[Mart. 23 Genn. - Varricchio - DI. - LEZ. 6] - In 1 ora 3 quesiti -

ES. 1. Siano dati i linguaggi
$L_1 = \{bac, bb, aba\}$ e $L_2 = \{bea, ba, aee\}$.
Descrivere, elencandone i suoi elementi, il linguaggio $L = L_1 \cdot L_2$.

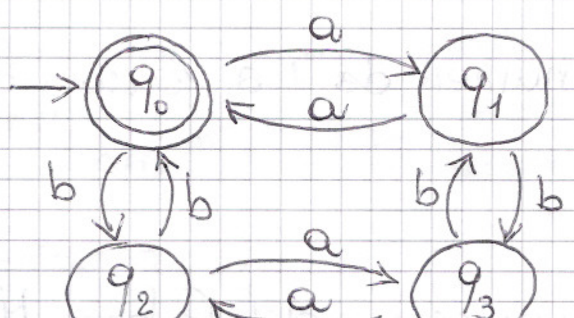
SOL: $L = L_1 \cdot L_2 = \{bacbea, bacba, bacaee, bbbea, bbba, bbaee, ababea, ababa, abaaee\}$

Se ci sono ripetizioni, bisogna levarle.

ES. 2. Dato l'automa deterministico $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ dove $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$ e la funzione di transizione δ è definita come segue: $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_2$, $\delta(q_1, a) = q_0$, $\delta(q_1, b) = q_3$, $\delta(q_2, a) = q_3$, $\delta(q_2, b) = q_0$, $\delta(q_3, a) = q_2$, $\delta(q_3, b) = q_1$.
Disegnare il grafo dell'automa. Fornire esempi di stringhe accettate. Eventualmente fornire una descrizione del linguaggio accettato (questo per la lode).

SOL: Se non è specificato, q_0 è lo stato iniziale.

	δ	a	b
$\rightarrow *$	q_0	q_1	q_2
	q_1	q_0	q_3
	q_2	q_3	q_0
	q_3	q_2	q_1



Per quanto riguarda gli ESEMPI: basta fare tentativi e cercare dei percorsi che, dallo stato iniziale, ritornano nello stato iniziale.

abba

aabb

La descrizione del LINGUAGGIO ACCETTATO è:
$\mathcal{L} = \{ w \mid w$ hanno un numero pari di a ed un numero pari di b $\}$

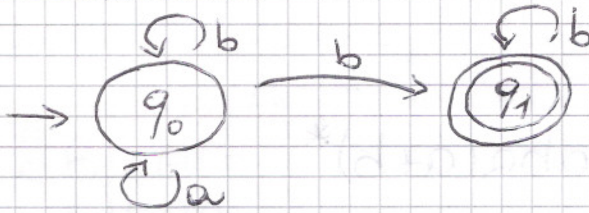
ES. 3. Dato l'automa non-deterministico $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ dove $Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $F = \{q_1\}$ e la funzione di transizione δ è definita come segue: $\delta(q_0, a) = \{q_0\}$, $\delta(q_0, b) =$ [illegible]
$\delta(q_1, a) = \emptyset$, $\delta(q_1, b) = \{q_1\}$. Disegnare il grafo dell'automa.

Costruire l'automa deterministico equivalente e rappresentarne il grafo. Descrivere il linguaggio accettato.

SOL.:

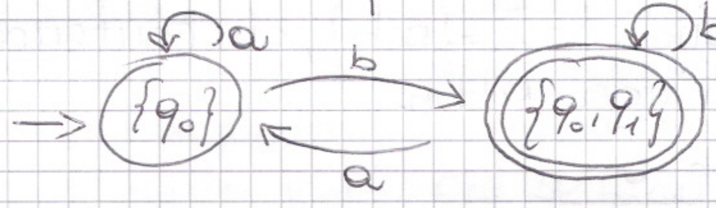
δ	a	b
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_1\}$

Grafo:



Non sono presenti archi etichettati con a

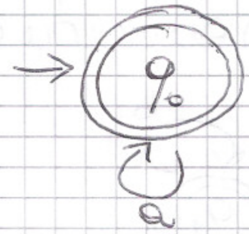
Grafo deterministico equivalente:



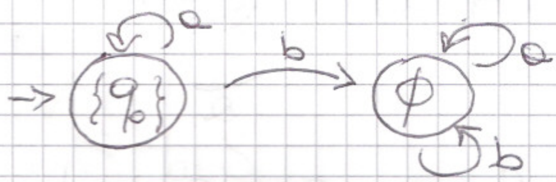
si fa prendendo i sottoinsiemi che sono 4; ma non è detto che vengano usati tutti e 4

$\{q_0\} \cup \{\phi\} = \{q_0\}$

ES:



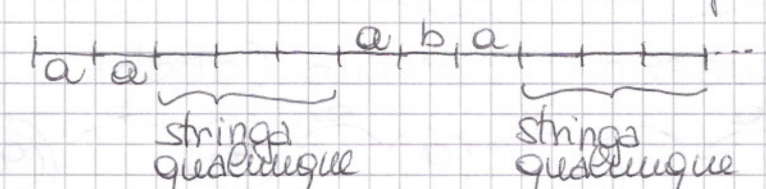
in questo automa, in cui $\Sigma = \{a, b\}$, non sono presenti archi etichettati con b. È un automa che accetta tutte le potenze di a a^*. La sua subset construction sarà:



L'insieme degli stati accettanti sarà $\{q_0, q_1\}$ perché dentro c'è q_1. Se vi era anche lo stato $\{q_1\}$, anche lui era accettante.
Il linguaggio accettato è $\mathcal{L} = \{w \mid w$ finiscono con una $b\}$

ES. 5. Sia L un linguaggio sull'alfabeto $\Sigma = \{a, b\}$ delle parole che iniziano per aa e contengono aba. Definire un'espressione regolare per L.

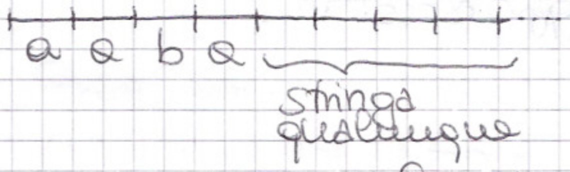
SOL. bisogna tener conto del caso in cui i due pezzi possono sovrapporsi.



l'espressione regolare sarà:

$$aa(a+b)^*\,aba\,(a+b)^*$$

Si possono sovrapporre in un solo modo:



la cui espressione regolare sarà:

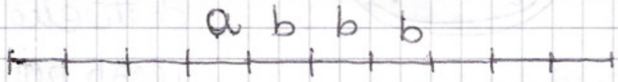
$$aaba(a+b)^*$$

le unisco e ottengo:

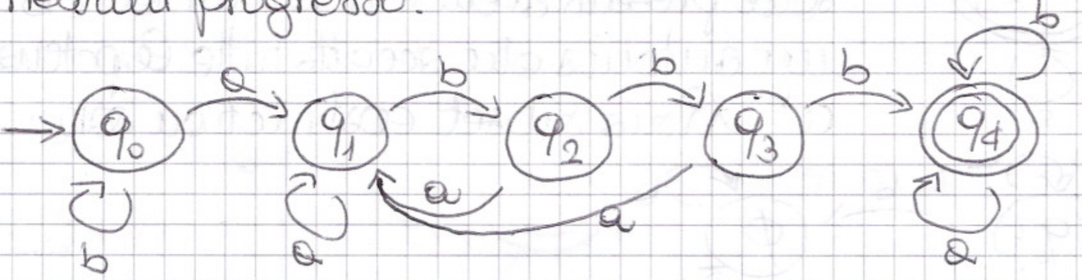
$$aa(a+b)^*aba(a+b)^* + aaba(a+b)^*$$

ES. 6. Costruire un automa deterministico che accetti tutte le stringhe, sull'alfabeto $\Sigma=\{a,b\}$, che contengono al proprio interno la stringa abbb.

SOL:

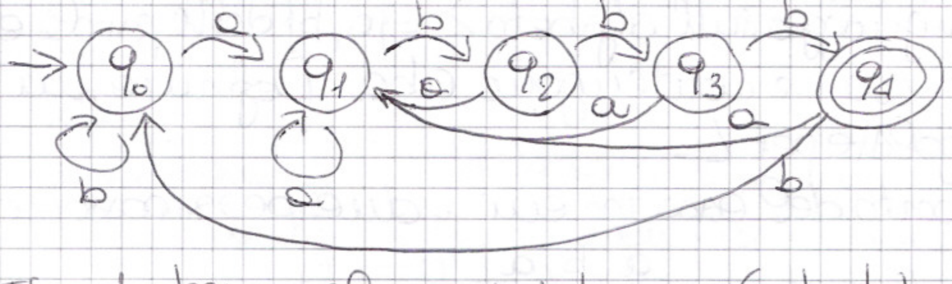


L'idea è: devo riconoscere la presenza di abbb, si parte dallo stato iniziale q_0 che rappresenta lo stato in cui non c'è nessun progresso.

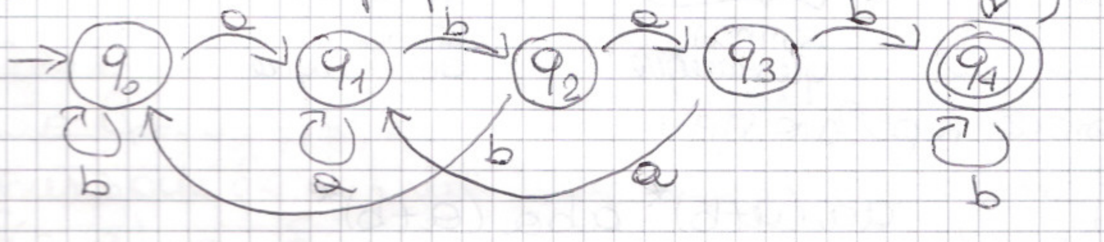


Una volta trovato (abbb), qualunque cosa ci sia dopo, accetto la stringa.

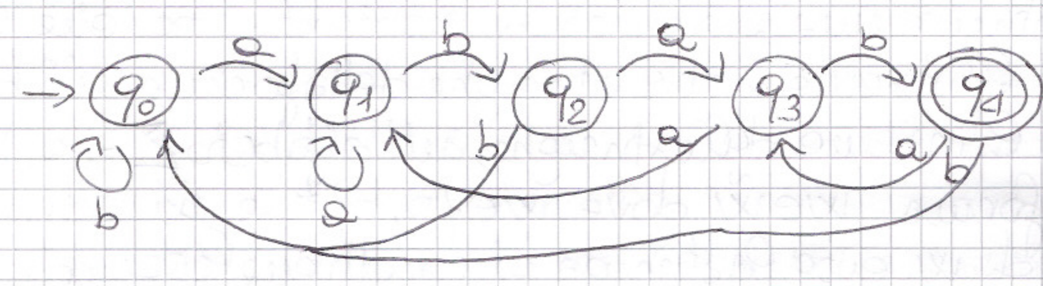
ES: trovare un automa deterministico che accetti tutte le stringhe che terminano con (abbb).



ES: che hanno al proprio interno (abab)



ES: che terminano con (abab)



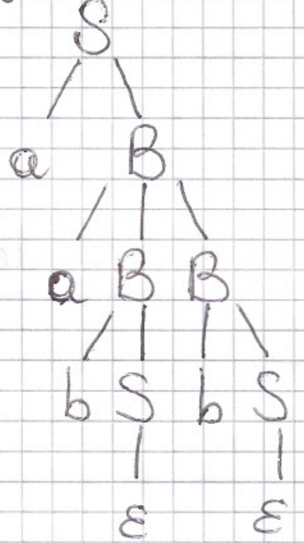
ES. 7. Data la grammatica $G = \langle \{S, A, B\}, \{a, b\}, P, S \rangle$ definita dalle seguenti regole:

$S \to aB \mid bA \mid \varepsilon$
$A \to aS \mid bAA$
$B \to bS \mid aBB$

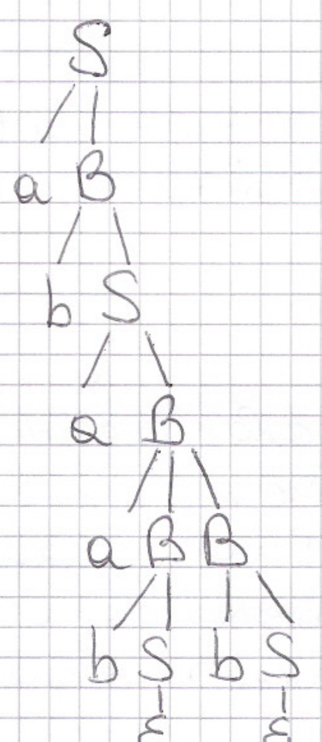
a) Fornire una derivazione ed un albero di derivazione per le parole (abaabb) e (aabb)

b) Eventualmente descrivere il linguaggio generato dalla grammatica G.

SOL:



La stringa inizia per a, quindi applico $S \to aB$
B deve generare, derivare abb, quindi applico $B \to aBB$.
applico $B \to bS$ 2 volte
e poi applico $S \to \varepsilon$



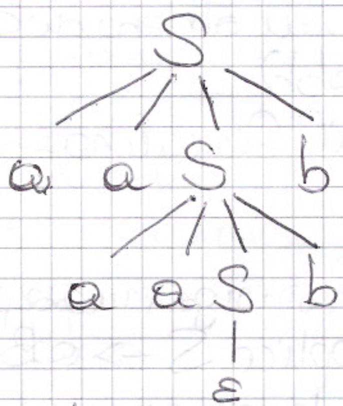
inizia per a. (una b in più)
inizia per b. (una con stesse a e b)

ES. 8. Per ciascuno dei seguenti linguaggi sull'alfabeto $\Sigma=\{a,b\}$ costruire una grammatica context-free che lo generi:
(a) l'insieme delle stringhe della forma $a^{2n}b^n$, con $n\geq 0$;
(b) l'insieme delle stringhe, sull'alfabeto $\Sigma=\{a,b,e\}$ della forma $we\tilde{w}$ dove $w\in\{a,b\}^*$ e $\tilde{w}$ indica l'inflesso di w, cioè la stringa che si ottiene leggendo w da destra verso sinistra.

SOL: $\mathcal{L}=\{a^{2n}b^n \mid n\geq 0\}$ $\quad\Sigma=\{a,b\}$

$S\rightarrow aaSb$
$S\rightarrow \varepsilon$

es (aaaabb), l'albero di derivazione è:



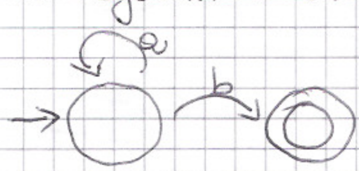
$\mathcal{L}=\{we\tilde{w} \mid w\in\{a,b\}^*\}$

hanno un blocco w di stringhe su $\{a,b\}$ e, poi, c'è una e e, poi, la stessa w, ma inflessa.

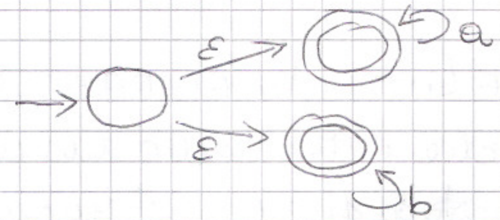
es: (abeba)
(abbebba)

$S\rightarrow aSa$
$S\rightarrow bSb$
$S\rightarrow e$

La famiglia dei linguaggi accettati dagli automi a stati finiti è la stessa delle espressioni regolari. C'è un automa, la cui espressione regolare si vede subito:



Il linguaggio è a^*b (espressione regolare), un certo numero di a e, poi, la b. Passare dall'automa all'espressione regolare è facile, si vede ad occhio. Se ci sono un certo numero di stati, la situazione è più complicata (non vi capiterà). Il passaggio dall'espressione regolare all'automa si può fare in modo sistematico. Posso costruire questo automa; archi che hanno ε come etichetta. Su questo automa, le stringhe accettate (il modello è non deterministico) sono le etichette dei cammini accettanti però ε non lo devo contare. aa è accettata, il cammino è εaa ma l'etichetta è aa. Nei cammini accettanti, la ε non si conta. È come se fosse una stringa vuota.



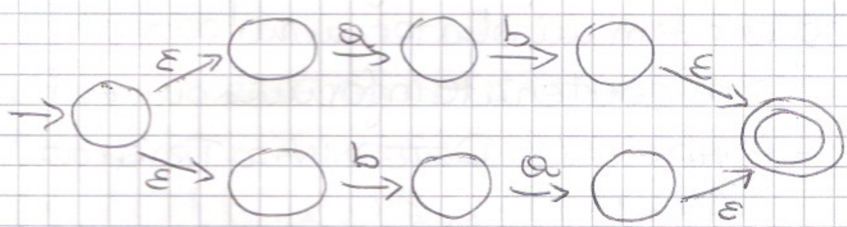
Es. 4. Costruire un'automa con ε-transizioni che accetti il linguaggio definito dalla seguente espressione regolare:

$$E=(ab+ba)^* bb^*$$

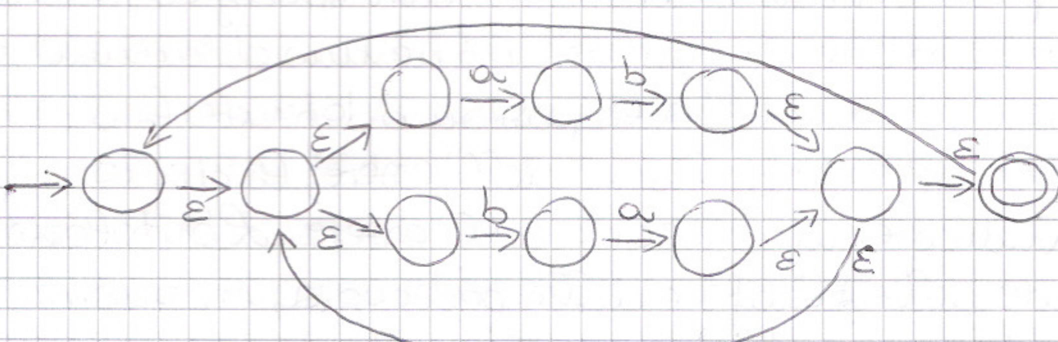
SOL: Posso partire nell'espressione regolare, quando nell'espressione regolare riconosco una stringa, posso costruire l'automa corrispondente. L'automa che accetta ab è

l'automa che accetta ba è:

ora devo farne l'unione. Questi non saranno più stati iniziali, ne creo un altro:



Ora faccio l'automa dello star; creo un nuovo stato iniziale e un altro finale.



Ma questo non lo farete al compito.