

# Esercizi di teoria dei linguaggi

## Indice

<b>1. Lezione 01 .....</b>	<b>2</b>
<b>2. Lezione 02 .....</b>	<b>3</b>
2.1. Esercizio 01 .....	3
2.2. Esercizio 02 .....	3
2.3. Esercizio 03 .....	4
2.4. Esercizio 04 .....	4
2.5. Esercizio 05 .....	5
2.6. Esercizio 06 .....	5
2.7. Esercizio 07 .....	6
2.8. Esercizio 08 .....	6
2.9. Esercizio 09 .....	6
<b>3. Lezione 03 .....</b>	<b>8</b>
3.1. Esercizio 01 .....	8
3.2. Esercizio 02 .....	8
3.3. Esercizio 03 .....	8
3.4. Esercizio 04 .....	9
<b>4. Lezione 04 .....</b>	<b>11</b>
4.1. Esercizio 01 .....	11
4.2. Esercizio 02 .....	12
4.3. Esercizio 03 .....	12

## **1. Lezione 01**

## 2. Lezione 02

### 2.1. Esercizio 01

Considerate l'alfabeto  $\Sigma = \{a, b\}$ .

- Fornite una grammatica context-free per il linguaggio delle stringhe palindrome di lunghezza pari su  $\Sigma$ , cioè per l'insieme  $\text{PAL}_{\text{pari}} = \{ww^R \mid w \in \Sigma^*\}$ .

Regole di produzione:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow aSa$ ;
- $S \rightarrow bSb$ .

- Modificate la grammatica precedente per generare l'insieme PAL di tutte le stringhe palindrome su  $\Sigma$ .

Regole di produzione:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow aSa$ ;
- $S \rightarrow bSb$ ;
- $S \rightarrow L$ ;
- $L \rightarrow a$ ;
- $L \rightarrow b$ .

- Per ogni  $k \in [0, 3]$  rispondete alla domanda “il linguaggio PAL é di tipo  $k$ ?” giustificando la risposta.

- Tipo 0: sì, ogni linguaggio é un linguaggio di tipo 0;
- Tipo 1: sì, per ogni regola di produzione  $\alpha \rightarrow \beta$  vale  $|\beta| \geq |\alpha|$ ;
- Tipo 2: sì, ogni regola di produzione  $\alpha \rightarrow \beta$  vede  $\alpha \in V$  e  $\beta \in (V \cup \Sigma^*)$ ;
- Tipo 3: no, la regola  $S \rightarrow aSa$  non é nella forma  $A \rightarrow aB$  oppure  $A \rightarrow a$ .

Se sostituiamo l'alfabeto con  $\Sigma = \{a, b, c\}$ , le risposte al punto precedente cambiano? E se lo sostituiamo con  $\Sigma = \{a\}$ ?

Se  $\Sigma = \{a, b, c\}$  le risposte non cambiano visto che vanno aggiunte le regole:

- $S \rightarrow cSc$ ;
- $L \rightarrow c$ .

Se  $\Sigma = \{a\}$  le regole di produzione diventano:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow a$ ;
- $S \rightarrow aSa$ ;

ma questo non fa cambiare le risposte.

### 2.2. Esercizio 02

Non ancora spiegato

### 2.3. Esercizio 03

Sia  $\Sigma = \{ (, ) \}$  un alfabeto i cui simboli sono la parentesi aperta e la parentesi chiusa.

Scrivete una grammatica context-free che generi il linguaggio formato da tutte le sequenze di parentesi correttamente bilanciate, come ad esempio  $((()())())$ .

Regole di produzione:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow (S)$ ;
- $S \rightarrow SS$ .

Risolvete il punto precedente per un alfabeto con due tipi di parentesi, come  $\Sigma = \{ (, ), [, ] \}$ , nel caso non vi siano vincoli tra i tipi di parentesi (le tonde possono essere contenute tra quadre e viceversa). Esempio  $[()([[]][[]])]$ , ma non  $[[][(())()]]$ .

Regole di produzione:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow (S)$ ;
- $S \rightarrow [S]$ ;
- $S \rightarrow SS$ .

Risolvete il punto precedente con  $\Sigma = \{ (, ), [, ] \}$ , con il vincolo che le parentesi quadre non possano mai apparire all'interno di parentesi tonde. Esempio  $[()([()])][[]]([()])$ , ma non  $[()([[]])][[]]$ .

Regole di produzione:

- $S \rightarrow \varepsilon$ ;
- $S \rightarrow [S]$ ;
- $S \rightarrow SS$ ;
- $S \rightarrow I$ ;
- $I \rightarrow \varepsilon$ ;
- $I \rightarrow (I)$ ;
- $I \rightarrow II$ .

### 2.4. Esercizio 04

Sia  $G = (V, \Sigma, P, S)$  la grammatica con  $V = \{S, B, C\}$ ,  $\Sigma = \{a, b, c\}$  e  $P$  contenente le seguenti produzioni:

- $S \rightarrow aSBC \mid aBC$ ;
- $CB \rightarrow BC$ ;
- $aB \rightarrow ab$ ;
- $bB \rightarrow bb$ ;
- $bC \rightarrow bc$ ;
- $cC \rightarrow cc$ .

Dopo avere stabilito di che tipo è  $G$ , provate a derivare alcune stringhe. Riuscite a dire da quali stringhe è formato il linguaggio generato da  $G$ ?

La grammatica  $G$  é di tipo 1.

Deriviamo qualche stringa:

- $S \rightarrow aBC \rightarrow abC \rightarrow abc$ ;
- $S \rightarrow aSBC \rightarrow aaBCBC \rightarrow aabCBC \rightarrow aabBCC \rightarrow aabbCC \rightarrow aabbC \rightarrow aabbcc$ .

Il linguaggio  $L(G)$  è l'insieme  $\{a^n b^n c^n \mid n \geq 1\}$ .

## 2.5. Esercizio 05

Sia  $G = (V, \Sigma, P, S)$  la grammatica con  $V = \{S, B, C\}$ ,  $\Sigma = \{a, b, c\}$  e  $P$  contenente le seguenti produzioni:

- $S \rightarrow aBSc \mid abc$ ;
- $Ba \rightarrow aB$ ;
- $Bb \rightarrow bb$ .

Dopo avere stabilito di che tipo é  $G$ , provate a derivare alcune stringhe. Riuscite a dire da quali stringhe é formato il linguaggio generato da  $G$ ?

La grammatica  $G$  é di tipo 1.

Deriviamo qualche stringa:

- $S \rightarrow abc$ ;
- $S \rightarrow aBSc \rightarrow aBabcc \rightarrow aaBbcc \rightarrow aabbcc$ .

Il linguaggio  $L(G)$  è l'insieme  $\{a^n b^n c^n \mid n \geq 1\}$ .

## 2.6. Esercizio 06

Sia  $G = (V, \Sigma, P, S)$  la grammatica con  $V = \{S, A, B, C, D, E\}$ ,  $\Sigma = \{a, b\}$  e  $P$  contenente le seguenti produzioni:

- $S \rightarrow ABC$ ;
- $AB \rightarrow aAD \mid bAE \mid \varepsilon$ ;
- $DC \rightarrow BaC$ ;
- $EC \rightarrow BbC$ ;
- $Da \rightarrow aD$ ;
- $Db \rightarrow bD$ ;
- $Ea \rightarrow aE$ ;
- $Eb \rightarrow bE$ ;
- $C \rightarrow \varepsilon$ ;
- $aB \rightarrow Ba$ ;
- $bB \rightarrow bB$ .

Dopo avere stabilito di che tipo é  $G$ , provate a derivare alcune stringhe. Riuscite a dire da quali stringhe é formato il linguaggio generato da  $G$ ?

La grammatica  $G$  é di tipo 1.

Deriviamo qualche stringa:

- $S \rightarrow ABC \xrightarrow{*} \varepsilon$ ;
- $S \rightarrow ABC \rightarrow aADC \rightarrow aABaC \xrightarrow{*} aa$ ;

- $S \xrightarrow{*} aABaC \rightarrow aaADaC \rightarrow aaAaDC \rightarrow aaAaBaC \rightarrow aaABaaC \xrightarrow{*} aaaa;$
- $S \xrightarrow{*} aABaC \rightarrow abAEaC \rightarrow abAaEC \rightarrow abAaBbC \rightarrow abABabC \xrightarrow{*} abab;$
- $S \rightarrow ABC \rightarrow bAEC \rightarrow bABbC \xrightarrow{*} bb;$
- $S \xrightarrow{*} bABbC \rightarrow bbAEbC \rightarrow bbAbEC \rightarrow bbAbBbC \rightarrow bbABbbC \xrightarrow{*} bbbb;$
- $S \xrightarrow{*} bABbC \rightarrow baADbC \rightarrow baAbDC \rightarrow baAbBaC \rightarrow baABbaC \xrightarrow{*} baba.$

Il linguaggio  $L(G)$  è l'insieme  $\{a^{2n} \cup b^{2n} \cup (ab)^{2n} \cup (ba)^{2n} \mid n \geq 0\}$ .

## 2.7. Esercizio 07

Sia  $G = (V, \Sigma, P, S)$  la grammatica con  $V = \{S, A, B, C, X, Y, L, R\}$ ,  $\Sigma = \{a\}$  e  $P$  contenente le seguenti produzioni:

- $S \rightarrow LXR;$
- $LX \rightarrow LYYA \mid aC;$
- $AX \rightarrow YYA;$
- $AR \rightarrow BR;$
- $YB \rightarrow BX;$
- $LB \rightarrow L;$
- $CX \rightarrow aC;$
- $CR \rightarrow \varepsilon.$

Riuscite a stabilire da quali stringhe é formato il linguaggio generato da  $G$ ?

Deriviamo qualche stringa:

- $S \rightarrow LXR \rightarrow aCR \rightarrow a;$
- $S \rightarrow LXR \rightarrow LYYAR \xrightarrow{*} LXXR \rightarrow aCXR \rightarrow aaCR \rightarrow aa;$
- $S \rightarrow LXR \xrightarrow{*} LXXR \rightarrow LYYAXR \rightarrow LYYYYYAR \xrightarrow{*} LXXXXXR \xrightarrow{*} aaaa.$
- $S \rightarrow LXR \xrightarrow{*} LXXXXXR \xrightarrow{*} LYYYYYYYYYAR \xrightarrow{*} LXXXXXXXXXR \xrightarrow{*} aaaaaaaaaa.$

Il linguaggio  $L(G)$  è l'insieme  $\{a^{2^n} \mid n \geq 0\}$ .

## 2.8. Esercizio 08

Modificate la grammatica dell'esercizio 07 in modo da ottenere una grammatica di tipo 1 che generi lo stesso linguaggio.

Modificando la regola  $LB \rightarrow L$  in  $LB \rightarrow CRL$  la grammatica diventa di tipo 1.

## 2.9. Esercizio 09

Dimostrate che la grammatica  $G = (\{A, B, S\}, \{a, b\}, P, S)$ , con l'insieme delle produzioni  $P$  elencate sotto, genera il linguaggio  $\{w \in \{a, b\}^* \mid \forall x \in \{a, b\}^* w \neq xx\}$ :

- $S \rightarrow AB \mid BA \mid A \mid B$
- $A \rightarrow aAa \mid aAb \mid bAa \mid bAb \mid a$
- $B \rightarrow aBa \mid aBb \mid bBa \mid bBb \mid b$

Consideriamo in primo luogo i “casi base”:

- $S \rightarrow A \rightarrow a$  va bene perché di lunghezza dispari;
- $S \rightarrow B \rightarrow b$  va bene perché di lunghezza dispari;
- $S \rightarrow AB \xrightarrow{*} ab$  va bene perché  $a \neq b$ ;
- $S \rightarrow BA \xrightarrow{*} ba$  va bene perché  $b \neq a$ .

Consideriamo poi  $S \rightarrow A \mid B$ :

$$\begin{aligned}
S \rightarrow A &\rightarrow aAa \xrightarrow{*} a^n Aa^n \rightarrow a^n aa^n; \\
&aAa \xrightarrow{*} ab^n Ab^n a \rightarrow ab^n ab^n a; \\
&aAa \xrightarrow{*} a\{a, b\}^n A\{a, b\}^n a \rightarrow a\{a, b\}^n a\{a, b\}^n a; \\
&aAb \xrightarrow{*} \dots . \\
S \rightarrow B &\rightarrow aBa \xrightarrow{*} a^n Ba^n \rightarrow a^n ba^n; \\
&aBa \xrightarrow{*} ab^n Bb^n a \rightarrow ab^n bb^n a; \\
&aBa \xrightarrow{*} a\{a, b\}^n B\{a, b\}^n a \rightarrow a\{a, b\}^n b\{a, b\}^n a; \\
&aBb \xrightarrow{*} \dots .
\end{aligned}$$

Tutte le stringhe che vengono generate vanno bene perché sono di lunghezza dispari.

Consideriamo infine  $S \rightarrow AB \mid BA$  in due casi:

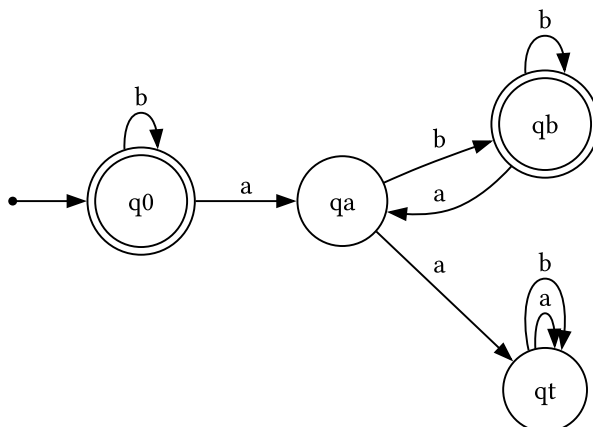
- se eseguiamo su  $A$  e  $B$  lo stesso numero di passi di derivazione abbiamo altri due casi:
  - usiamo regole con lo “stesso contesto”, ma alla fine avremo un carattere diverso nella posizione dove sono presenti  $A$  e  $B$ ;
  - usiamo regole con “diverso contesto”, ma la prima regola che rispecchia questa casistica ha almeno un carattere diverso (oltre ad avere il carattere in  $A$  e  $B$  diverso alla fine della derivazione);
- se eseguiamo su  $A$  e  $B$  un numero diverso di passi di derivazione, abbiamo due punti di partenza:
  - partiamo da  $AB$  e indichiamo con  $n$  la lunghezza della stringa derivata da  $A$  e con  $k$  la lunghezza della stringa derivata da  $B$ , con  $k > n$ . Per ottenere due stringhe della stessa lunghezza devo rimuovere da  $k$  un numero  $\frac{n-k}{2}$  di caratteri e appenderli a  $n$ , ottenendo due stringhe di lunghezza  $t$ . Prima dell'ultimo passo di derivazione di  $A$  la variabile  $A$  era in posizione  $\frac{n-1}{2}$ , mentre ora si trova in posizione  $\frac{t-1}{2} - \frac{n-k}{4}$  perché prima mi devo prima posizionare nel “nuovo centro” e poi mi devo spostare di una posizione indietro ogni due caratteri che avevo aggiunto. Facciamo lo stesso ragionamento per trovare l'indice dell'ultima  $B$  di  $B$ . Le due posizioni trovate sono le stesse, ma prima dell'ultima derivazione in  $A$  si aveva una  $A$  e in  $B$  si aveva una  $B$ , che però generano rispettivamente  $a$  e  $b$ , quindi otteniamo due stringhe che sono sempre diverse.
  - partiamo da  $BA$  e facciamo lo stesso discorso, basta invertire l'ordine delle stringhe.

Abbiamo quindi dimostrato che  $L(G) = \{w \in \{a, b\}^* \mid \forall x \in \{a, b\}^* w \neq xx\}$ .

### 3. Lezione 03

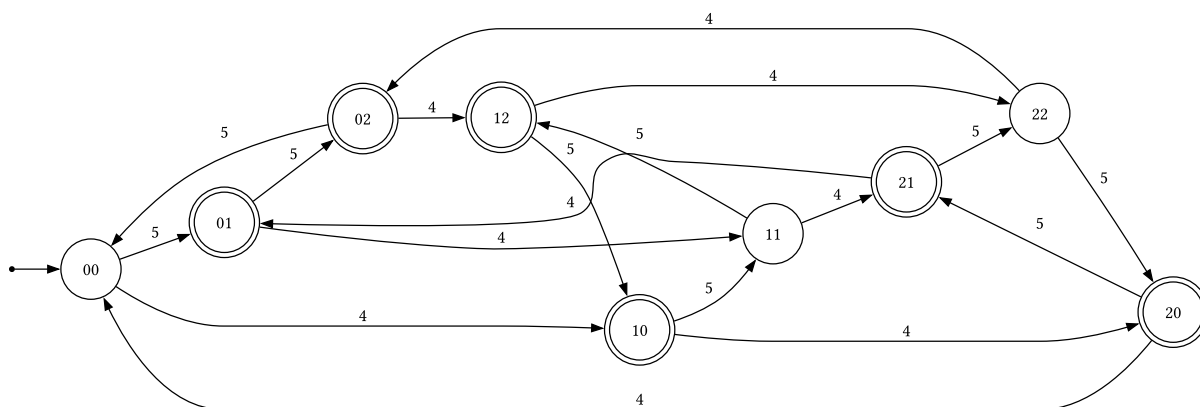
#### 3.1. Esercizio 01

Costruite un automa a stati finiti che riconosca il linguaggio formato da tutte le stringhe sull'alfabeto  $\{a, b\}$  nelle quali ogni  $a$  é seguita immediatamente da una  $b$ .



#### 3.2. Esercizio 02

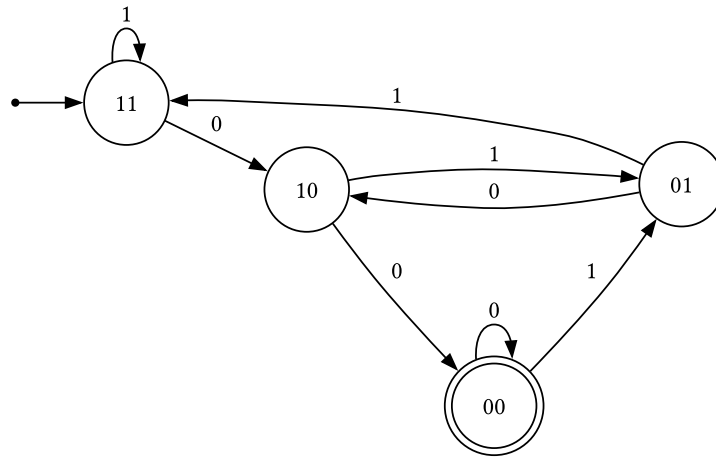
Costruite un automa a stati finiti che riconosca il linguaggio formato da tutte le stringhe sull'alfabeto  $\{4, 5\}$  che, interpretate come numeri in base 10, rappresentano numeri interi che *non* sono divisibili per 3.



#### 3.3. Esercizio 03

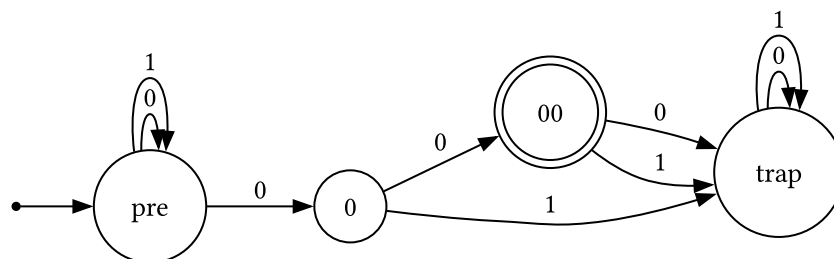


Costruite un automa a stati finiti deterministico che riconosca il linguaggio formato da tutte le stringhe sull'alfabeto  $\{0, 1\}$  che, interpretate come numeri in notazione binaria, denotano multipli di 4.



Utilizzando il non determinismo si riesce a costruire un automa con meno stati? Generalizzate l'esercizio a multipli di  $2k$ , dove  $k > 0$  è un intero fissato.

Utilizzando il non determinismo utilizziamo ancora 4 stati.

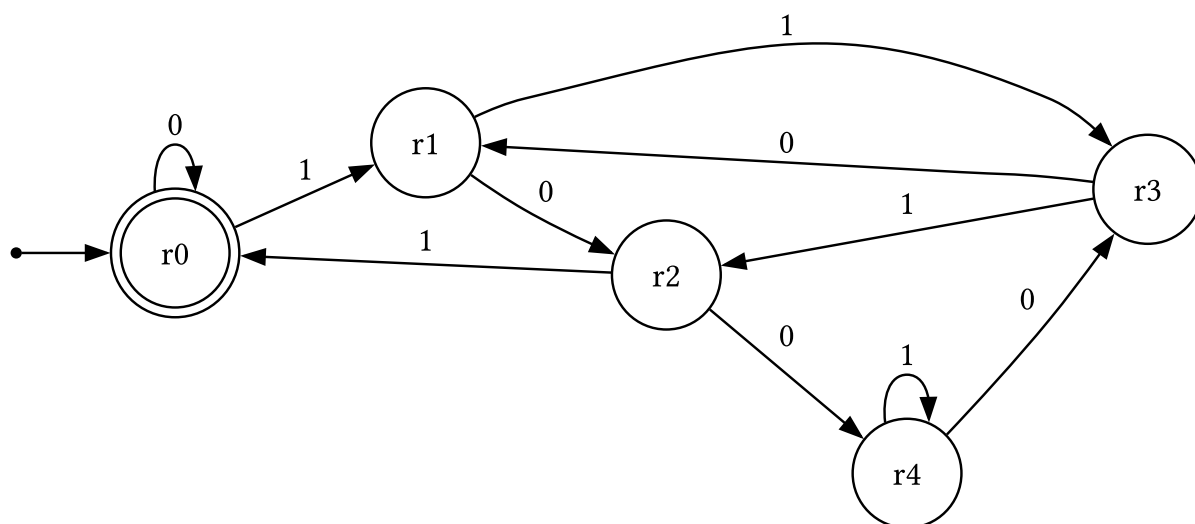


Generalizzando a multipli di  $2k$ , con  $k > 0$ , abbiamo:

- per il DFA  $2^k$  stati;
- per il NFA  $k + 2$  stati.

### 3.4. Esercizio 04

Costruite un automa a stati finiti che riconosca il linguaggio formato da tutte le stringhe sull'alfabeto  $\{0, 1\}$  che, interpretate come numeri in notazione binaria, rappresentano multipli di 5.



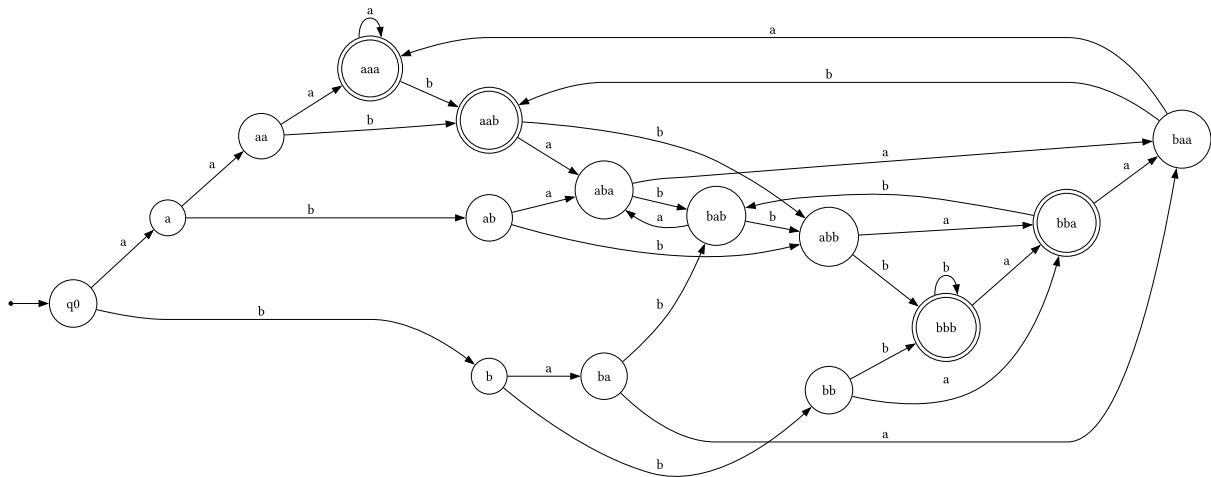
## 4. Lezione 04

### 4.1. Esercizio 01

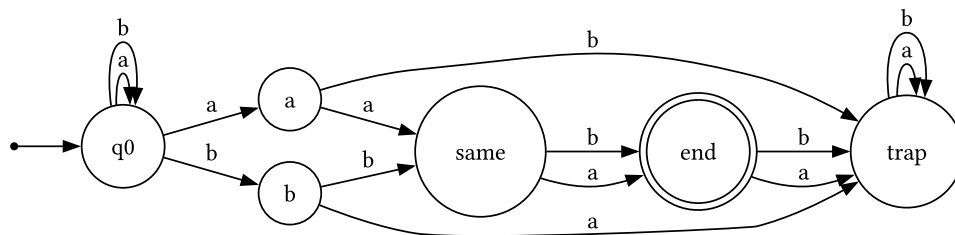
Considerate il linguaggio

$$L = \{w \in \{a, b\}^* \mid \text{il penultimo e il terzultimo simbolo di } w \text{ sono uguali}\}.$$

Costruite un automa a stati finiti deterministico che accetta  $L$ .



Costruite un automa a stati finiti non deterministico che accetta  $L$ .



Dimostrate che per il linguaggio  $L$  tutte le stringhe di lunghezza 3 sono distinguibili tra loro.

	<i>aaa</i>	<i>aab</i>	<i>aba</i>	<i>abb</i>	<i>baa</i>	<i>bab</i>	<i>bba</i>	<i>bbb</i>
<i>aaa</i>	-	<i>a</i>	$\varepsilon$	$\varepsilon$	$\varepsilon$	$\varepsilon$	<i>a</i>	<i>aa</i>
<i>aab</i>	-	-	$\varepsilon$	$\varepsilon$	$\varepsilon$	$\varepsilon$	<i>bb</i>	<i>b</i>
<i>aba</i>	-	-	-	<i>b</i>	<i>a</i>	<i>aa</i>	$\varepsilon$	$\varepsilon$

<i>abb</i>	-	-	-	-	<i>aa</i>	<i>b</i>	$\varepsilon$	$\varepsilon$
<i>baa</i>	-	-	-	-	-	<i>a</i>	$\varepsilon$	$\varepsilon$
<i>bab</i>	-	-	-	-	-	-	$\varepsilon$	$\varepsilon$
<i>bba</i>	-	-	-	-	-	-	-	<i>a</i>
<i>bbb</i>	-	-	-	-	-	-	-	-

Dimostrate che per il linguaggio  $L$  la parola vuota é distinguibile da tutte le stringhe di lunghezza 3.

	<i>aaa</i>	<i>aab</i>	<i>aba</i>	<i>abb</i>	<i>baa</i>	<i>bab</i>	<i>bba</i>	<i>bbb</i>
$\varepsilon$	$\varepsilon$	$\varepsilon$	<i>ab</i>	<i>a</i>	<i>a</i>	<i>ba</i>	$\varepsilon$	$\varepsilon$

Utilizzando i risultati precedenti, ricavate un limite inferiore per il numero di stati di ogni automa deterministico che accetta  $L$ .

L'insieme  $X = \{w \in \{a, b\}^+ \mid |w| = 3\}$  é un insieme di parole tutte distinguibili tra loro rispetto al linguaggio  $L$ , come dimostrato nei punti precedenti, quindi ogni DFA per  $L$  deve avere almeno  $|X|$  stati, ovvero almeno 8 stati.

## 4.2. Esercizio 02

Costruite un insieme di stringhe distinguibili tra loro per ognuno dei seguenti linguaggi:

- $L_1 = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ,
- $L_2 = \{a^n b^n \mid n \geq 0\}$ ,
- $L_3 = \{ww^R \mid w \in \{a, b\}^*\}$  dove, per ogni stringa  $w$ ,  $w^R$  indica la stringa  $w$  scritta al contrario.

$X_1 = \{\varepsilon, a, b, ab\}$ .

$X_2$  ha cardinalità infinita.

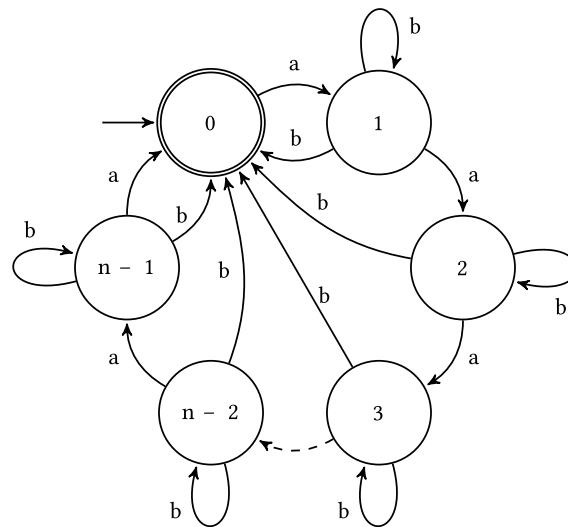
$X_3$  ha cardinalità infinita.

Per alcuni di questi linguaggi riuscite ad ottenere insiemi di stringhe distinguibili di cardinalità infinita? Cosa significa ciò?

I linguaggi che hanno insiemi di stringhe distinguibili di cardinalità infinita sono linguaggi non di tipo 3.

## 4.3. Esercizio 03

Considerate l'automa di Meyer e Fischer  $M_n$  presentato nella Lezione 4 (caso peggiore della costruzione per sottoinsiemi) e mostrato nella seguente figura:



Descrivete a parole la proprietà che deve soddisfare una stringa per essere accettata da  $M_n$ . Riuscite a costruire un automa non deterministico, diverso da  $M_n$ , per lo stesso linguaggio, basandovi su tale proprietà?

Non lo so fare.