CS116-Automata Theory and Formal Languages

Lecture 4

Properties of Regular Languages

Computer Science Department

1st Semester 2025-2026

For regular languages L_1 and L_2
we will prove that:

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

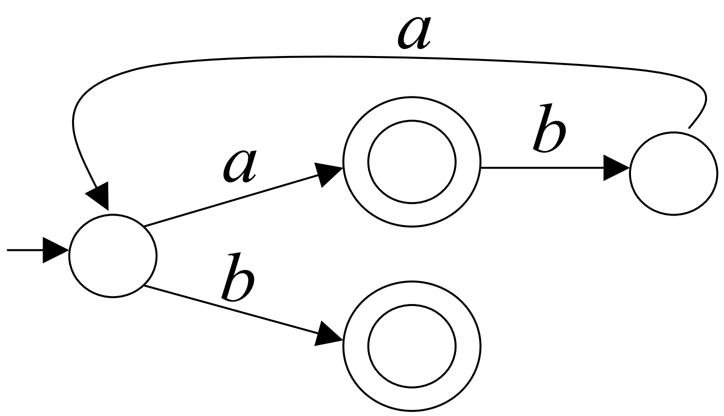
Are regular
Languages

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

Union:	$L_1 \cup L_2$
Concatenation:	$L_1 L_2$
Star:	$L_1{}^*$
Reversal:	$L_1{}^R$
Complement:	$\overline{L_1}$
Intersection:	$L_1 \cap L_2$

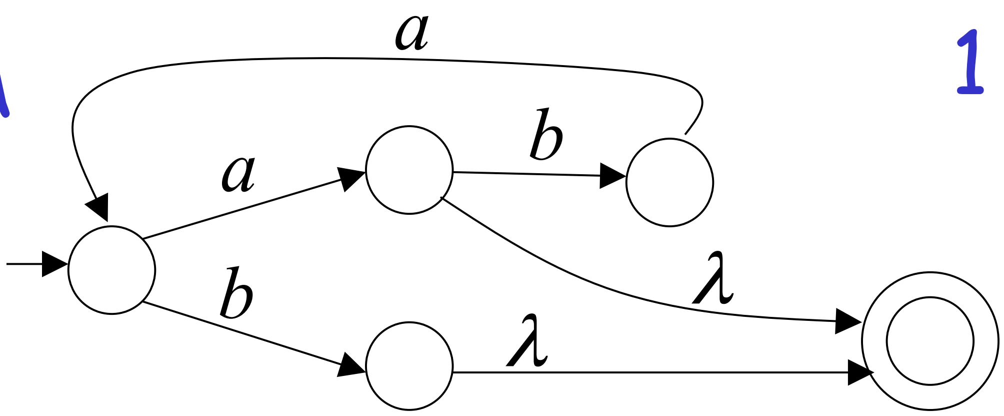
A useful transformation: use one accept state

NFA



2 accept states

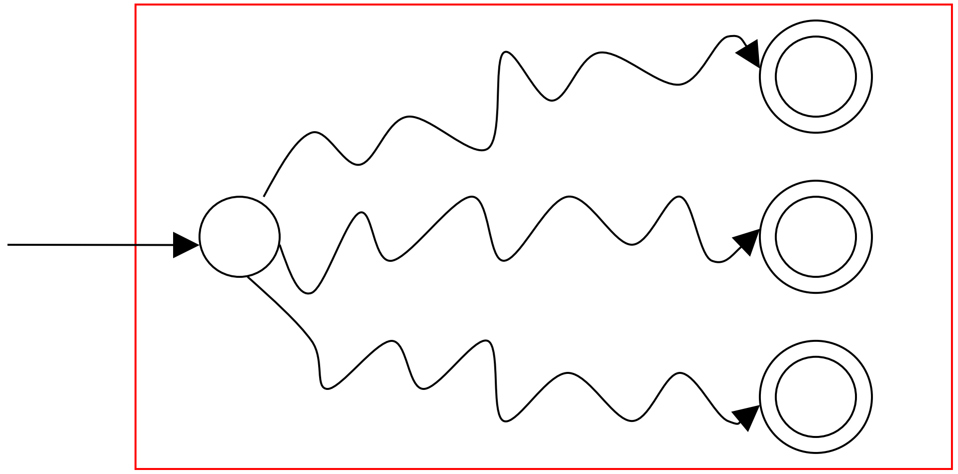
Equivalent NFA



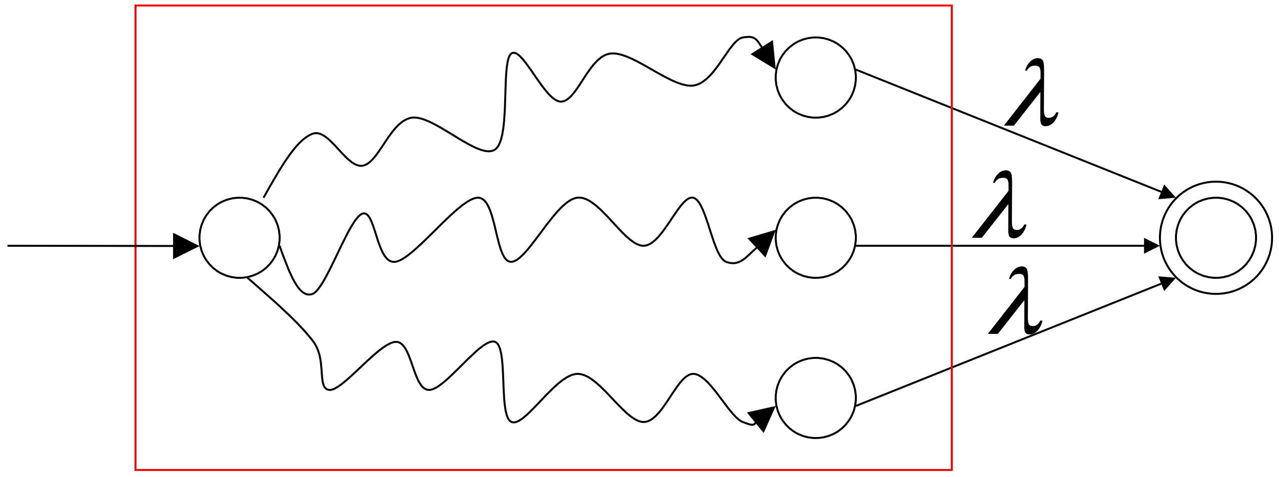
1 accept state

In General

NFA



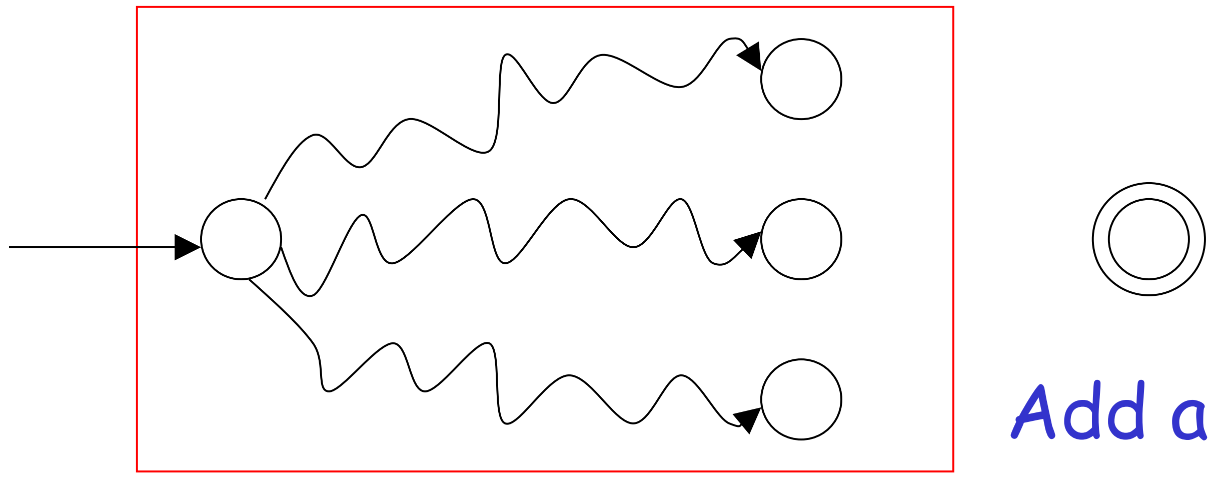
Equivalent NFA



Single accepting state

Extreme case

NFA without accepting state



Add an accepting state without transitions

Take two languages

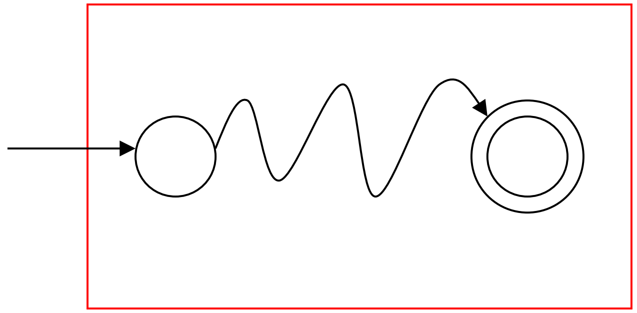
Regular language L_1

Regular language L_2

$$L(M_1) = L_1$$

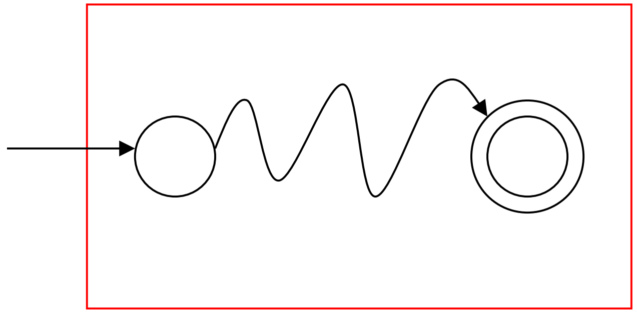
$$L(M_2) = L_2$$

NFA M_1



Single accepting state

NFA M_2



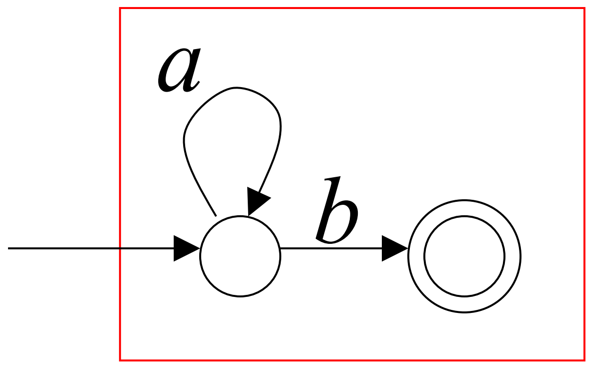
Single accepting state

Example

$n \quad 0$

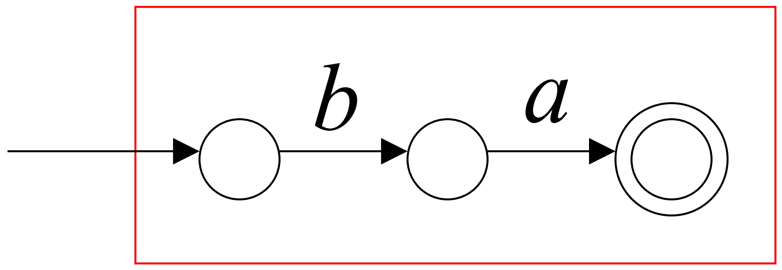
$L_1 = \{a^n b\}$

M_1



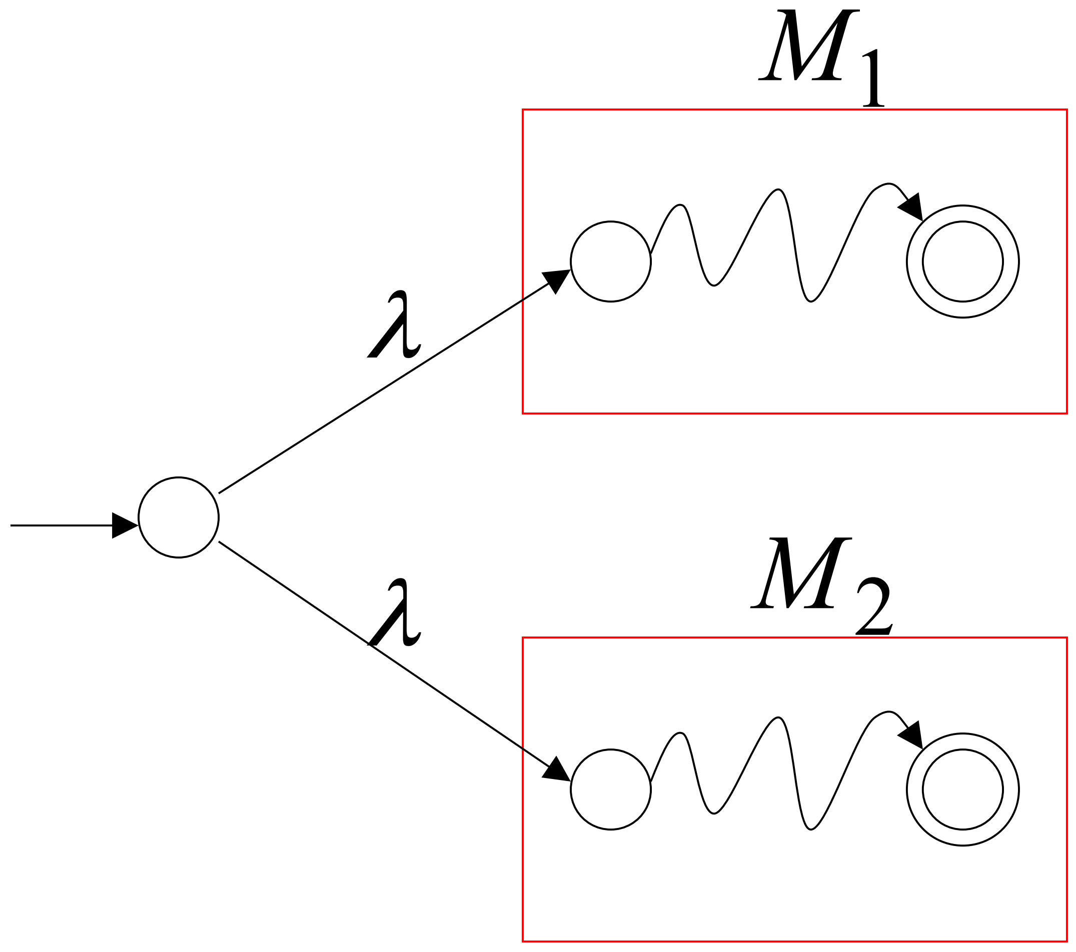
$L_2 = \{ba\}$

M_2



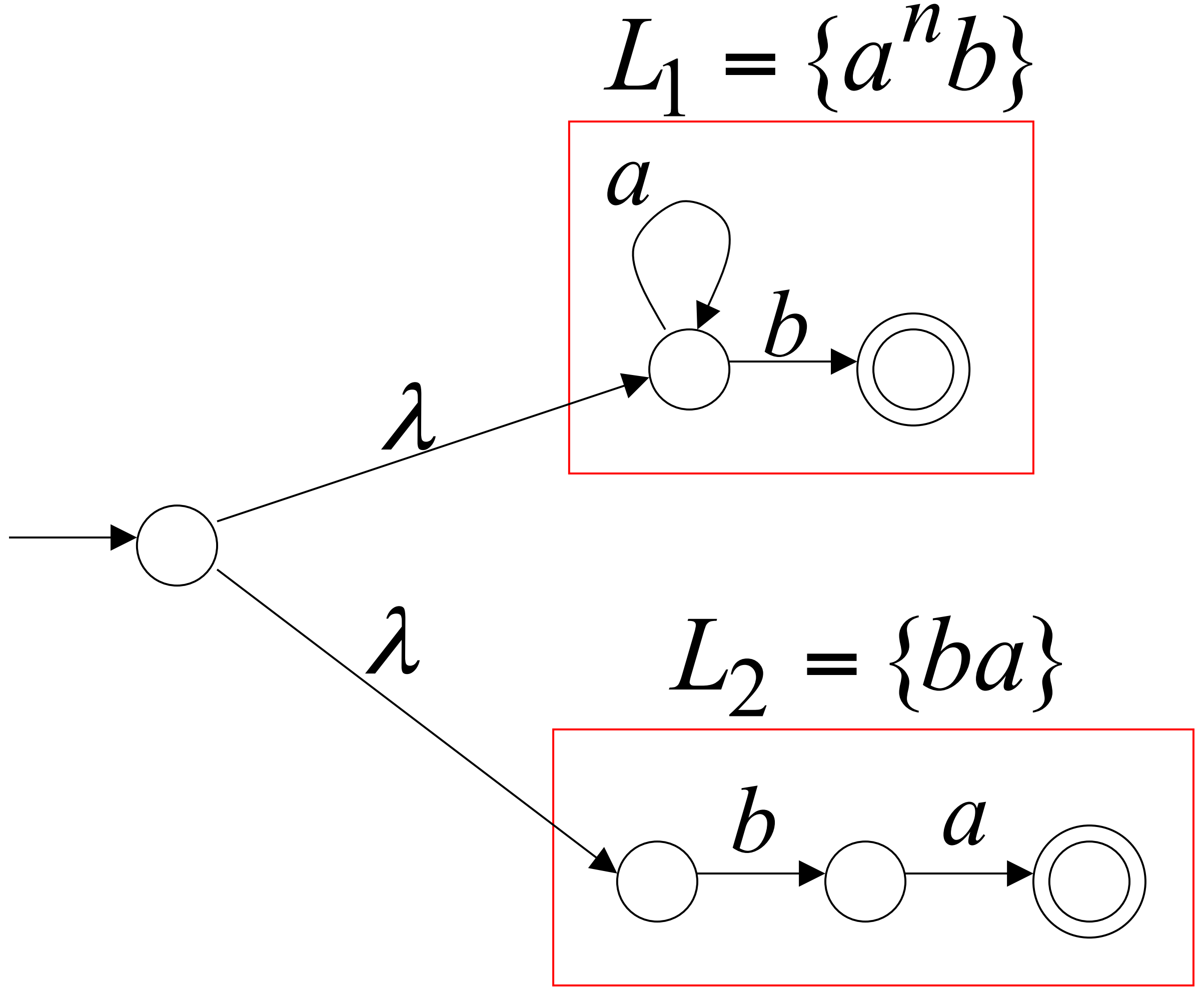
Union

NFA for $L_1 \cup L_2$



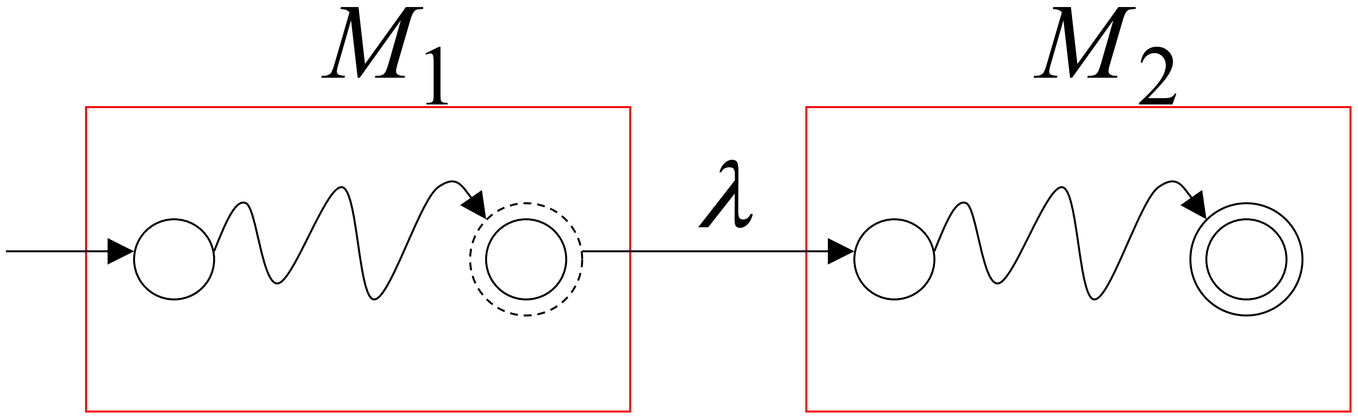
Example

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



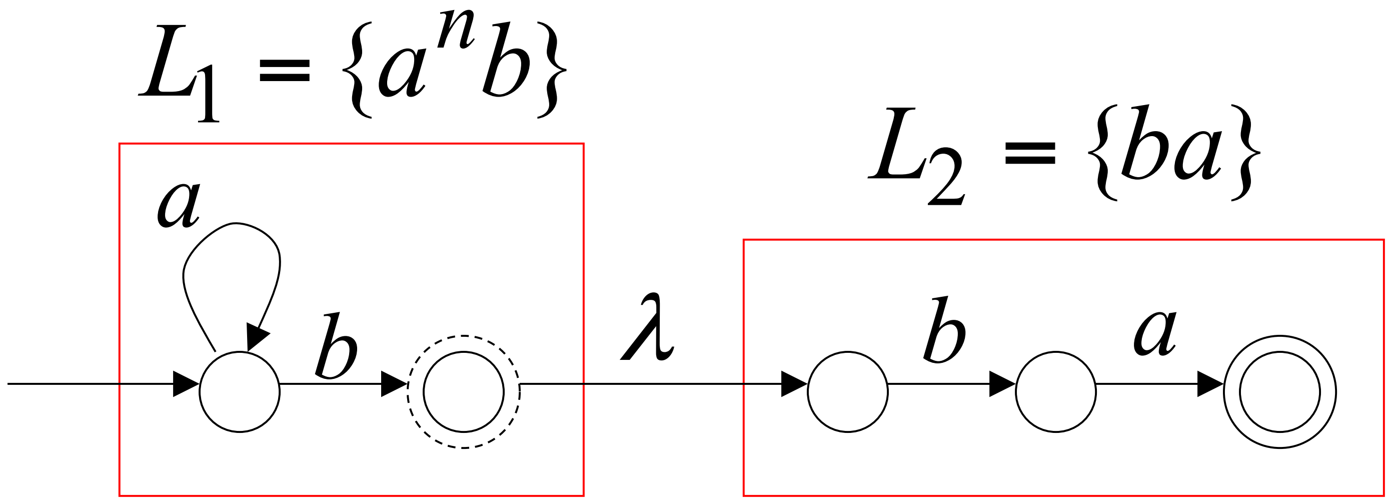
Concatenation

NFA for $L_1 L_2$



Example

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

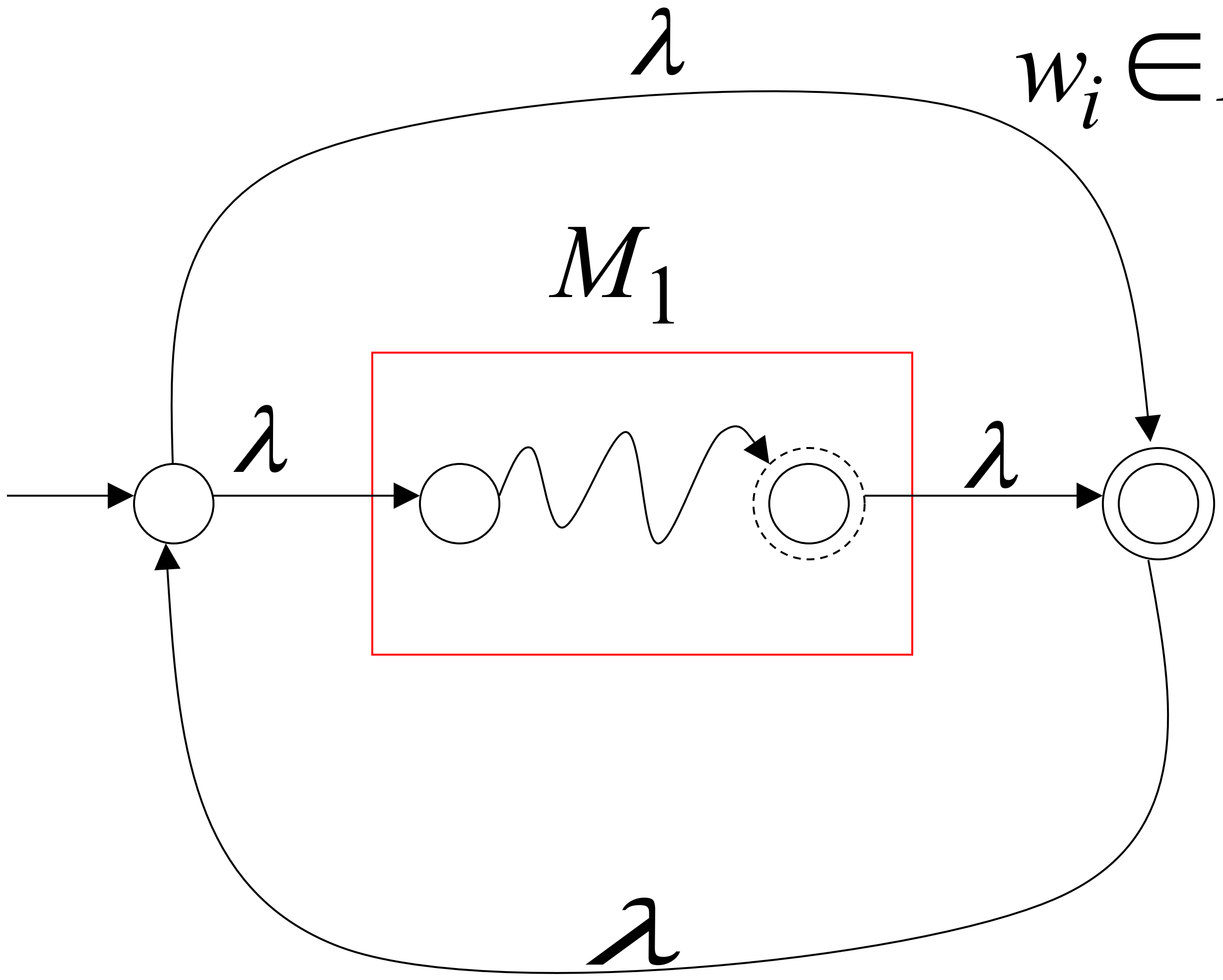


Star Operation

NFA for $L_1{}^*$

$w = w_1 w_2 \cdots w_k$

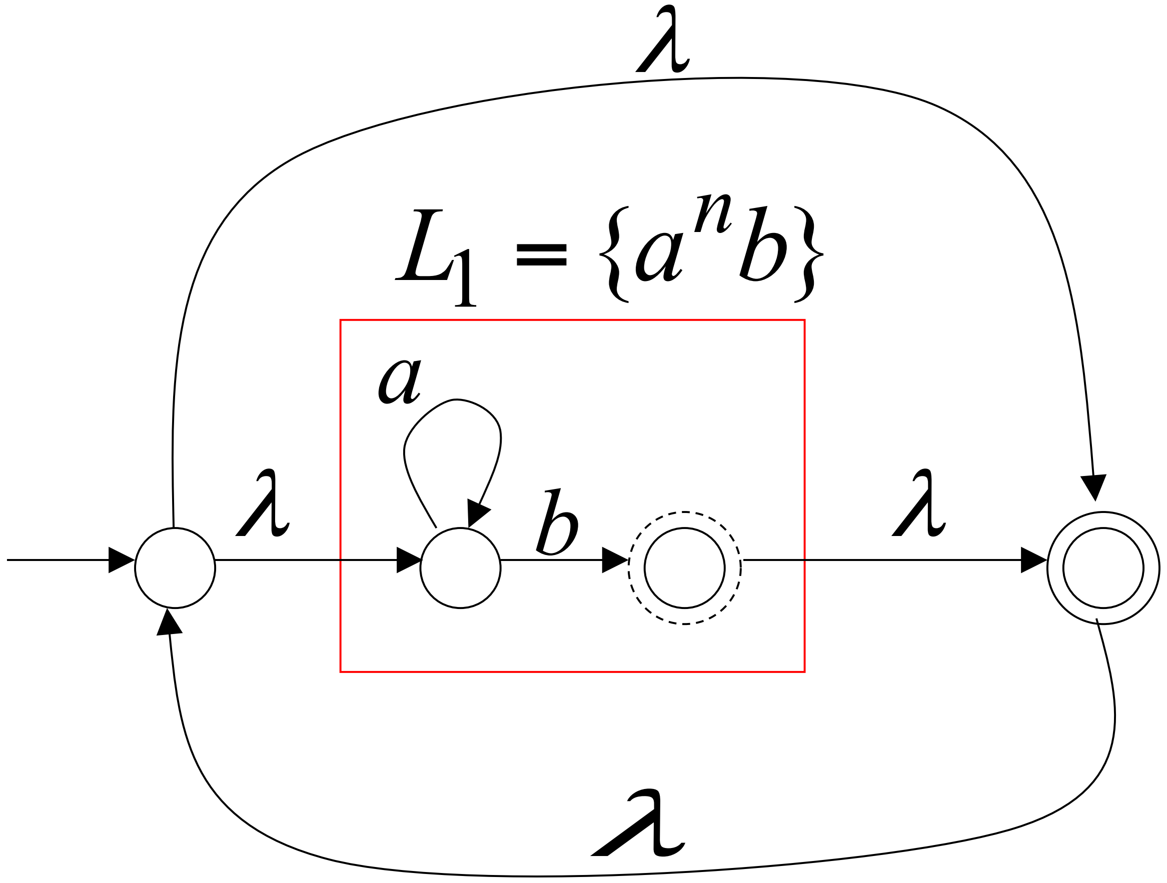
$w_i \in L_1$



$\lambda \in L_1{}^*$

Example

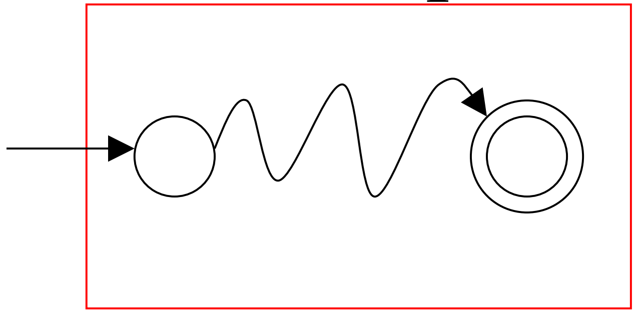
NFA for $L_1{}^* = \{a^n b\}^*$



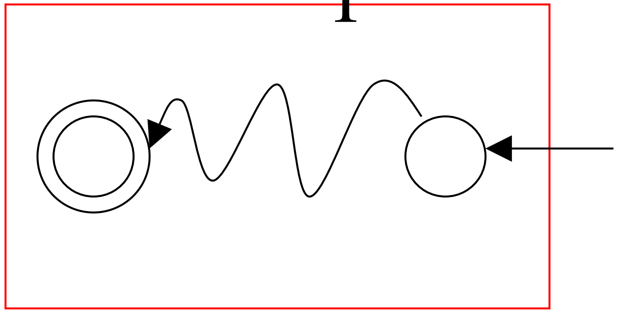
Reverse

NFA for $L_1{}^R$

L_1 M_1



$M_1{}'$



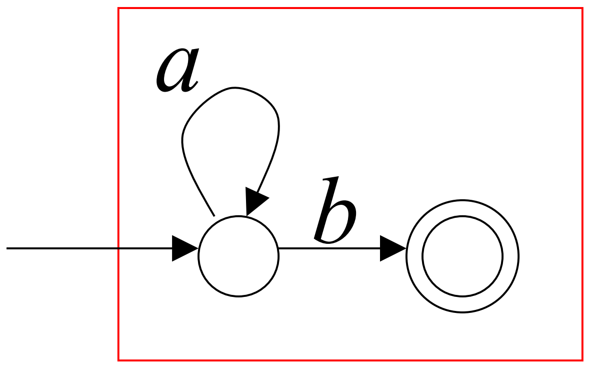
1. Reverse all transitions

2. Make initial state accepting state and vice versa

Example

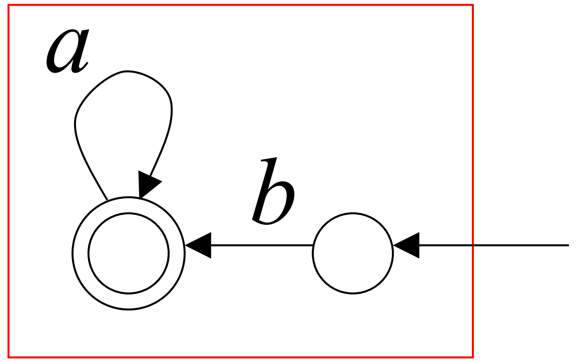
M_1

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

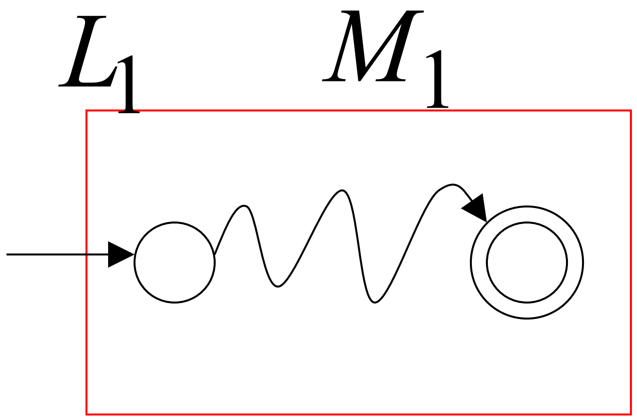


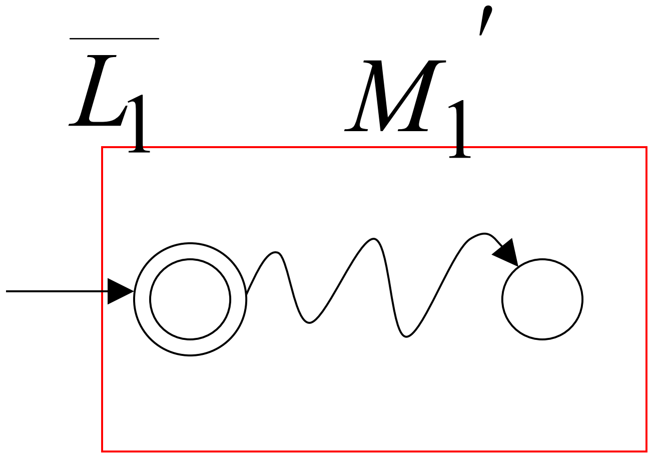
M_1'

$L_1^R = \{ba^n\}$



Complement



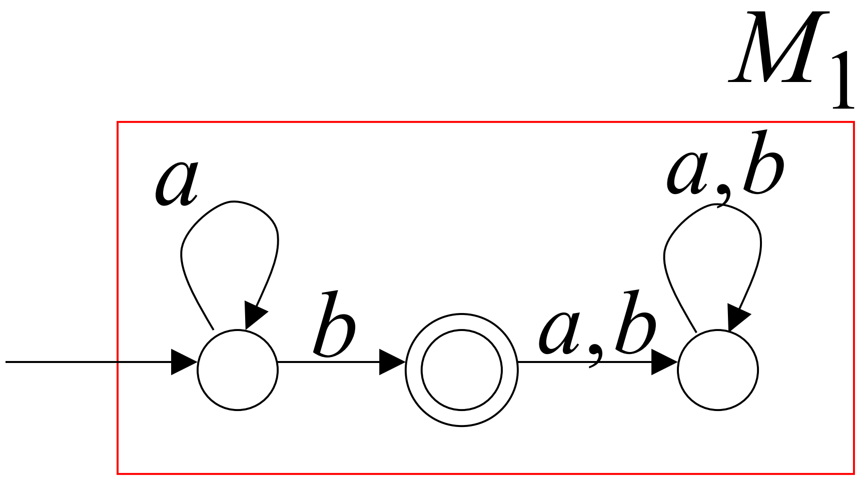


1. Take the DFA that accepts L_1

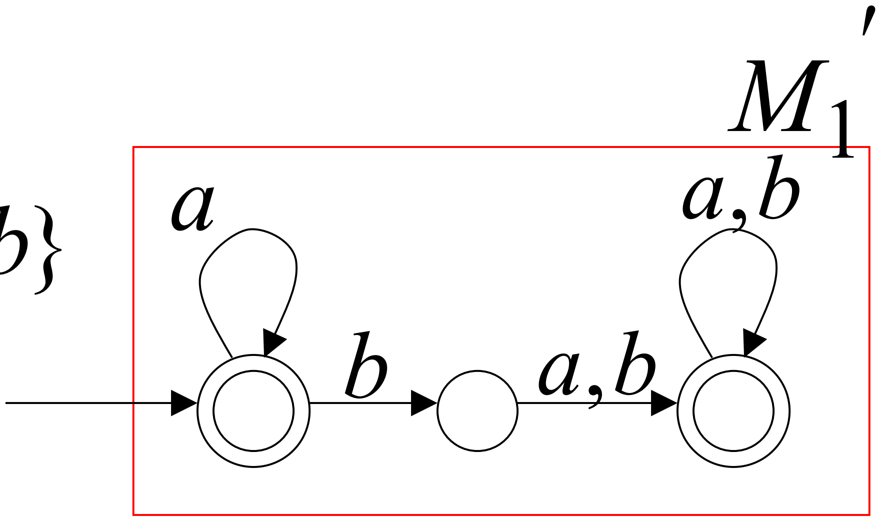
2. Make accepting states non-final, and vice-versa

Example

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



$$\overline{L_1} = \{a,b\}^* - \{a^n b\}$$



Intersection

$$\left.\begin{array}{l} L_1 \text{ regular} \\ \\ L_2 \text{ regular} \end{array}\right\} \xrightarrow{\text{We show}} \begin{array}{c} L_1 \cap L_2 \\ \text{regular} \end{array}$$

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

	$L_1 \, , \; L_2$	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
$\Rightarrow$	$L_1 \cap L_2$	regular

Example

$$\left.\begin{array}{ll} L_1 = \{a^n b\} & \text{regular} \\ L_2 = \{ab, ba\} & \text{regular} \end{array}\right\} \Rightarrow L_1 \cap L_2 = \{ab\}$$

regular

Another Proof for Intersection Closure

Machine M_1

DFA for L_1

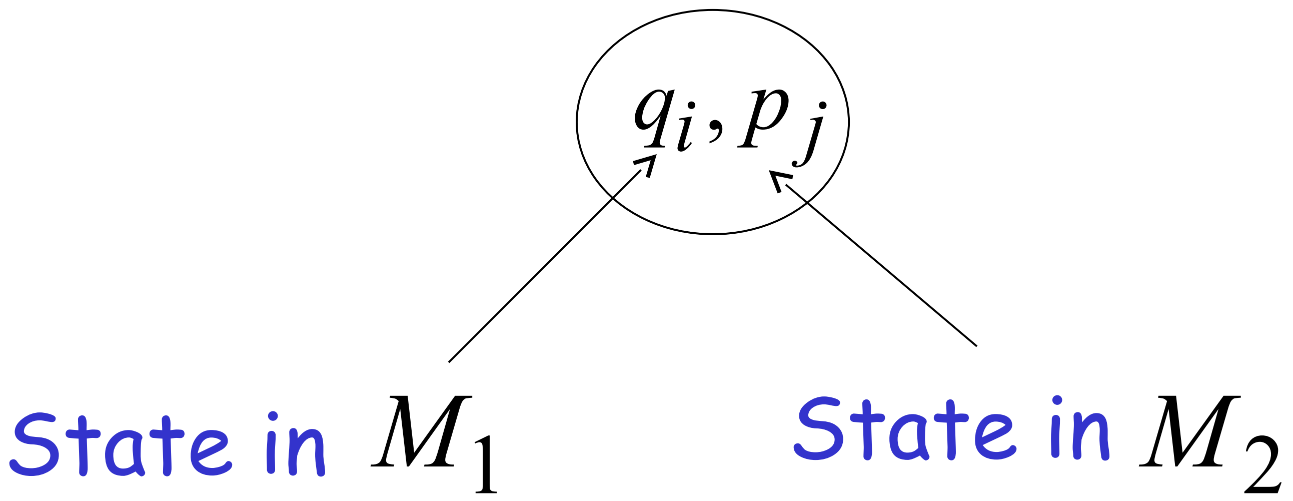
Machine M_2

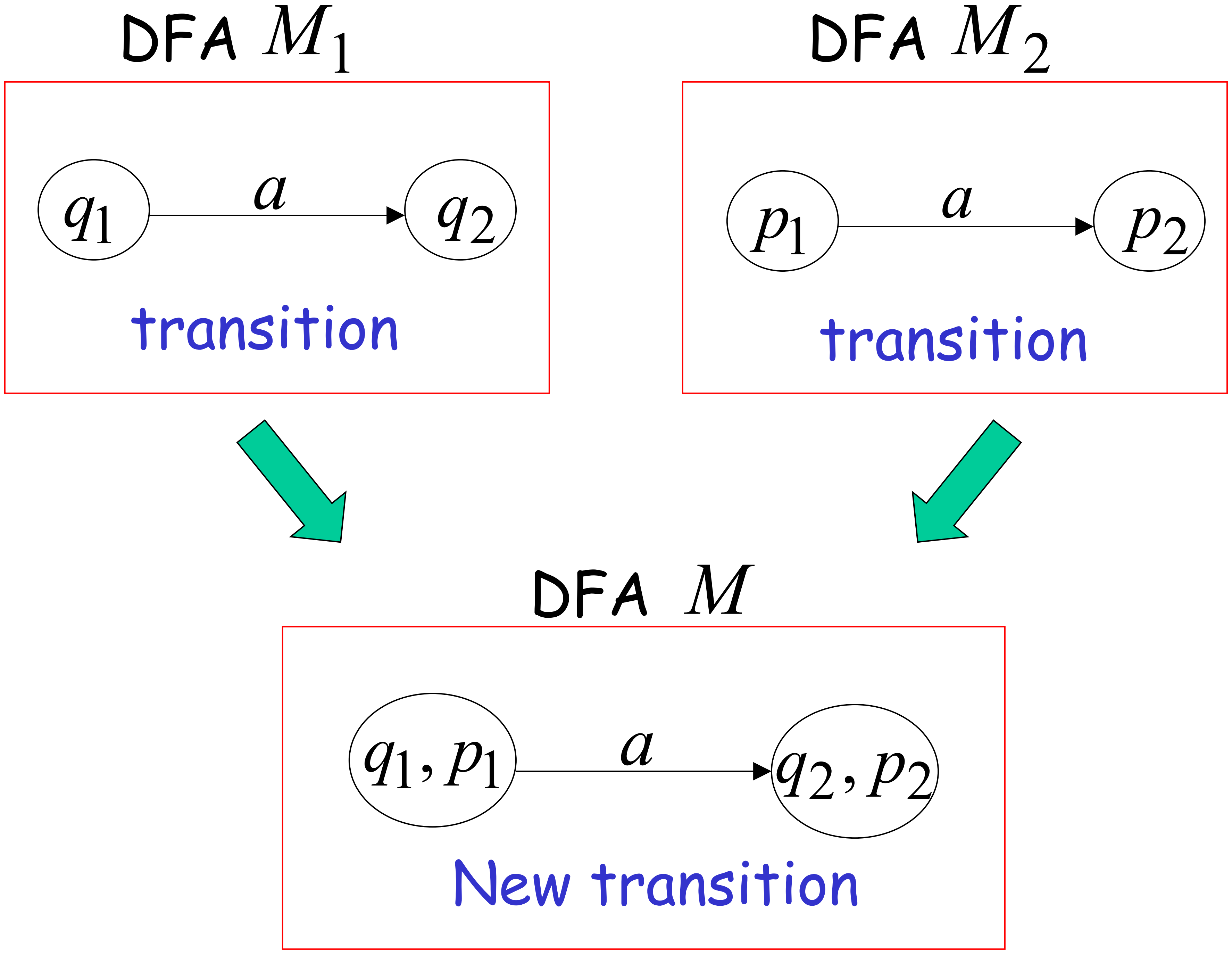
DFA for L_2

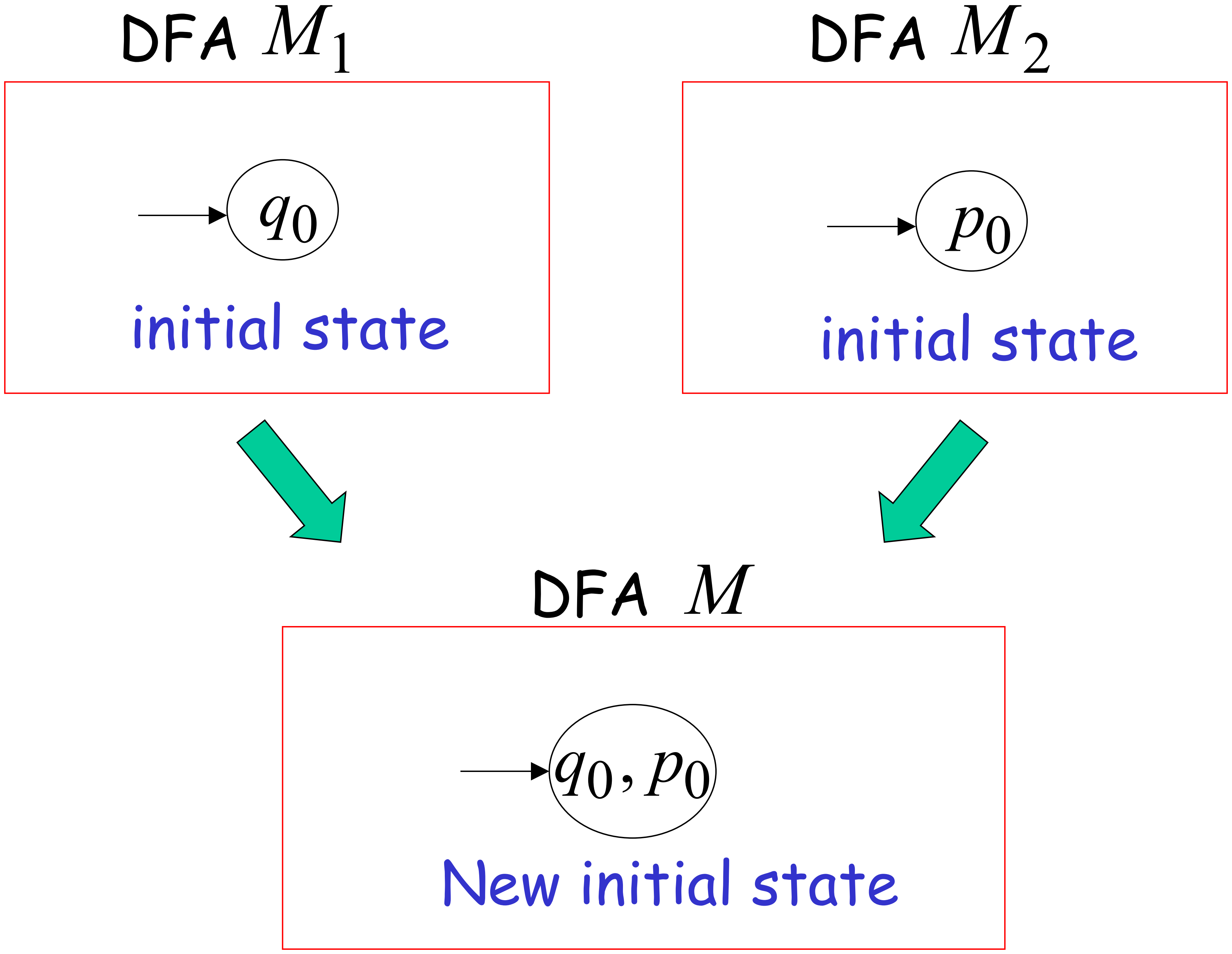
Construct a new DFA M that accepts $L_1 \cap L_2$

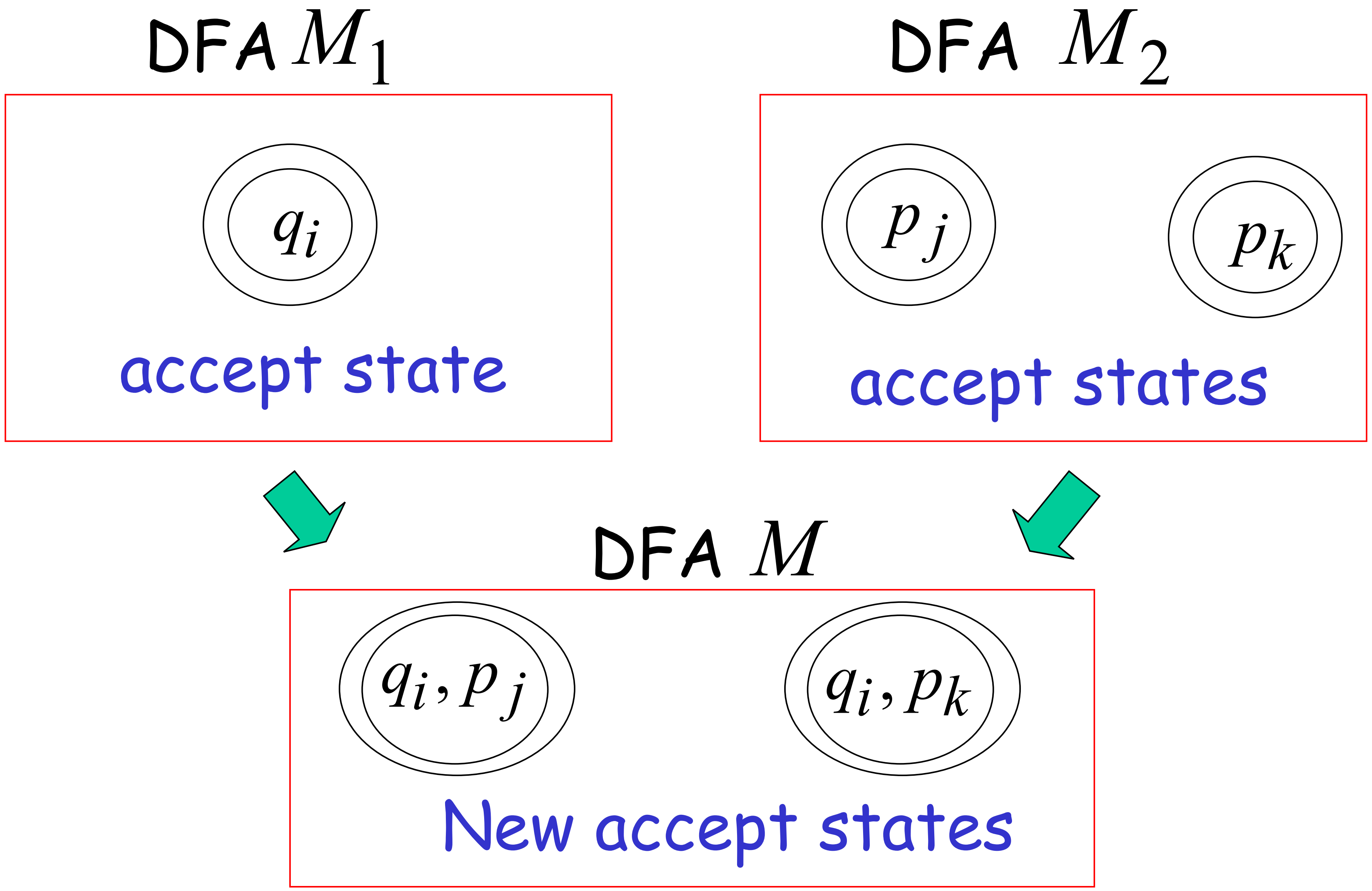
M simulates in parallel M_1 and M_2

States in M




DFA M_1
q_1
a
q_2
transition
DFA M_2
p_1
a
p_2
transition
DFA M
q_1, p_1
a
q_2, p_2
New transition


DFA M_1
q_0
initial state
DFA M_2
p_0
initial state
DFA M
q_0, p_0
New initial state



Both constituents must be accepting states

Example:

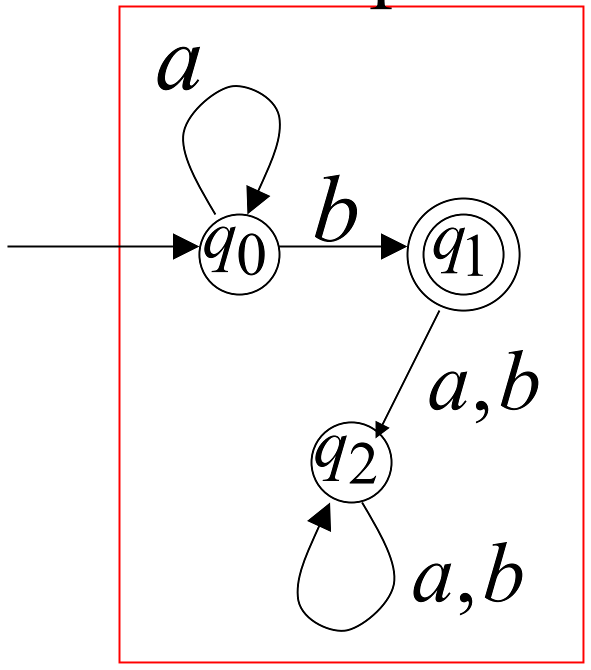
$n \quad 0$

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

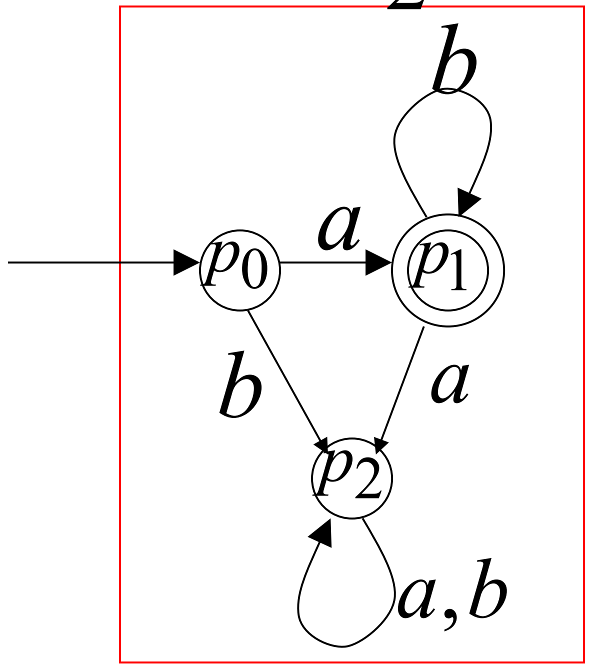
$m \quad 0$

$$L_2 = \{ab^m\}$$

M_1



M_2



Automaton for intersection

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

$$\begin{tikzcd}
{} \arrow[r] & q_0, p_0 \arrow[r, "a"] \arrow[d, "b"] & q_0, p_1 \arrow[r, "b"] \arrow[d, "a"] & q_1, p_1 \arrow[r, "a"] \arrow[d, "b"] & q_2, p_2 \arrow[loop above, "{a,b}"] \\
& q_1, p_2 \arrow[rrru, bend right, "{a,b}"'] & q_0, p_2 \arrow[l, "b"'] \arrow[loop below, "a"] & q_2, p_1 \arrow[ur, "a"] \arrow[loop below, "b"] &
\end{tikzcd}$$

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

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