Properties of Regular Languages

For regular languages L_1 and L_2 we will prove that:

$$\left.\begin{array}{rl} \text{Union:} & L_1 \cup L_2 \\ \text{Concatenation:} & L_1 L_2 \\ \text{Star:} & L_1{}^* \\ \text{Reversal:} & L_1{}^R \\ \text{Complement:} & \overline{L_1} \\ \text{Intersection:} & L_1 \cap L_2 \end{array}\right\} \text{Are regular Languages}$$

We say: Regular languages are **closed under**

Union: $L_1 \cup L_2$

Concatenation: $L_1 L_2$

Star: $L_1{}^*$

means empty string is accepted

Reversal: $L_1{}^R$

you'll reverse it by making the accepting state the initial state

learn how to take the

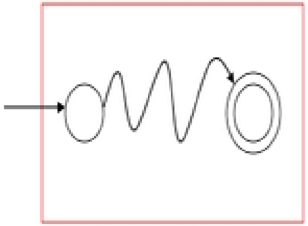
Complement: $\overline{L_1}$

Intersection: $L_1 \cap L_2$

Regular language L_1

$L(M_1) = L_1$

NFA M_1

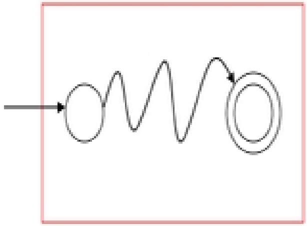


Single accepting state

Regular language L_2

$L(M_2) = L_2$

NFA M_2

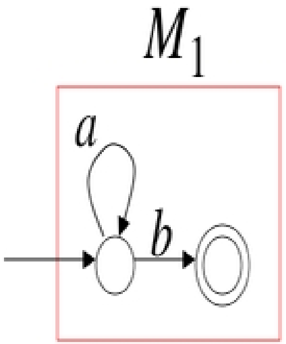


Single aceepting state

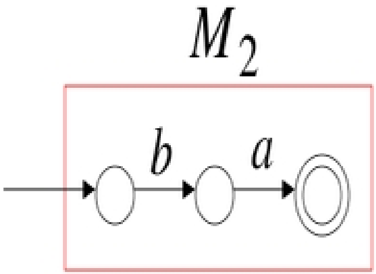
Example

$n \geq 0$

$L_1 = \{a^n b\}$

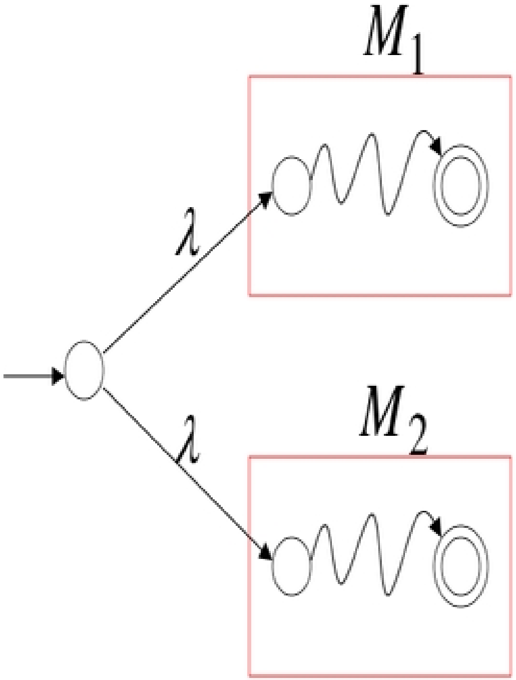


$L_2 = \{ba\}$



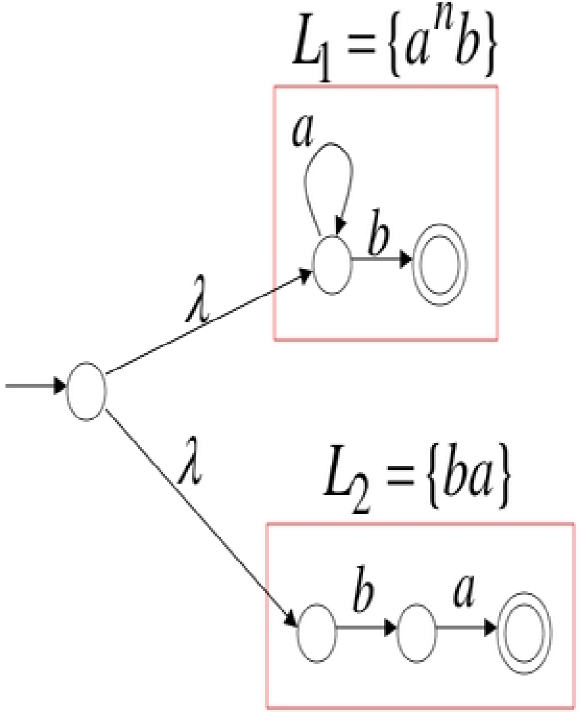
Union

NFA for $L_1 \cup L_2$



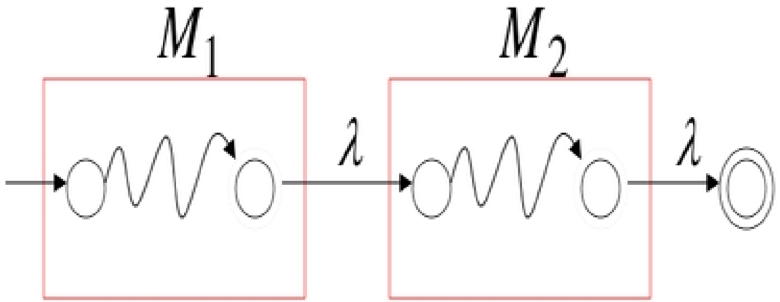
Example

NFA for $L_1 \cup L_2 = \{a^n b\} \cup \{ba\}$



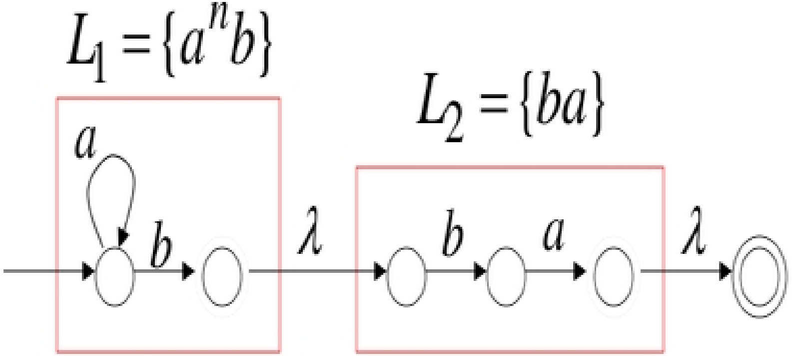
Concatenation

NFA for $L_1 L_2$



Example

NFA for $L_1 L_2 = \{a^n b\}\{ba\} = \{a^n bba\}$

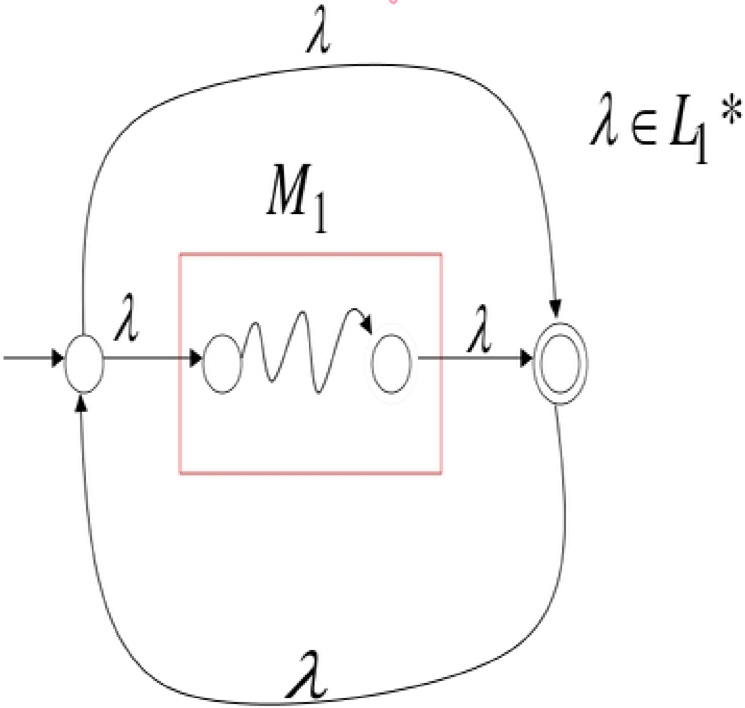


Star Operation

star means empty string is accepted

NFA for $L_1{}^*$

initial state must be an accepting state

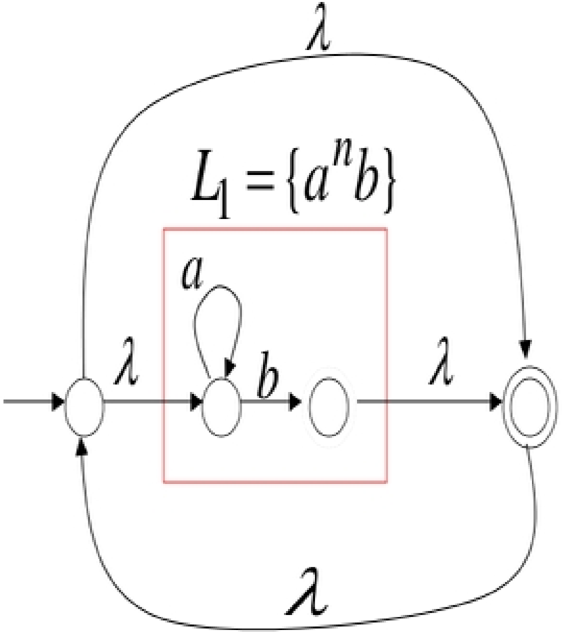


Example

NFA for $L_1^* = \{a^n b\}^*$

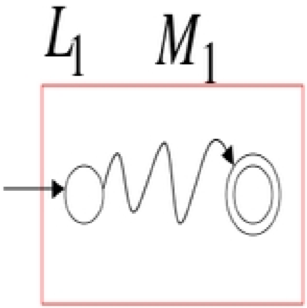
$w = w_1 w_2 \cdots w_k$

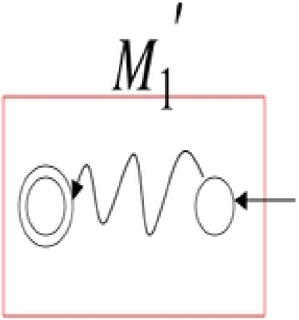
$w_i \in L_1$



Reverse

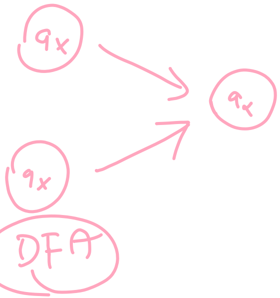
NFA for L_1^R



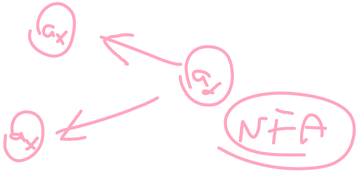


the assignment actually

*

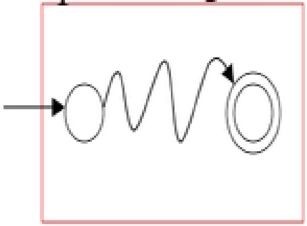


when you reverse this you'll get

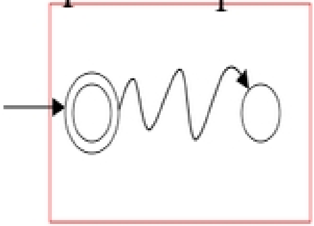


Complement

L_1 M_1



$\overline{L_1}$ M_1'



ex trap state will be the initial? state and initial? state will be the trap state

Intersection

L_1 regular

L_2 regular

We show

$L_1 \cap L_2$

regular

Proof?

L_1, L_2 regular

$\Rightarrow \overline{L_1}, \overline{L_2}$ regular

$\Rightarrow \overline{L_1} \cup \overline{L_2}$ regular

$\Rightarrow \overline{\overline{L_1} \cup \overline{L_2}}$ regular

DeMorgan's Law: $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

$\Rightarrow L_1 \cap L_2$ regular

Example

$L_1 = \{a^n b\}$ regular

$L_2 = \{ab, ba\}$ regular

$\Rightarrow L_1 \cap L_2 = \{ab\}$

regular

Another Proof for Intersection Closure

Machine M_1

FA for L_1

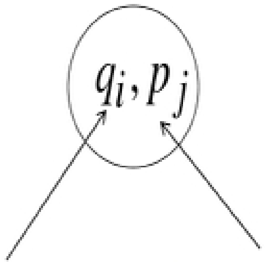
Machine M_2

FA for L_2

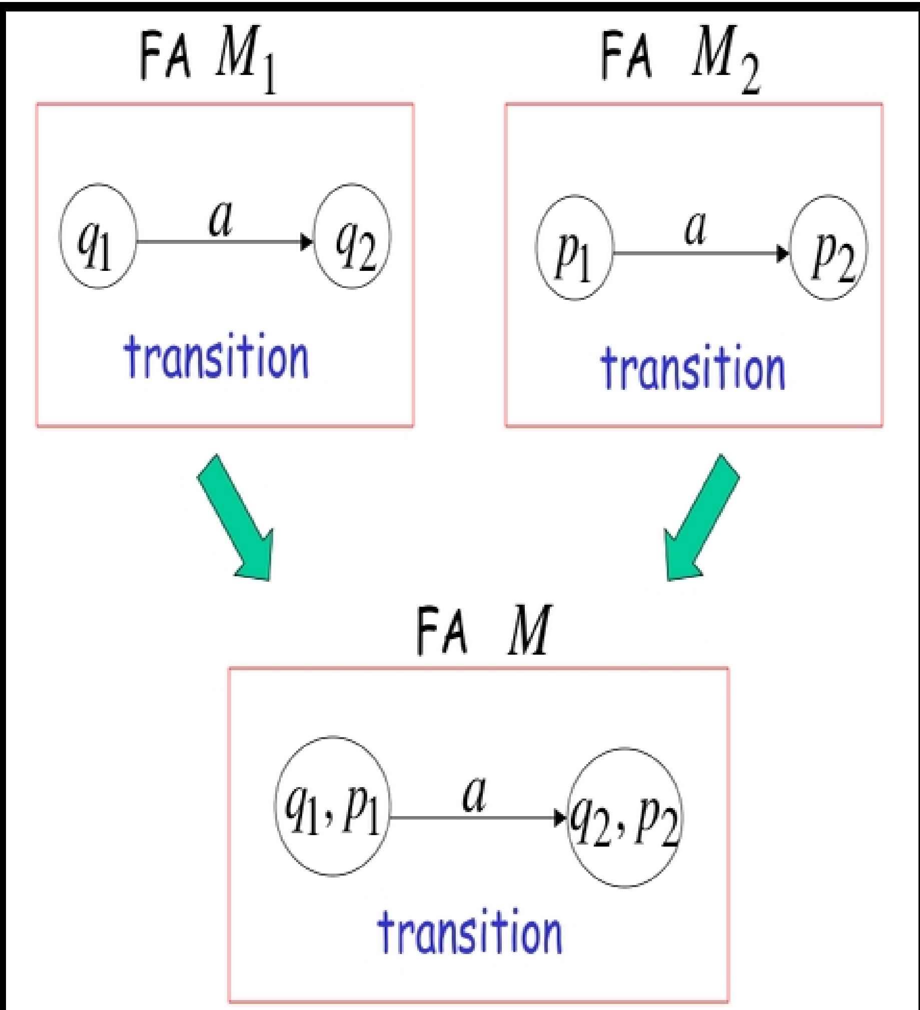
Construct a new FA M that accepts $L_1 \cap L_2$

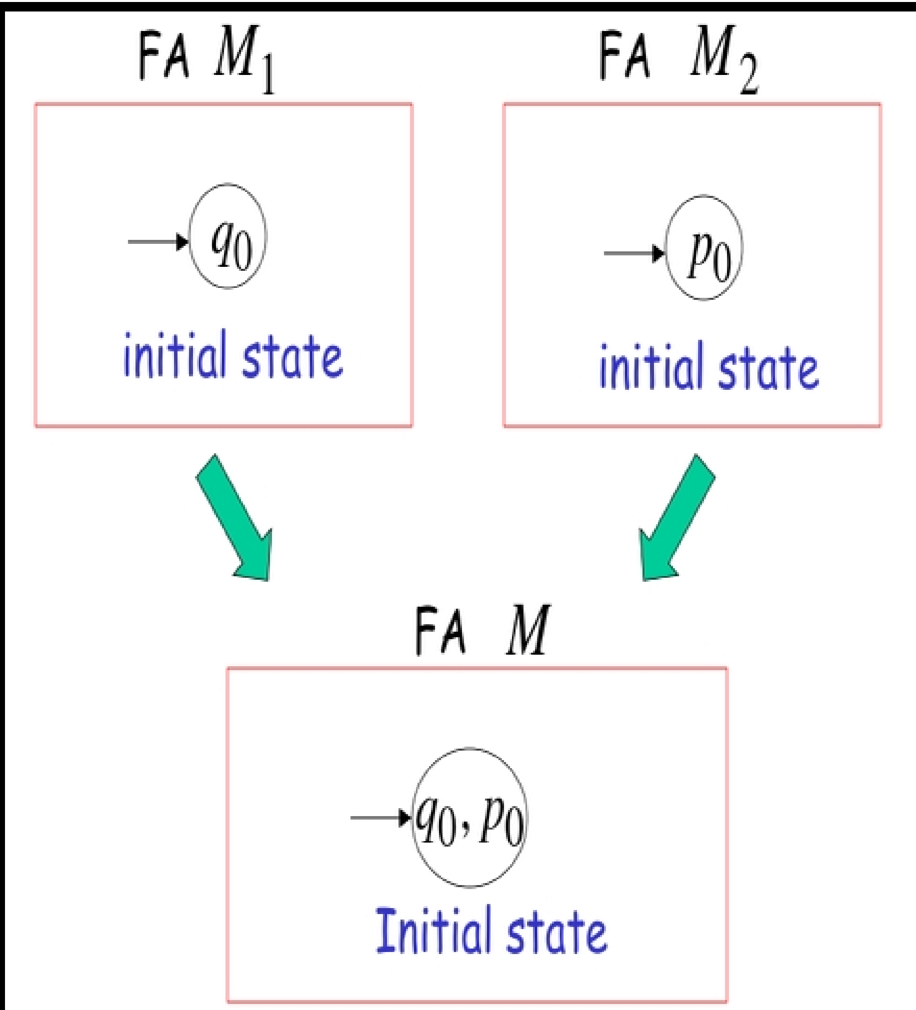
M simulates in parallel M_1 and M_2

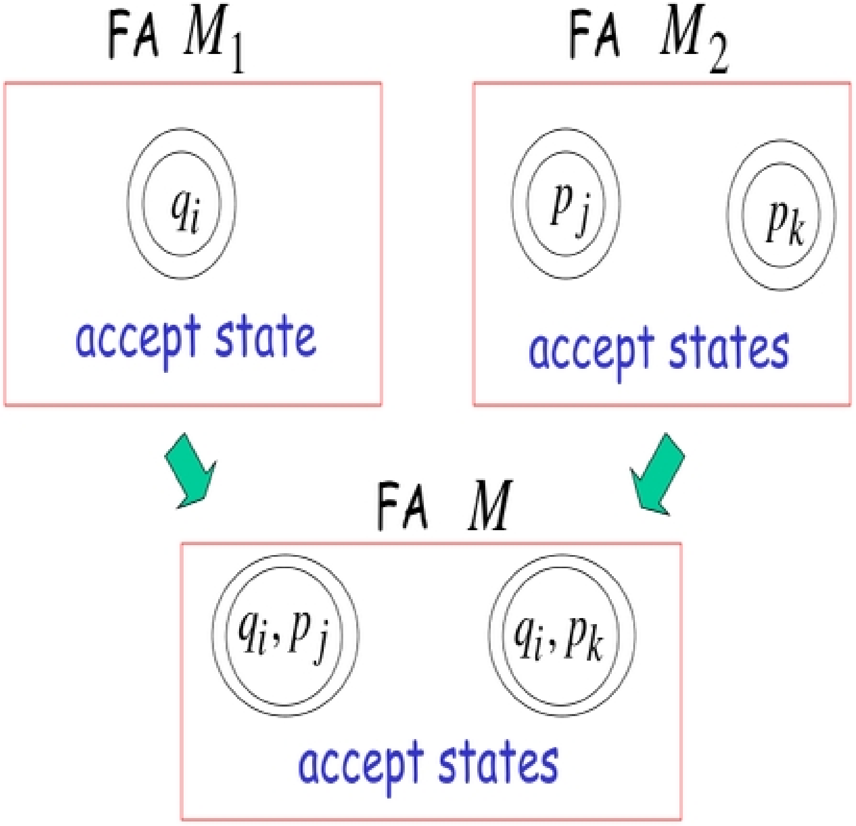
States in M



State in M_1 State in M_2


FA M_1
FA M_2
q_1
a
q_2
transition
p_1
a
p_2
transition
FA M
q_1, p_1
a
q_2, p_2
transition

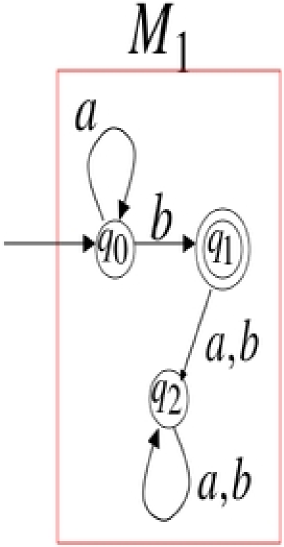

FA M_1
FA M_2
q_0
initial state
p_0
initial state
FA M
q_0, p_0
Initial state

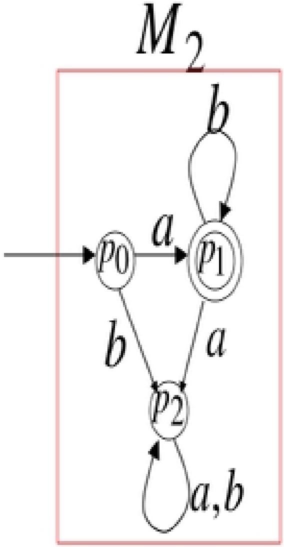


Both constituents must be accepting states

Example:

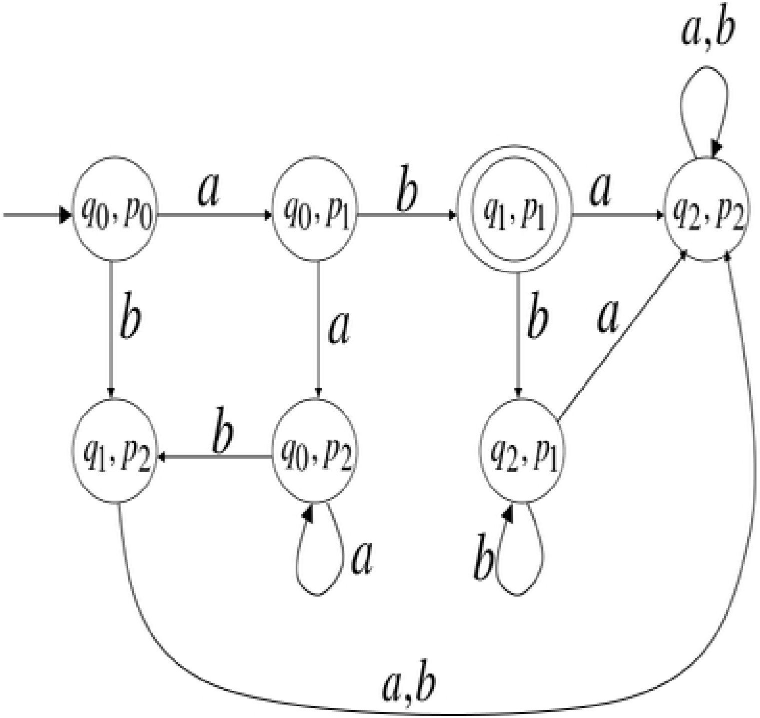
$$L_1 = \{a^n b\} \quad n \geq 0 \qquad\qquad L_2 = \{ab^m\} \quad m \geq 0$$





Automaton for intersection

$$L = \{a^n b\} \cap \{ab^n\} = \{ab\}$$



M simulates in parallel M_1 and M_2

M accepts string w if and only if

M_1 accepts string w and

M_2 accepts string w

$$L(M) = L(M_1) \cap L(M_2)$$