$ e da questo l'automa $\overline{A}$ che riconosce $\overline{\overline{L}(r) \cup L(s)}$: evidentemente, $r\sharp s \in L$ se e solo se $L(\overline{A}) = \emptyset$, condizione decidibile.

Quesito: Si definisca una grammatica in CNF che generi il linguaggio $L = \{a^n b^m | n + m > 0, n \neq m\}$.

Soluzione: *Una grammatica che genera L è ad esempio*

$$\begin{array}{rcl} S & \rightarrow & aSb|aA|aB \\ A & \rightarrow & aA|\varepsilon \\ B & \rightarrow & bB|\varepsilon \end{array}$$

Eliminazione ε-produzioni.

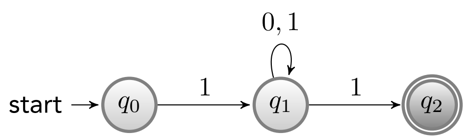
$$\begin{array}{rcl} S & \rightarrow & aSb|aA|aB|a|b \\ A & \rightarrow & aA|a \\ B & \rightarrow & bB|b \end{array}$$

Come si può osservare, non ci sono produzioni unitarie né simboli inutili.

Forma normale di Chomsky.

$$\begin{array}{rcl} S & \rightarrow & XV|UA|VB|a|b \\ X & \rightarrow & US \\ A & \rightarrow & UA|a \\ B & \rightarrow & VB|b \\ U & \rightarrow & a \\ V & \rightarrow & b \end{array}$$

Quesito: Dato il seguente ASFND A



si derivi un grammatica regolare $\mathcal{G}$ tale che $L(\mathcal{G}) = L(A)$

Soluzione: *Applicando la costruzione nota, si ottiene la grammatica*

$$\begin{array}{rcl} S_0 & \rightarrow & 1S_1 \\ S_1 & \rightarrow & 0S_1|1S_1|1 \end{array}$$

Quesito: Si consideri il linguaggio $L = \{a^r b^s c^t | t = r - s\}$. Dimostrare che questo linguaggio non è regolare.

Soluzione: *Si ricorda che, per il pumping lemma sui linguaggi regolari, se L fosse regolare allora esisterebbe una costante n tale che ogni una stringa $\sigma \in L$ con $|\sigma| > n$ può essere decomposta nella forma $\sigma = uvw$ (con $|uv| \leq n$, $|v| \geq 1$) in modo tale che $uv^i w \in L$ per ogni $i \geq 0$.*

È sufficiente quindi, per mostrare che L non è regolare, individuare, dato n, una stringa $\sigma \in L$ con $|\sigma| > n$ per la quale mostrare che $uv^iw \notin L$ per qualche $i \geq 0$, per ogni decomposizione $\sigma = uvw$.

Si consideri allora una qualunque stringa $\sigma = a^n b^m c^{n-m} \in L$ (con $0 \leq m \leq n$). Evidentemente ogni decomposizione $a^n b^m c^{n-m} = uvw$ con $|uv| \leq n$ e $|v| \geq 1$ sarà tale che $uv = a^h$ per qualche h e quindi $v = a^k$ con $k \geq 1$. Ma allora la stringa $uv^2w = a^{n+k} b^m c^{n-m} \notin L$, in quanto $r = n + k$, $s = m$, $t = n - m$ e $t \neq r - s$.

Quesito: Si definisca una grammatica context free che generi il linguaggio $L = \{a^r b^s c^t | t = r - s\}$.

Soluzione: *Osservando che $r = s + t$, una possibile grammatica che generi L è ad esempio:*

$$\begin{array}{lcl} S & \to & aSc|U \\ U & \to & aUb|\varepsilon \end{array}$$

Quesito: Si definisca un automa a pila (eventualmente non deterministico) che accetti il linguaggio $L = \{a^r b^s c^t | t = r - s\}$.

Soluzione: *L'automa può essere derivato portando dapprima la grammatica precedente in forma normale di Greibach, applicando poi la costruzione standard di un NPDA che riconosca lo stesso linguaggio. La presenza di ε in L può essere non considerata nella costruzione dell'automa, introducendo poi la possibilità per l'automa stesso di riconoscere la stringa vuota.*

La grammatica precedente può essere portata in forma ridotta come

$$\begin{array}{lcl} S & \to & aSc|aUb|ab|\varepsilon \\ U & \to & aUb|ab \end{array}$$

e quindi in CNF per generare $L - \{\varepsilon\}$ come

$$\begin{array}{lcl} S & \to & XC|YB|AB \\ U & \to & YB|AB \\ X & \to & AS \\ Y & \to & AU \\ A & \to & a \\ B & \to & b \\ C & \to & c \end{array}$$

e da questa la grammatica in GNF per $L - \{\varepsilon\}$,

$$\begin{array}{lcl} S & \to & aSC|aUB|aB \\ U & \to & aUB|aB \\ X & \to & aS \\ Y & \to & aU \\ A & \to & a \\ B & \to & b \\ C & \to & c \end{array}$$

Da questa deriva il seguente NPDA che riconosce $L - \{\varepsilon\}$ per pila vuota:

$$
\begin{aligned}
\Sigma &= \{a, b, c\} \\
\Gamma &= \{S, U, X, Y, A, B, C\} \\
Q &= \{q_0\} \\
Z_0 &= S
\end{aligned}
$$

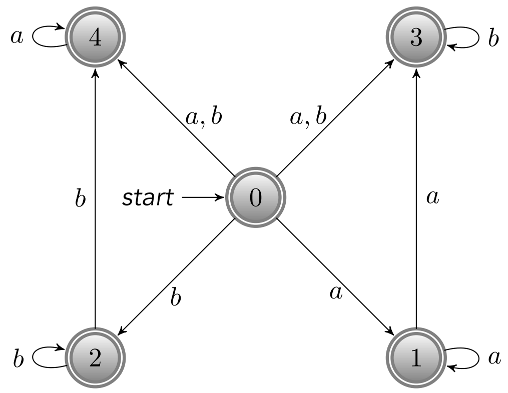
con la funzione di transizione (lo stato q_0 è sottinteso)

	S	U	X	Y	A	B	C
a	SC UB B	UB B	S	U	ε		
b						ε	
c							ε

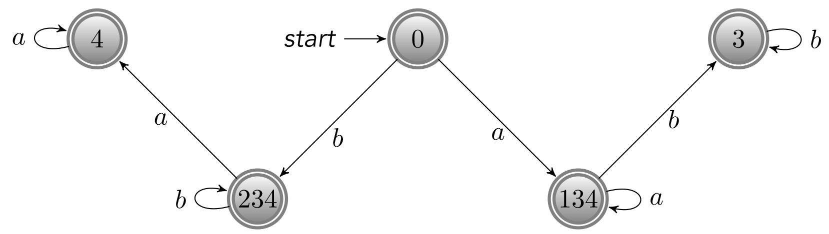
Per tener conto di $\varepsilon \in L$ si può applicare la procedura standard, introducendo uno stato iniziale q_0' e una coppia di transizioni $\delta(q_0', \varepsilon, S) = \{(q_0', \varepsilon), (q_0, S)\}$.

Quesito: Data l'espressione regolare $a^*b^* + b^*a^*$, costruire una automa a stati finiti deterministico che riconosca il linguaggio descritto da essa.

Soluzione: *Possiamo definire un ASFND che riconosce il linguaggio.*



a da questo, mediante la procedura standard, l'ASFD equivalente



Quesito: Sia dato il linguaggio

$$L = \{w \in \{a, b\}^* | w \text{ non è della forma } vv\}$$

Mostrare se L è regolare o meno.

Soluzione: *Possiamo utilizzare il Pumping lemma per mostrare facilmente che*

$$\overline{L} = \{w \in \{a,b\}^* | w \text{ è della forma } vv\}$$

non è regolare.

Per la chiusura dei linguaggi regolari rispetto al complemento, neanche L è regolare.

Quesito: Definire un automa a pila che accetti il linguaggio

$$L = \{a^n b^m | 1 \le n \le m\}$$

per pila vuota.

Soluzione: *Un possibile PDA legge la sequenza iniziale di a e ponendo sulla pila un simbolo A per ogni simbolo letto. L'automa cambia stato per leggere la sequenza di b, eliminando i caratteri A dalla pila. Se si raggiunge il fondo della pila (il simbolo Z_0) la stringa va accettata, completando la lettura degli eventuali b mancanti ed eliminando poi Z_0.*

	(q_0, Z_0)	(q_0, A)	(q_1, Z_0)	(q_1, A)
a	(q_0, AZ_0)	(q_0, AA)	-	-
b	-	(q_1, ε)	(q_1, Z_0)	(q_1, ε)
ε	-	-	(q_1, ε)	-

Quesito: Si definisca una grammatica in CNF equivalente alla seguente

$$\begin{array}{lcl} S & \to & 0A0|1B1|BB \\ A & \to & C \\ B & \to & S|A \\ C & \to & S|\varepsilon \end{array}$$

Soluzione: *Eliminazione ε-produzioni.*

- *Simboli annullabili: S, A, B, C*
- *Grammatica risultante*

$$\begin{array}{lcl} S & \to & 0A0|1B1|BB|00|11|B \\ A & \to & C \\ B & \to & S|A \\ C & \to & S \end{array}$$

Eliminazione produzioni unitarie.

- *Risulta: $U(S) = \{A, B, C\}$, $U(A) = \{S, B, C\}$, $U(B) = \{S, A, C\}$, $U(C) = \{S, A, B\}$*
- *Grammatica risultante*

$$\begin{array}{lcl} S & \to & 0A0|1B1|BB|00|11 \\ A & \to & 0A0|1B1|BB|00|11 \\ B & \to & 0A0|1B1|BB|00|11 \\ C & \to & 0A0|1B1|BB|00|11 \end{array}$$

Eliminazione simboli inutili.

- *Tutti in non terminali sono fecondi. C risulta non raggiungibile.*
- *Grammatica risultante*
-

$$
\begin{array}{lcl}
S & \rightarrow & 0A0|1B1|BB|00|11 \\
A & \rightarrow & 0A0|1B1|BB|00|11 \\
B & \rightarrow & 0A0|1B1|BB|00|11
\end{array}
$$

Trasformazione in CNF:

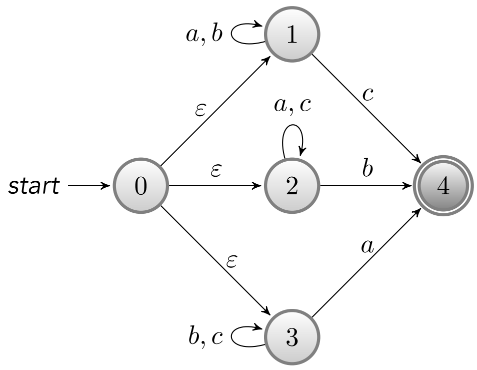
$$
\begin{array}{lcl}
S & \rightarrow & ZX|UY|BB|ZZ|UU \\
A & \rightarrow & ZX|UY|BB|ZZ|UU \\
B & \rightarrow & ZX|UY|BB|ZZ|UU \\
X & \rightarrow & AZ \\
Y & \rightarrow & BU \\
Z & \rightarrow & 0 \\
U & \rightarrow & 1
\end{array}
$$

Quesito: Sia dato il linguaggio

$$L = \{w \in \{a, b, c\}^+ | \text{ l'ultimo carattere in } w \text{ non è comparso prima}\}$$

Si definisca un automa a stati finiti che accetti L.

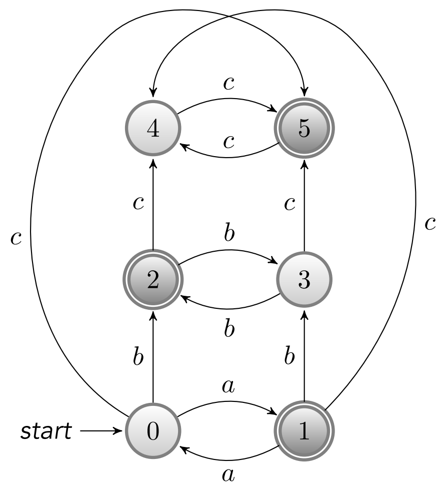
Soluzione:



Quesito: Si definisca una grammatica di tipo 3 che generi il seguente linguaggio

$$L = \{a^n b^m c^k | n + m + k \text{ dispari}\}$$

Soluzione: *Definiamo un ASF che riconosce L*



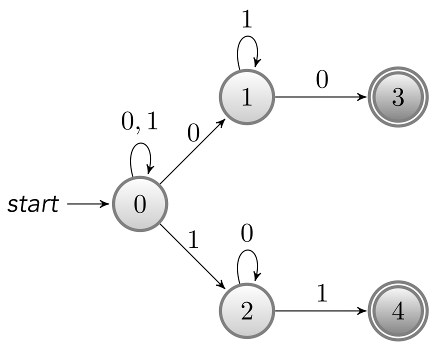
da cui deriva la grammatica seguente, con assioma A_0

$$\begin{array}{lcl} A_0 & \rightarrow & aA_1|bA_2|cA_5|a|b|c \\ A_1 & \rightarrow & aA_0|bA_3|cA_4 \\ A_2 & \rightarrow & bA_3|cA_4 \\ A_3 & \rightarrow & bA_2|cA_5|b|c \\ A_4 & \rightarrow & cA_5|c \\ A_5 & \rightarrow & cA_4 \end{array}$$

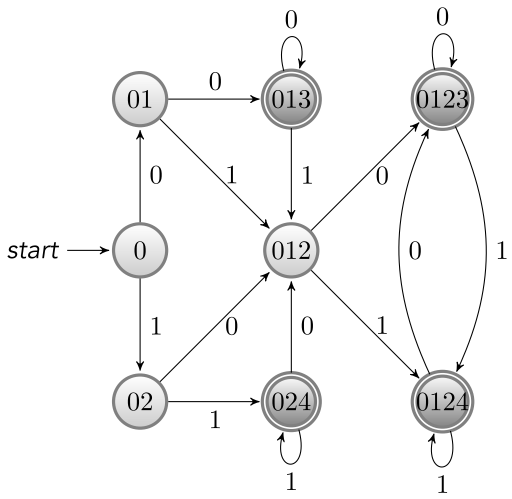
Quesito: Definire un ASFD che riconosca il linguaggio

$$L = \{w \in \{0,1\}^+ | \text{ l'ultimo carattere di } w \text{ è già apparso nella stringa}\}$$

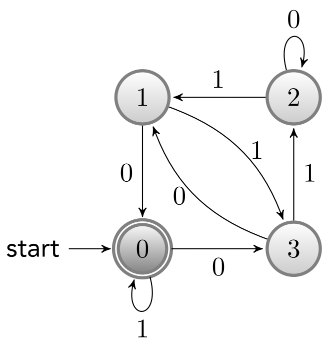
Soluzione: *È utile definire inizialmente un ASFND che accetti L, come ad esempio*



L'AFD cercato può essere derivato dal precedente, ottenendo



Quesito: Definire una espressione regolare che descriva il linguaggio riconosciuto dal seguente ASFD



Soluzione: *Una possibile soluzione prevede la derivazione della grammatica regolare equivalente*

$$\begin{array}{lcl} A_0 & \rightarrow & 1A_0|0A_3|1 \\ A_1 & \rightarrow & 0A_0|1A_3|0 \\ A_2 & \rightarrow & 0A_2|1A_1 \\ A_3 & \rightarrow & 0A_1|1A_2 \end{array}$$

E da questa, manipolando il sistema di epressioni corrispondente, l'espressione regolare cercata.

$$\begin{cases} A_0 = 1A_0 + 0A_3 + 1 \\ A_1 = 0A_0 + 1A_3 + 0 \\ A_2 = 0A_2 + 1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

$$\begin{cases} A_0 = 1A_0 + 0(1A_1 + 1A_2) + 1 \\ A_1 = 0A_0 + 1(1A_1 + 1A_2) + 0 \\ A_2 = 0A_2 + 1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

$$\begin{cases} A_0 = 1A_0 + 0(1+10^*1)A_1 + 1 \\ A_1 = 0A_0 + 1(1+10^*1)A_1 + 0 \\ A_2 = 0^*1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

$$\begin{cases} A_0 = 1A_0 + 0(1+10^*1)(1(1+10^*1))^*(0A_0+0) + 1 \\ A_1 = (1(1+10^*1))^*(0A_0+0) \\ A_2 = 0^*1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

$$\begin{cases} A_0 = (1+0(1+10^*1)(1(1+10^*1))^*0)A_0 + 0(1+10^*1)(1(1+10^*1))^*0 + 1 \\ A_1 = (1(1+10^*1))^*(0A_0+0) \\ A_2 = 0^*1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

$$\begin{cases} A_0 = (1+0(1+10^*1)(1(1+10^*1))^*0)^*(0(1+10^*1)(1(1+10^*1))^*0 + 1) \\ A_1 = (1(1+10^*1))^*(0A_0+0) \\ A_2 = 0^*1A_1 \\ A_3 = 1A_1 + 1A_2 \end{cases}$$

Quindi il linguaggio è descritto dall'espressione regolare

$$(1+0(1+10^*1)(1(1+10^*1))^*0)^*(0(1+10^*1)(1(1+10^*1))^*0+1)$$

Quesito: Mostrare che il linguaggio

$$L = \{ww|w \in \{a,b\}^*\}$$

non è regolare.

Soluzione: *È sufficiente utilizzare il pumping lemma per i linguaggi regolari.*

Dato n, scegliamo ad esempio la stringa $\sigma = a^nb^na^nb^n$. Qualunque decomposizione $\sigma = uvw$ che soddisfi i vincoli posti dal pumping lemma ($|uv| \leq n$, $|v| > 0$) dovrà essere tale che $uv = a^k$ per qualche $k \leq n$. Di conseguenza, $v = a^h$ per $1 \leq h \leq k$ e, considerando la stringa $\sigma' = uv^2w$, si può osservare che $\sigma' = a^{n+h}b^na^nb^n \notin L$.

Quesito: Definire un automa a pila che accetti il linguaggio

$$L = \{w \in \{a,b\}^+ | \#_a(w) \geq \#_b(w)\}$$

Dove $\#_c(x)$ indica il numero di occorrenze del carattere c nella stringa x.

Soluzione: *L'automa non deve fare altro che mantenere traccia, sulla pila, della differenza tra il numero di caratteri a e il numero di caratteri b letti fino a ora (o vice versa, a seconda che siano stati letti più a o più b). La stringa è accettata se al termine della sua lettura la pila è vuota o contiene tutti simboli A. Per accettare per pila vuota l'automa prevede che in qualunque istante in cui il numero di a lette è almeno pari al numero di b possa entrare in uno stato q_1 di svuotamento della pila.*

	(q_0, Z_0)	(q_0, A)	(q_0, B)	(q_1, Z_0)	(q_1, A)
a	(q_0, AZ_0)	(q_0, AA)	(q_0, ε)	-	-
b	(q_0, BZ_0)	(q_0, ε)	(q_0, BB)	-	-
ε	(q_1, ε)	(q_1, ε)	-	(q_1, ε)	(q_1, ε)

Quesito: Sia L il linguaggio generato dalla seguente grammatica context free

$$S \rightarrow \varepsilon|0S1S|1S0S$$

derivare una grammatica in Forma Normale di Greibach che generi $L - \{\varepsilon\}$.

Soluzione: *Il primo passo prevede la derivazione della grammatica in forma ridotta equivalente.*
Eliminazione delle ε produzioni:

$$S \rightarrow 0S1S|1S0S|01S|0S1|01|1S0|10S|10$$

Non ci sono produzioni unitarie o simboli inutili.
Forma normale di Chomsky:

$$\begin{array}{rcl} S & \rightarrow & XY|YX|ZY|XU|ZU|YZ|UX|UZ \\ X & \rightarrow & ZS \\ Y & \rightarrow & US \\ Z & \rightarrow & 0 \\ U & \rightarrow & 1 \end{array}$$

Forma normale di Greibach:

- *dopo la prima fase*

$$\begin{array}{rcl} S & \rightarrow & XY|YX|ZY|XU|ZU|YZ|UX|UZ \\ X & \rightarrow & ZS \\ Y & \rightarrow & US \\ Z & \rightarrow & 0 \\ U & \rightarrow & 1 \end{array}$$

- *dopo la seconda fase*

$$\begin{array}{rcl} S & \rightarrow & 0SY|1SX|0Y|1U|0U|0Z|1X|1Z \\ X & \rightarrow & 0S \\ Y & \rightarrow & 1S \\ Z & \rightarrow & 0 \\ U & \rightarrow & 1 \end{array}$$

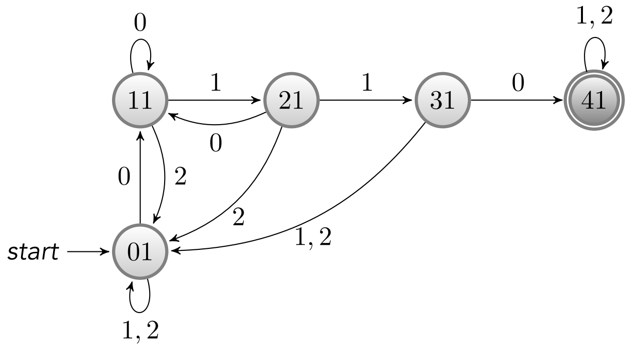
Quesito: Definire un ASFND che accetti il seguente linguaggio

$$L = \{w \in \{0,1,2\}^+\}$$

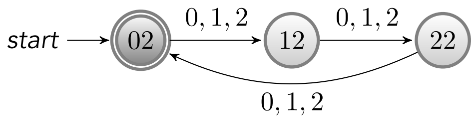
dove:

- 0110 compare in w e inoltre:
  - $|w|$ è un multiplo di 3 oppure
  - 22 non compare in w

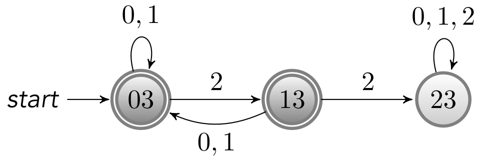
Soluzione: *Automa $\mathcal{A}_1$, riconosce le stringhe che includono 0110 come sottostringa*



Automa $\mathcal{A}_2$, riconosce le stringhe di lunghezza pari a un multiplo di 3



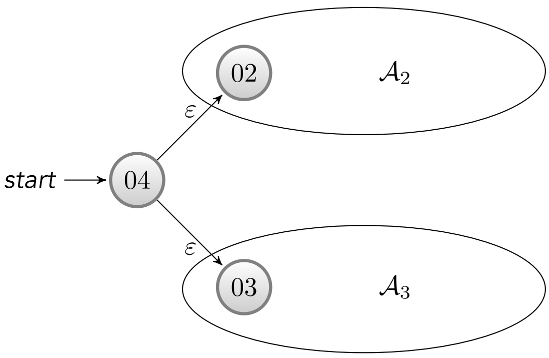
Automa $\mathcal{A}_3$, riconosce le stringhe che non contengono 22 come sottostringa



L'ASFND richiesto può essere ottenuto a partire da questi nel modo seguente, tenendo conto che

$$L = \overline{\overline{L(\mathcal{A}_1)} \cup \overline{(L(\mathcal{A}_2) \cup L(\mathcal{A}_3))}}$$

1. $L(\mathcal{A}_2) \cup L(\mathcal{A}_3)$ *viene accettato dall'ASFND* $\mathcal{A}_4$ *ottenuto applicando la nota composizione per l'unione di due linguaggi*



2. $\overline{L(\mathcal{A}_2) \cup L(\mathcal{A}_3)}$ *viene riconosciuto dall'automa* $\mathcal{A}_5$ *ottenuto a partire dall'ASFD equivalente a* $\mathcal{A}_4$*, invertendo stati finali e non finali*
3. $\overline{L(\mathcal{A}_1)}$ *viene riconosciuto dall'automa* $\mathcal{A}_6$ *ottenuto invertendo stati finali e non finali di* $\mathcal{A}_1$
4. $\overline{L(\mathcal{A}_1)} \cup \overline{(L(\mathcal{A}_2) \cup L(\mathcal{A}_3))}$ *viene accettato dall'ASFND* $\mathcal{A}_7$ *ottenuto applicando la stessa composizione precedente a* $\mathcal{A}_5$ *e* $\mathcal{A}_6$
5. *L'ASFND voluto può essere ottenuto da* $\mathcal{A}_7$ *derivandone l'ASFD equivalente e scambiando stati finali e non.*

Quesito: Mostrare se il seguente linguaggio è o meno context free:

$$L = \{w_1w_2w_3 : w_1 \in \{a,b\}^+, w_2 \in \{c,d\}^+, w_3 \in \{e,f\}^+, |w_1| = |w_2| = |w_3|\}$$

Soluzione: *Il linguaggio non è context-free. Per dimostrare cioò utilizziamo il pumping lemma per i linguaggi di tipo* 2.

Dato n*, consideriamo la stringa* $\sigma = a^nc^ne^n$*. Se consideriamo le decomposizioni* $\sigma = uvwxy$ *con* $|vwx| \leq n$ e $|vx| \geq 1$ *si hanno due casi possibili:*

- *sia* v *che* x *sono sequenze di stessi caratteri (ad esempio* $v = a^k$ e x=c^h*): si osservi che in tal caso uno dei tre caratteri che compaiono in* σ *non compare in* vx*. Di conseguenza la stringa* $\sigma' = uv^2wx^2y$ *non presenta lo stesso numero di* a*,* c *ed* e*, e quindi non appartiene al linguaggio. Si osservi che come caso particolare si ha* $v = \varepsilon$ *o* $x = \varepsilon$*: la conclusione deriva anche in questo caso.*
- *almeno una tra* v *e* x *non è una sequenza di stessi caratteri (ad esempio,* $v = a^hc^k$*): in tal caso,* $v^2 = a^hc^ka^hc^k$ e $\sigma' = uv^2wx^2y$ *non appartiene al linguaggio.*

In conclusione, dato che per ogni decomposizione possibile di σ*, che soddisfi le condizioni del pumping lemma, si ha* $\sigma \notin L$*, concludiamo che* L *non è context free.*

Quesito: Sia dato un automa a stati finiti deterministico $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ con

1. $\Sigma = \{a, b\}$
2. $Q = \{0, 1, 2, 3, 4, 5, 6\}$
3. $q_0 = 0$
4. $F = \{2, 3, 5, 6\}$

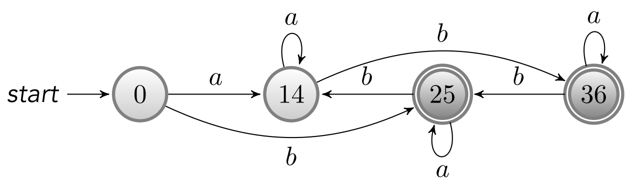
e δ descritta dalla seguente tabella di transizione

	0	1	2	3	4	5	6
a	1	1	2	6	4	5	3
b	5	6	4	5	3	4	2

Derivare un automa $\mathcal{A}'$ equivalente ad $\mathcal{A}$ con minimo numero di stati

Soluzione: *Applicando la procedura nota per l'individuazione di coppie di stati equivalenti, derivano le seguenti classi di equivalenza:* $\{0\}$, $\{1, 4\}$, $\{2, 5\}$, $\{3, 6\}$.

L'automa minimo sarà:



Quesito: Definire una grammatica in forma normale di Greibach che generi il linguaggio

$$L = \{a^mb^n | m \neq n\}$$

Soluzione: *Il linguaggio può essere generato dalla grammatica*

$$\begin{array}{rcl} S & \to & aSb|A|B \\ A & \to & aA|a \\ B & \to & bB|b \end{array}$$

La grammatica non ha ε-produzioni. L'eliminazione delle produzione unitarie fornisce:

$$\begin{array}{rcl} S & \to & aSb|aA|bB|a|b \\ A & \to & aA|a \\ B & \to & bB|b \end{array}$$

Dato non ci sono simboli inutili, la grammatica è in forma ridotta. In forma normale di Chomsky,

$$\begin{array}{rcl} S & \to & WY|XA|YB|a|b \\ A & \to & XA|a \\ B & \to & YB|b \\ W & \to & XS \\ X & \to & a \\ Y & \to & b \end{array}$$

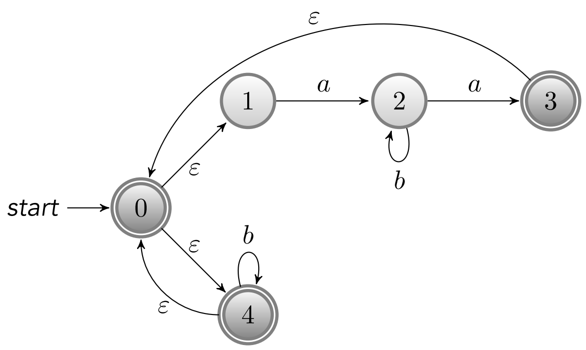
La grammatica in forma normale di Greibach deriva immediatamente se consideriamo l'ordinamento S, A, B, W, X, Y dei non terminali, e risulta essere:

$$\begin{array}{rcl} S & \to & aSY|aA|bB|a|b \\ A & \to & aA|a \\ B & \to & bB|b \\ W & \to & aS \\ X & \to & a \\ Y & \to & b \end{array}$$

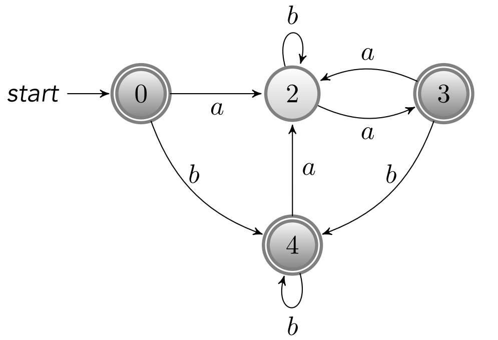
Quesito: Derivare un ASFD che riconosca il linguaggio descritto dall'espressione regolare

$$(ab^*a + b^*)^*$$

Soluzione: *Per composizione, possiamo derivare l'ASFND con ε-transizioni che accetta il linguaggio*



Eliminando le ε-transizioni, otteniamo il seguente ASFND, che risulta in effetti deterministico



Quesito: Sia dato il linguaggio $L = \{(ab)^i(cb)^j(ab)^k | i, j, k > 0; k = 2i\}$. L è regolare? Dimostrare la risposta data.

Soluzione: *Il linguaggio non è regolare. Si può dimostrare ciò utilizzando il pumping lemma.*

Bob: *sceglie n*

Alice: *sceglie la stringa $\sigma = (ab)^n cb(ab)^{2n}$*

Bob: *sceglie uv, prefisso di σ di lunghezza al più n. Necessariamente, quindi, uv è sottostringa di $(ab)^n$. Due casi sono possibili:*

1. *$|v|$ è dispari, per cui inizia e termina per lo stesso carattere, ad es. $v = bz_1b$, con $z_1 = (ab)^r a, r \geq 0$*
2. *$|v|$ è pari, per cui inizia e termina con caratteri diversi, ad es. $v = az_2b$, con $z_2 = (ba)^r, r \geq 0$*

Alice: *pone $i = 2$ e:*

1. *se $|v|$ è dispari, ottiene una stringa in cui compaiono, nella prima parte, due caratteri successivi uguali, ad es. $uvvw = ubz_1bbz_1bw \notin L$,*
2. *se $|v|$ è pari, ottiene una stringa $(ab)^{n+|v|/2}cb(ab)^{2n} \notin L$*

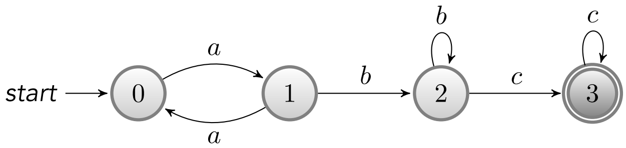
Quesito: Dimostrare che il seguente linguaggio è regolare $L = \{a^k b^j c^i | i, j, k > 0\}$ dove k è dispari e $i > 2$, oppure j è dispari e $i \leq 3$.

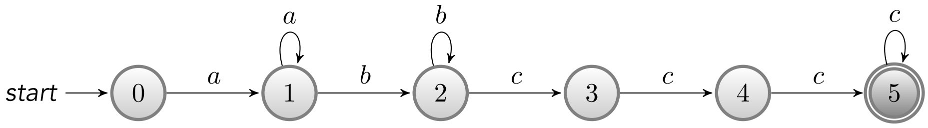
Soluzione: *Si considerino i linguaggi seguenti:*

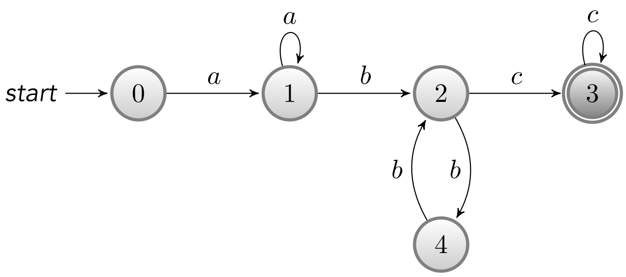
- $L_1 = \{a^k b^j c^i | i, j, k > 0, k$ *dispari*$\}$

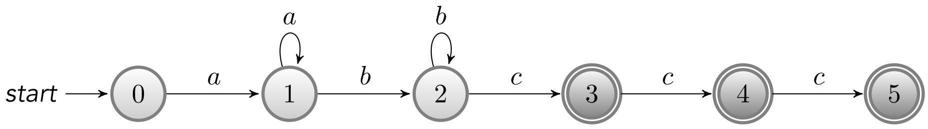
- $L_2 = \{a^k b^j c^i | j, k > 0, i > 2\}$
- $L_3 = \{a^k b^j c^i | i, j, k > 0, j$ *dispari*$\}$
- $L_4 = \{a^k b^j c^i | i, j, k > 0, i \leq 3\}$

Chiaramente, $L = (L_1 \cap L_2) \cup (L_3 \cap L_4)$. Inoltre L_1, L_2, L_3, L_4 possono essere mostrati regolari derivando in modo immediato ASF che li riconoscono:









L risulta regolare per le proprietà di chiusura della classe dei linguaggi regolari.

Quesito: Si definisca una grammatica context free che generi il linguaggio $L = \{a^r b^s c^t a^m c^n | s = r + t; n \geq 2m; r, t, m, n \geq 0\}$. ***Soluzione***: *Una possibile soluzione è la grammatica*

$$
\begin{array}{lcl}
S & \rightarrow & ABCD \\
A & \rightarrow & aAb \mid \varepsilon \\
B & \rightarrow & bBc \mid \varepsilon \\
C & \rightarrow & aCcc \mid \varepsilon \\
D & \rightarrow & cD \mid \varepsilon
\end{array}
$$

Quesito: Si definisca una grammatica in Forma Normale di Greibach che generi il seguente linguaggio.

$$L = \{w \in \{a, b, c\}^+ || w| \text{ pari e } w \text{ inizia e termina con lo stesso carattere}\}$$

Soluzione: *Una possibile grammatica context free che genera L è:*

$$\begin{array}{rcl} S & \rightarrow & aAa|bAb|cAc \\ A & \rightarrow & aB|bB|cB|\varepsilon \\ B & \rightarrow & aA|bA|cA \end{array}$$

Per portarla in CNF, eliminano l'ε-produzione, ottendendo

$$\begin{array}{rcl} S & \rightarrow & aAa|bAb|cAc|aa|bb|cc \\ A & \rightarrow & aB|bB|cB \\ B & \rightarrow & aA|bA|cA|a|b|c \end{array}$$

La grammatica ottenuta non ha produzioni unitarie né simboli inutili, per cui è in forma ridotta. Può essere quindi derivata da essa la grammatica in CNF seguente:

$$\begin{array}{rcl} S & \rightarrow & XU|YV|ZW|XX|YY|ZZ \\ A & \rightarrow & XB|YB|ZB \\ B & \rightarrow & XA|YA|ZA|a|b|c \\ U & \rightarrow & AX \\ V & \rightarrow & AY \\ W & \rightarrow & AZ \\ X & \rightarrow & a \\ Y & \rightarrow & b \\ Z & \rightarrow & c \end{array}$$

E da essa, nel modo seguente, la grammatica in GNF:

$$\begin{array}{rcl} S & \rightarrow & aU|bV|cW|aX|bY|cZ \\ A & \rightarrow & aB|bB|cB \\ B & \rightarrow & aA|bA|cA|a|b|c \\ U & \rightarrow & aBX|bBX|cBX \\ V & \rightarrow & aBY|bBY|cBY \\ W & \rightarrow & aBZ|bBZ|cBZ \\ X & \rightarrow & a \\ Y & \rightarrow & b \\ Z & \rightarrow & c \end{array}$$