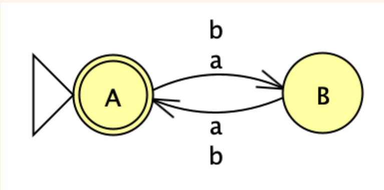
Esercizio 1

A lezione non è stata fatta la dimostrazione, ma si può dimostrare che se due linguaggi sono regolari allora la loro unione è regolare, quindi la risposta è **VERO**.

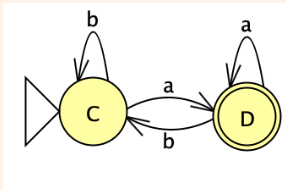
Nota ~~inutile ai fini dell'esercizio~~:

Se prendessimo ad esempio:

- $\mathcal{L}_1 = \{w \in \{a, b\}^* | \ w \ ha \ una \ lunghezza \ pari\}$
- $\mathcal{L}_2 = \{w \in \{a, b\}^* | \ w \ termina \ con \ a\}$

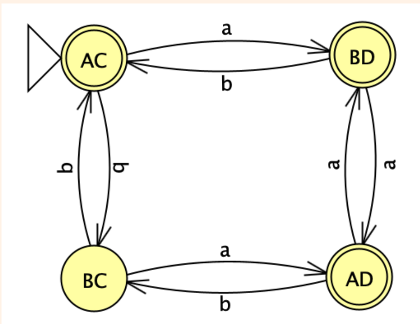


DFA per $\mathcal{L}_1$

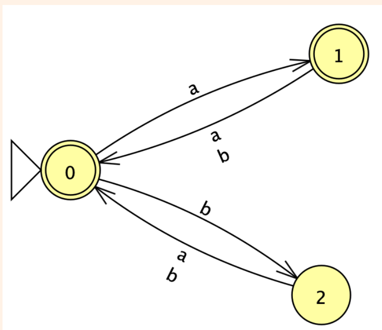


DFA per $\mathcal{L}_2$

L'unione di $\mathcal{L}_1$ con $\mathcal{L}_2$ sta a significare che la parola su $\{a, b\}$ deve essere pari oppure finire con la lettera a. Per costruire l'automa che verifichi l'unione potremo leggere la parola contemporaneamente in entrambi i DFA e se al termine della parola mi trovo in uno stato finale di uno dei due DFA allora la parola appartiene all'unione.
Possiamo costruire il DFA finale quindi facendo il prodotto cartesiano degli stati, quello iniziale sarà AC mentre quelli finali saranno AC, AD e BD ottenendo il seguente DFA, confermando che l'unione è regolare.



Andando ad 'occhio' si poteva comunque costruire un DFA che riconosceva $\mathcal{L}_1 \cup \mathcal{L}_2$:



Esercizio 2

Sulle slide o dispensa studenti.

Esercizio 3

Per la costruzione di un DFA partendo da un NFA partiamo dallo stato iniziale A e calcoliamo la sua ϵ-chiusura:

$$closure(A) = \{A\ , B,\ C,\ E\} = T_0$$

Controllando le transazioni da T_0:

- a-transizione: $closure(D,\ E) = \{D,\ E\} = T_1$
- b-transizione: $closure(A,\ E) = \{A\ , B,\ C,\ E\} = T_0$

Facciamo la stessa cosa da T_1:

- a-transizione: $closure(E) = \{E\} = T_2$
- b-transizione: $closure(A,\ B) = \{A\ , B,\ C,\ E\} = T_0$

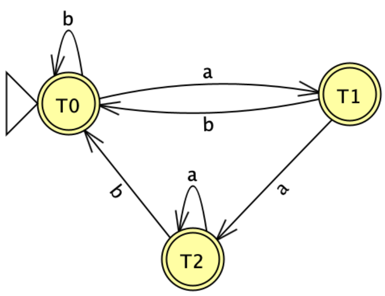
Facciamo la stessa cosa da T_2:

- a-transizione: $closure(E) = \{E\} = T_2$
- b-transizione: $closure(A) = \{A\ , B,\ C,\ E\} = T_0$

La funzione di transazione del DFA sarà:

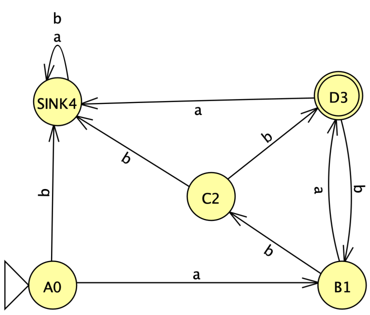
	a	**b**
T_0	T_1	T_0
T_1	T_2	T_0
T_2	T_2	T_0

Quindi se Q è lo stato iniziale del DFA allora: $Q[ab] = T_0 = A, B, C, E$



Esercizio 4

D ha una funzione di transizione parziale. Per renderla totale, aggiungiamo uno stato 'sink' che assorbe tutte le transizioni non definite:



	a	**b**
A	B	$Sink$
B	D	C
C	D	$Sink$
D	$Sink$	B
$Sink$	$Sink$	$Sink$

Funzione di Transizione

Per minimizzare l'automa, analizziamo gli stati di equivalenza usando il metodo di raffinamento successivo:

0-equivalenti	$\{A,\ B,\ C,\ Sink\}$	$\{D\}$			
1-equivalenti	$\{A,\ Sink\}$	$\{B,\ C\}$	$\{D\}$		
2-equivalenti	$\{A\}$	$\{Sink\}$	$\{B\}$	$\{C\}$	$\{D\}$

Spiegazione

- **Stati 0-equivalenti**: si distinguono gli stati finali e non finali
- **Stati 1-equivalenti**:
  - $\{A, Sink\}$ transitano entrambi verso lo stesso insieme $\{A,\ B,\ C,\ Sink\}$ sia con *a-transazione* che con *b-transazione*
  - $\{B, C\}$ transitano entrambi verso lo stesso insieme $\{D\}$ con *a-transazione* e in $\{A,\ B,\ C,\ Sink\}$ con *b-transazione*.
- **Stati 2-equivalenti**:
  - A e $Sink$ vanno divisi perché la *a-transazione* li porta a due insiemi diversi
  - in modo analogo anche B e C vanno separati perché la *b-transazione* li porta a due insiemi diversi

Il DFA è già minimo poiché nessun ulteriore raggruppamento è possibile.
Quindi lo stato a cui corrisponde $P[abab]$ è B.

Esercizio 5

Iniziamo calcolando i $first$ e i $follow$.

First:

- B può derivare ϵ. Pertanto, $First(B) = \{\epsilon\}$
- A può derivare $BcBaA$ quindi aggiungiamo i $First(B)$ ai $First(A)$, $First(B)$ contiene ϵ quindi consideriamo $First(cBaA)$ e aggiungiamo c ai $First(A)$, A può anche derivare ϵ quindi aggiungiamo anche ϵ
- S può derivare AaB quindi aggiungiamo i $First(A)$ ai $first(S)$, essendoci ϵ nei $first(A)$ allora dobbiamo aggiungere $first(aB)$ ai $first(S)$, infine S può derivare b che sarà da aggiungere ai suoi $first$

	first	follow
S	b a c	\$
A	ϵ c	a
B	ϵ	c a \$

Follow:

- $Follow(S)$: S è lo start symbol, quindi $Follow(S) = \{\$\}$, S non compare in nessun body quindi ci possiamo fermare
- $Follow(A)$: nella produzione $S \to AaB$, A è seguito da 'aB', quindi $First(aB) = \{a\}$ è in $Follow(A)$
- $Follow(B)$:
  - nella produzione $S \to AaB$, B è l'ultimo simbolo della produzione dobbiamo aggiungere i $Follow(S)$ ai $Follow(B)$
  - nella produzione $A \to BcBaA$, B è seguito da 'cBaA', quindi $First(cBaA) = \{c\}$ è in $Follow(B)$

Le regole per popolare la tabella di parsing LL(1) sono:

- per ogni produzione $A \to \alpha$ nella grammatica:
  - per ogni terminale 'b' in $First(\alpha)$ (escludendo ϵ) bisogna aggiungere $A \to \alpha$ alla voce della tabella $M[A, b]$.
  - se ϵ è in $First(\alpha)$ allora per ogni terminale 'x' in $Follow(A)$ bisogna aggiungere $A \to \alpha$ alla voce della tabella $M[A, x]$.
- imposta tutte le voci vuote nella tabella a 'errore'

La tabella di parsing $LL(1)$ è la seguente e la risposta è evidenziata:

	a	**b**	**c**	**$**
S	$S \to AaB$	$S \to b$	$S \to AaB$	
A	$A \to \epsilon$		$A \to BcBaA$	
B	$B \to \epsilon$		$B \to \epsilon$	$B \to \epsilon$