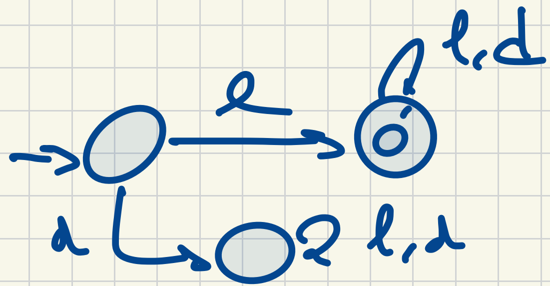
1 St October

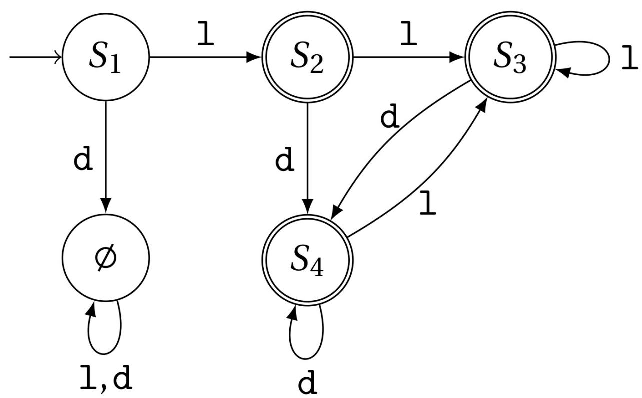
$l(l+d)^*$

DFA



Smaller

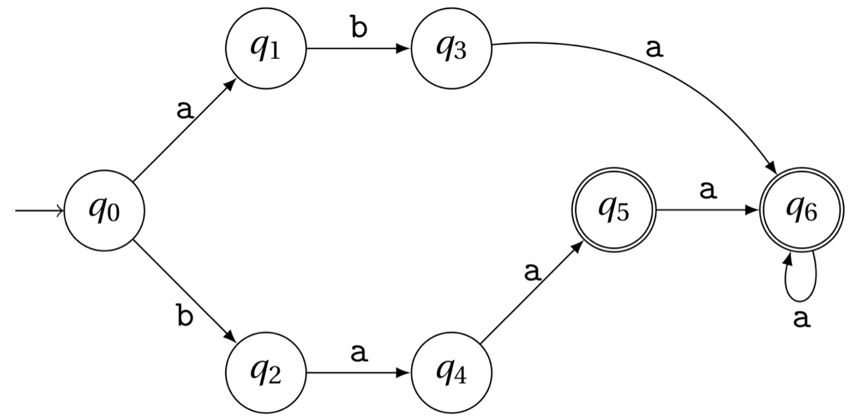
Smallest!



bigger

Vin-

Minimisation of DFA



Can we merge q_5 and q_6 ?

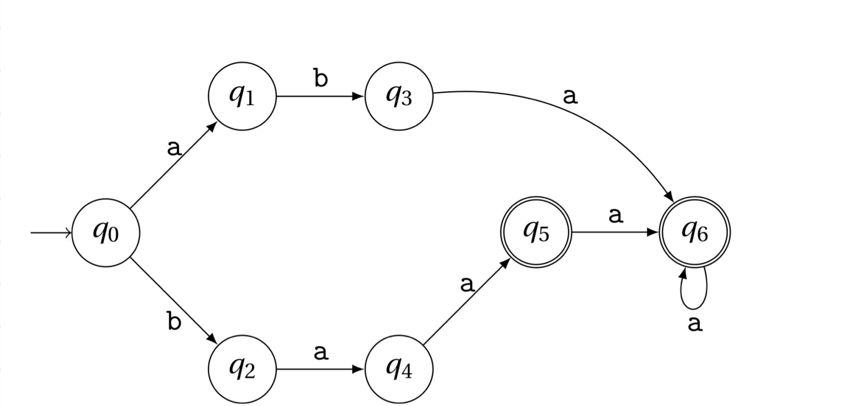
If q_5 is initial → the automaton accepts a^*

→ q_6 a^*

What about q_3, q_5, q_6 ? → language aa^*

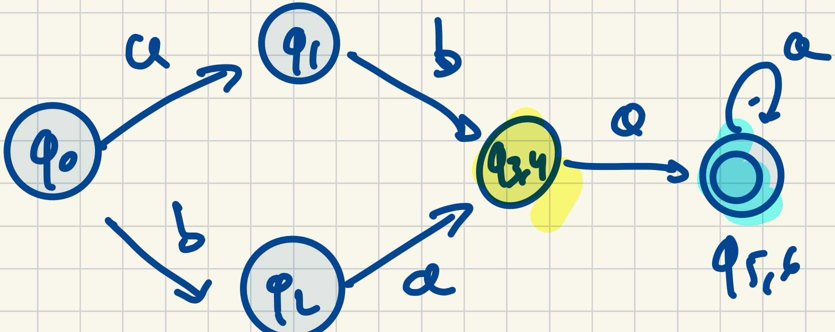
We need a notion of accepted language from a state q = all the words accepted if q becomes the initial state.

We merge states iff they accept the same language.



$(ab + ba) \cdot a^+$

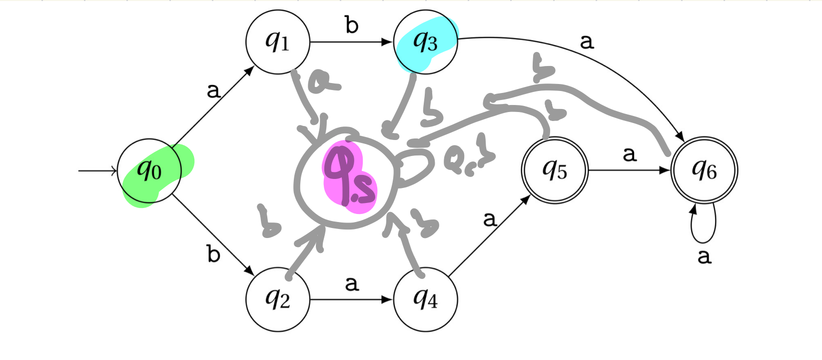
State	L_q
q_0	$ab\,a^+ + ba\,a^+$
q_1	$b\,a^+$
q_2	$a\,a^+$
q_3	$a\,a^* = a^+$
q_4	$a\,a^* = a^+$
q_5	a^*
q_6	a^*



Minimisation algorithm

We build a matrix P indexed by the states.
s.t. $P[q_i, q_j] = 1$ iff q_i accept the same language as q_j.

We will build this matrix by refining an initial matrix where the accepting states are not equivalent to the non-accepting ones.



	q_1	q_2	q_3	q_4	q_5	q_6	q_s
q_0	1	1	~~1~~ 0	1	0	0	1
q_1		1	1	1	0	0	1
q_2			1	1	0	0	1
q_3				1	0	0	1
q_4					0	0	1
q_5						1	0
q_6							0

To refine:

? q_0 and q_3

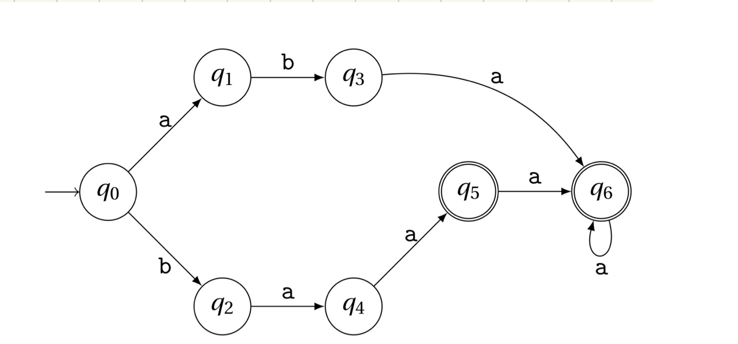
$$q_0 \xrightarrow{a} q_1$$

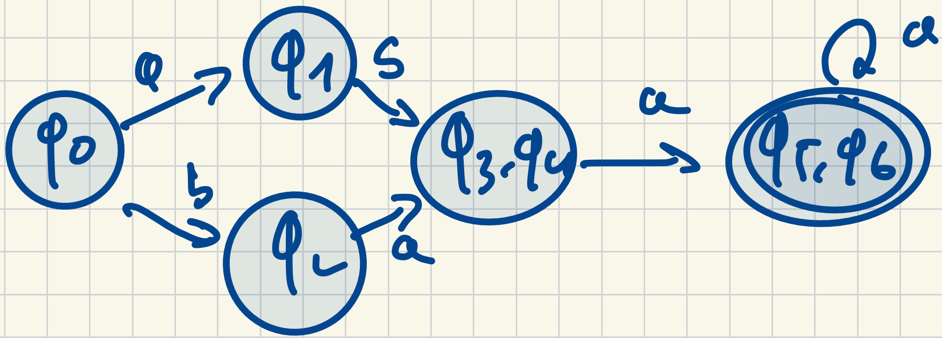
$$q_3 \xrightarrow{a} q_6$$

q_0 not equivalent to q_3

Eventually....

q_1	q_2	q_3	q_4	q_5	q_6	q_s	
0	0	0	0	0	0	**0**	q_0
	0	0	0	0	0	0	q_1
		0	0	0	0	0	q_2
			1	0	0	0	q_3
				0	0	0	q_4
					1	0	q_5
						0	q_6

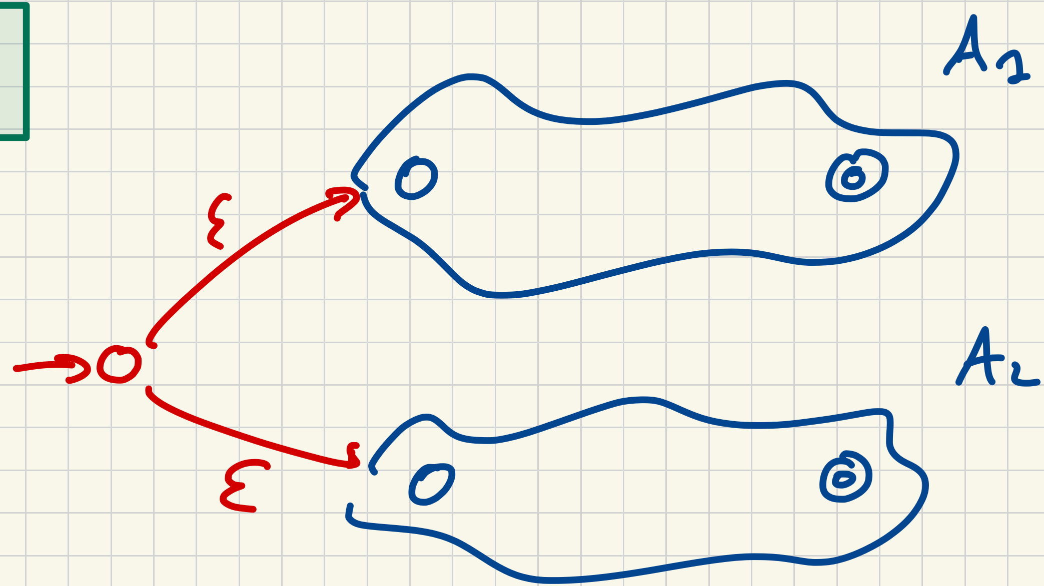




Operations on regular languages

$L_1, L_2 \rightsquigarrow A_1, A_2$

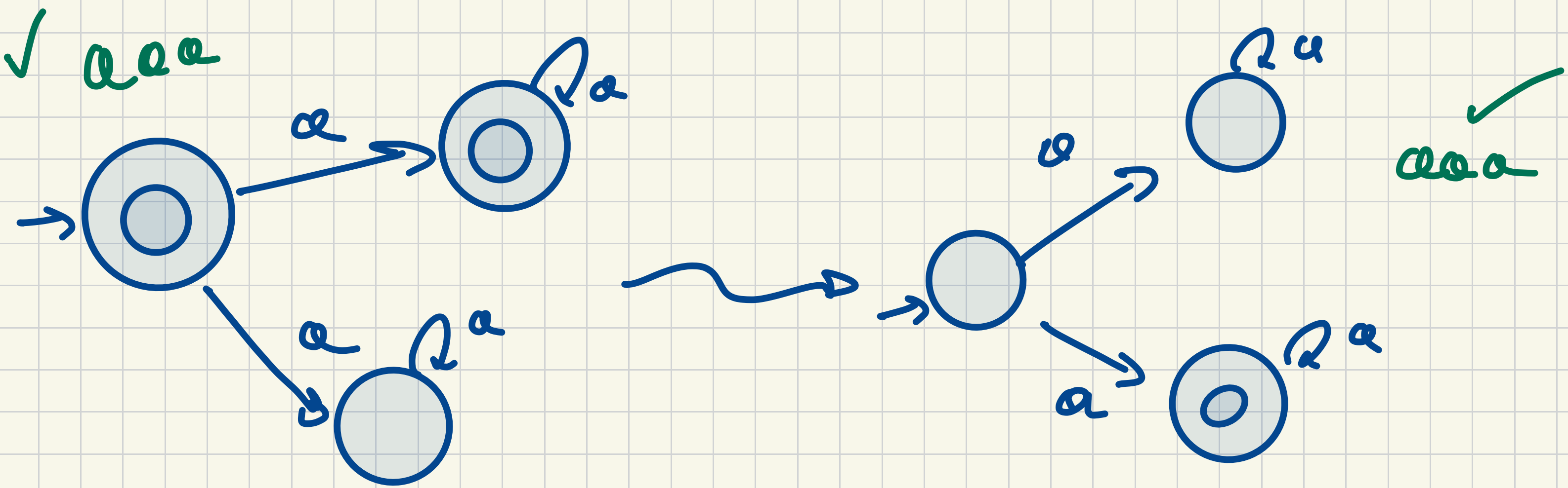
$\boxed{L_1 \cup L_2}$



$\overline{L_1} \longrightarrow$ Complement.

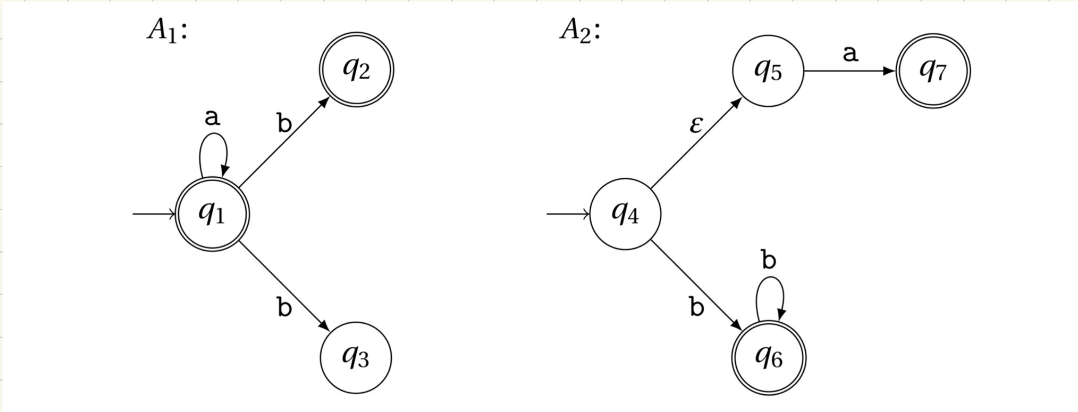
Works if the automaton is a DFA.

1st idea : Swap accepting and non-accepting states.



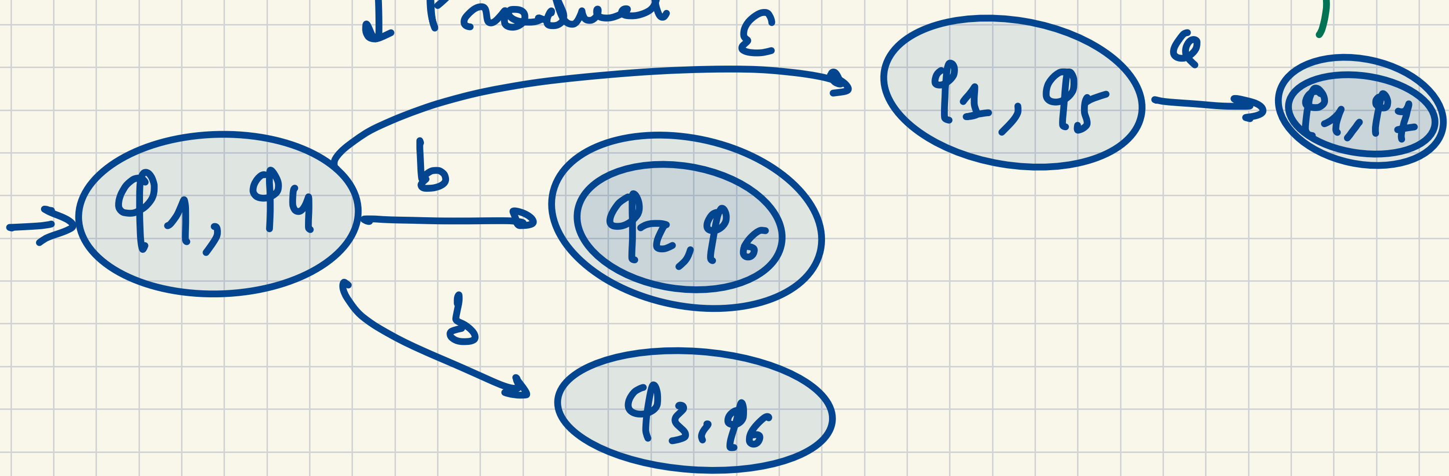
$a^*.(b+\varepsilon)$ $\qquad$ $a+b^+$

Intersection



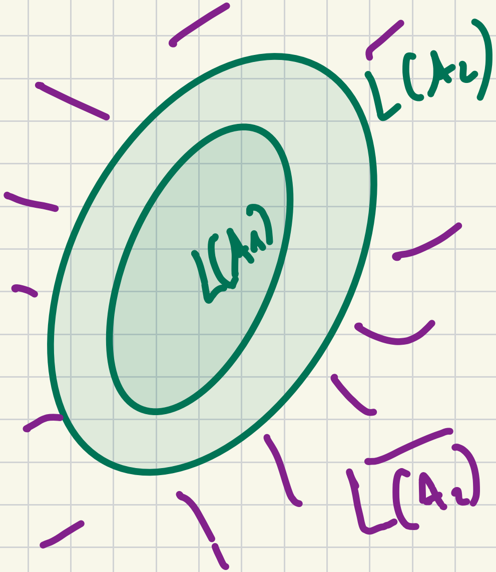
Product

Both must be accepting



Inclusion:

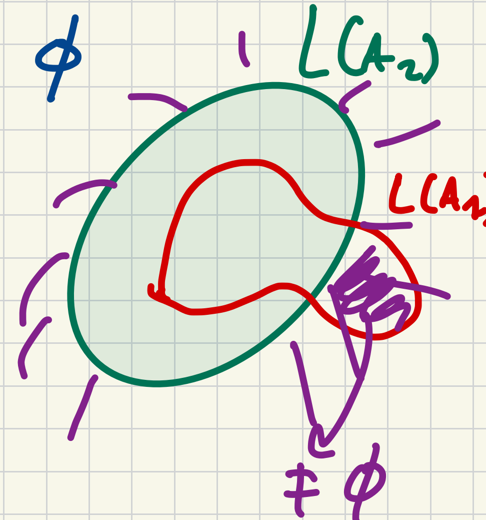
$$L_1 \overset{?}{\subseteq} L_2$$



$$L(A_1) \subseteq L(A_2)$$

$$\iff$$

$$L(A_1) \cap \overline{L(A_2)} = \emptyset$$



Equality?

$$L(A_1) = L(A_2)$$

$$\Leftrightarrow$$

$$L(A_1) \subseteq L(A_2)$$

and

$$L(A_2) \subseteq L(A_1)$$

Not all languages are regular

$L_{()}$ = all the well-parenthesised words.

is not regular.

Proof We assume that $L_{()}$ is regular

Then, there is a finite automaton $A_{()}$ that accepts $L_{()}$. Let us assume that $A_{()}$ has n states.

$$w = \underbrace{(\cdots (}_{n} \underbrace{) \cdots)}_{n} \in L_{()}$$

$$\in L(A_{()})$$

$$w = \underbrace{(\cdots(}_{n}\ \underbrace{)\cdots)}_{n} \in L_{()}$$

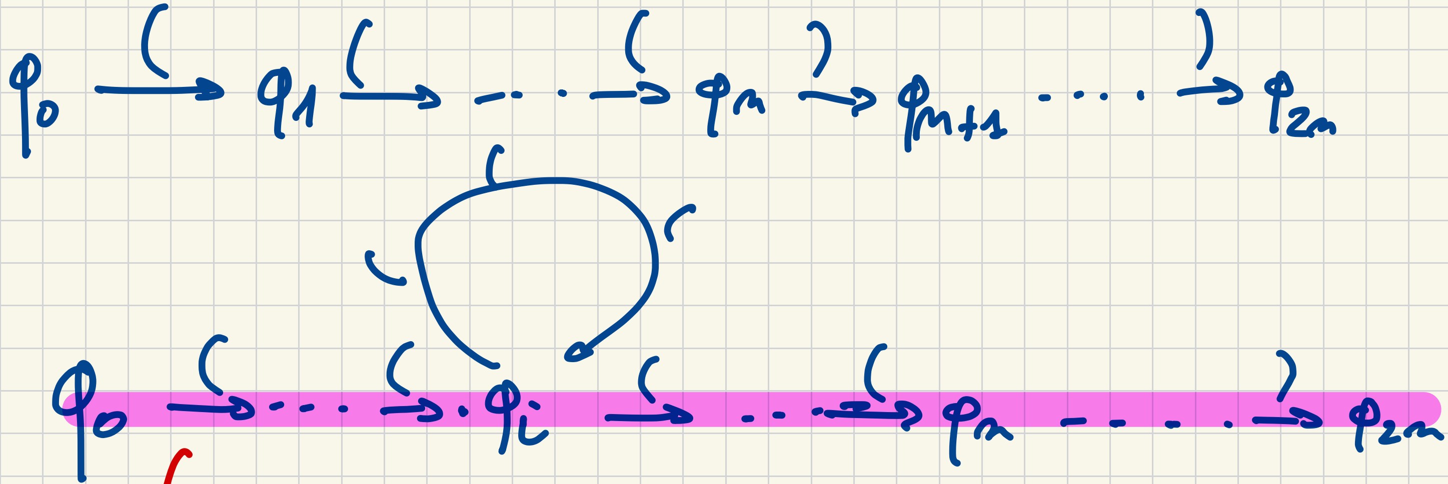
$$\in L(A_{()})$$

Let's look at an accepting run of $L(A_{()})$ on w

$$q_0 \xrightarrow{(} q_1 \xrightarrow{(} \cdots \xrightarrow{(} q_n \xrightarrow{)} q_{n+1} \cdots \xrightarrow{)} q_{2n}$$

number of state visited : $n+1$

↳ we need to see twice the same state!!



the purple path exists and accepts $(^m)^n$ $m < n$

$$L(A_{()}) \neq L_{()}$$