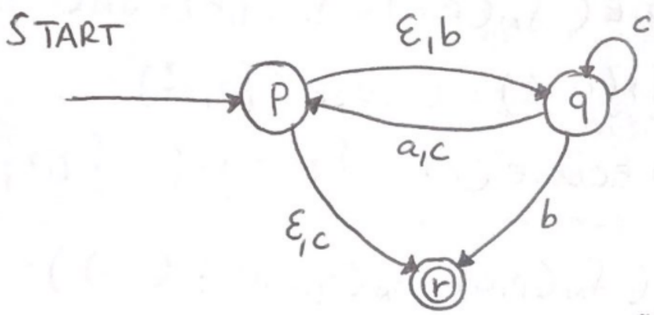
LINGUAGGI DI COMPUTABILITÀ

costruire il DFA equivalente al seguente ε-NFA, mostrando tutti i passaggi.



tutti + gli stati raggiungibili con ε + lo stato stesso.

1 Scrivo gli ECLOSE di tutti gli stati:

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

$$ECLOSE(q) = \{q\}$$

$$ECLOSE(r) = \{r\}$$

2

contiene stato finale — stato iniziale

			a	b	c
* →	$\{p, q, r\}$	A	$\{p, q, r\}$ A	$\{q, r\}$ B	$\{p, q, r\}$ A
*	$\{q, r\}$	B	$\{p, q, r\}$ A	$\{r\}$ C	$\{p, q, r\}$ A
*	$\{r\}$	C	$\emptyset$ D	$\emptyset$ D	$\emptyset$ D
	$\emptyset$	D	$\emptyset$ D	$\emptyset$ D	$\emptyset$ D

$$\delta_D(\{p,q,r\}, a) = \text{ECLOSE}(\delta_N(p,a) \cup \delta_N(q,a) \cup \delta_N(r,a))$$
$$= \text{ECLOSE}(\emptyset \cup \{p\} \cup \emptyset) = \text{ECLOSE}(\{p\}) =$$
$$= \text{ECLOSE}(p) = \{p,q,r\}$$

$$\delta_D(\{p,q,r\}, b) = \text{ECLOSE}(\delta_N(p,b) \cup \delta_N(q,b) \cup \delta_N(r,b)) =$$
$$= \text{ECLOSE}(\{q\} \cup \{r\} \cup \emptyset) = \text{ECLOSE}(\{q,r\}) =$$
$$\text{ECLOSE}(q) \cup \text{ECLOSE}(r) = \{q\} \cup \{r\} = \{q,r\}$$

$$\delta_D(\{p,q,r\}, c) = \text{ECLOSE}(\delta_N(p,c) \cup \delta_N(q,c) \cup \delta_N(r,c)) =$$
$$\text{ECLOSE}(\{r\} \cup \{q,p\} \cup \emptyset) = \text{ECLOSE}(\{r,q,p\}) =$$
$$\text{ECLOSE}(r) \cup \text{ECLOSE}(q) \cup \text{ECLOSE}(p) = \{p,q,r\}$$

$$\delta_D(\{q,r\}, a) = \text{ECLOSE}(\delta_N(q,a) \cup \delta_N(r,a)) = \text{ECLOSE}(\{p\} \cup \emptyset) =$$
$$\text{ECLOSE}(\{p\}) = \text{ECLOSE}(p) = \{p,q,r\}$$

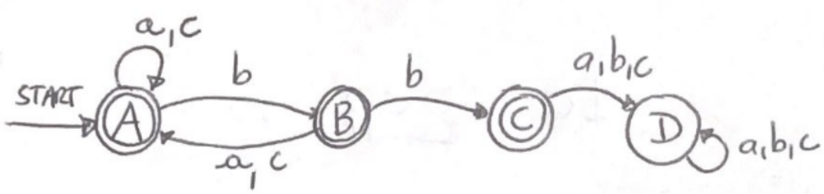
$$\delta_D(\{q,r\}, b) = \text{ECLOSE}(\delta_N(q,b) \cup \delta_N(r,b)) = \text{ECLOSE}(\{r\} \cup \emptyset) =$$
$$\text{ECLOSE}(\{r\}) = \text{ECLOSE}(r) = \{r\}$$

$$\delta_D(\{q,r\}, c) = \text{ECLOSE}(\delta_N(q,c) \cup \delta_N(r,c)) = \text{ECLOSE}(\{q,p\} \cup \emptyset) =$$
$$\text{ECLOSE}(q) \cup \text{ECLOSE}(p) = \{p,q,r\}$$

$$\delta_D(\{r\}, a) = \text{ECLOSE}(\delta_N(r,a)) = \text{ECLOSE}(\emptyset) = \emptyset$$

$$\delta_D(\{r\}, b) = \text{ECLOSE}(\delta_N(r,b)) = \text{ECLOSE}(\emptyset) = \emptyset$$

$$\delta_D(\{r\}, c) = \text{ECLOSE}(\delta_N(r,c)) = \text{ECLOSE}(\emptyset) = \emptyset$$

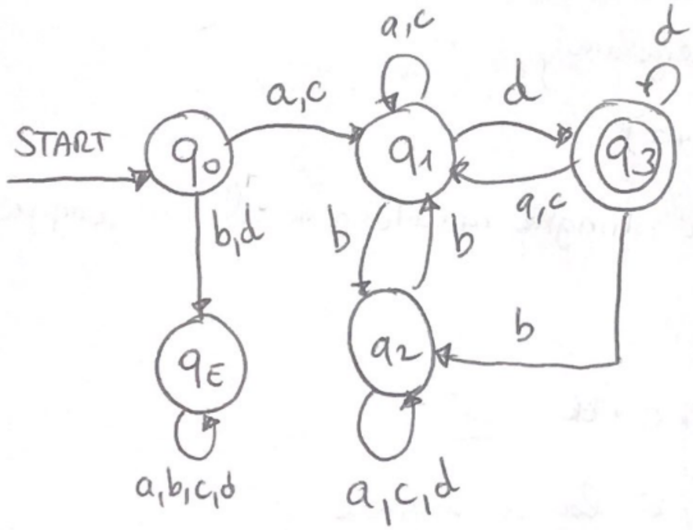


Costruire il DFA che riconosce il linguaggio

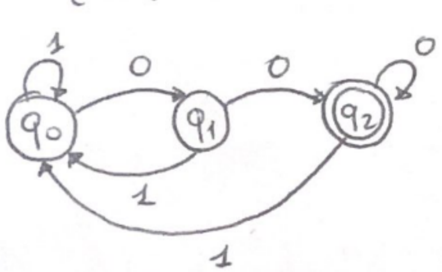
$L = \{ w \in \{a,b,c,d\}^* \mid w$ inizia con a o con c, termina con d e contiene un numero pari di b$\}$.

abcbd
cd
cbbdbabad
$\Big\} \in L$

bbd
cbdbbd
$\Big\} \notin L$



$L = \{ w \in \{0,1\}^* \mid w$ è una stringa che finisce con 00$\}$.



insieme di stringhe che contengono tre zeri consecutivi

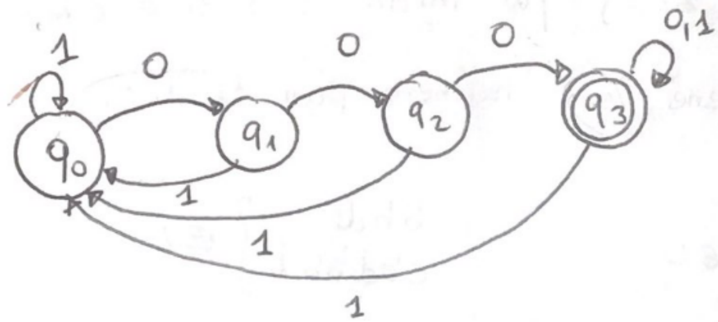
$\underbrace{000}$

$1\underbrace{000}1$ $\qquad$ $1001\underbrace{000}$

$11\underbrace{000}1$

$\underbrace{000}110$

$0000 1$



---

data l'espressione regolare

$$a a^* (bb+c)^* (b+a)$$

stabilire se le seguenti stringhe appartengono al linguaggio motivando la risposta

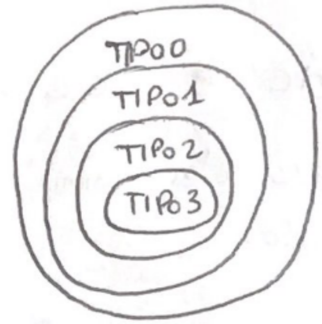
abbca = $a \cdot \varepsilon \cdot bb \cdot c \cdot a$ <u>SÌ</u>

bbcb = <u>NO</u> non c'è la a iniziale

aab = $a \cdot a \cdot b$ <u>SÌ</u>

abbba = $a \cdot \varepsilon \cdot bb \cdot b$ MANCA LA A FINALE <u>NO</u>

GERARCHIA DI CHOMSKY



- TIPO 0: nessuna restrizione
- TIPO 1: $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$
- TIPO 2: Context-free $A \to XY$
- TIPO 3: regolari

$$\underbrace{\begin{array}{l} A \to aB \\ A \to a \end{array}}_{\text{lineare a dx}} \quad \Bigg| \quad \underbrace{\begin{array}{l} A \to Ba \\ A \to a \end{array}}_{\text{lineare a SX}}$$

Data la grammatica, dove $\Sigma = \{a, b, c, d\}$

$S \to Sd \mid Ac$

$B \to a$

$A \to Bb \mid Cc$

$C \to Cb \mid a$

① Dire di che tipo è la grammatica: tipo 3, lineare a sinistra.

② Mostrare una derivazione per la stringa e una derivazione per la stringa abcdd e abcc:

abcdd

$\underline{S} \Rightarrow \underline{S}d \Rightarrow \underline{S}dd \Rightarrow \underline{A}cdd \Rightarrow \underline{B}bcdd \Rightarrow abcdd$

doppie frecce!!!

abcc

$S \Rightarrow \underline{A}c \Rightarrow \underline{C}cc \Rightarrow \underline{C}bcc \Rightarrow abcc$

3 Definire l'espressione regolare che denota il linguaggio generato.

$$(ab + ab^*c)\ c\ d^*$$

$$\underbrace{a\ (b+ab^*c)\,cd^*}$$

CFG = Context free grammar

Si consideri il linguaggio $L = \{ w \in \{a,b,c\}^* \mid w = a^m c b^n a c^{m+n} \}$ con $m \geq 0$, $\underline{n \geq 0}$

$a^m c b^n a c^{m+n}$

$\underbrace{a^m c \underbrace{b^n a c^n}_{A} c^m}_{S}$

es:

acbaccccc

$S \rightarrow aSc \mid cA$

$A \rightarrow bac \mid bAc$

mostra una derivazione:

acbacc

$S \Rightarrow a\underline{S}c \Rightarrow ac\underline{A}c \Rightarrow acbac$

albero di derivazione: