# Properties of Regular Languages

1

For regular languages $L_1$ and $L_2$
we will prove that:

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

Costas Busch - LSU 2

2

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: $\overline{L_1}$

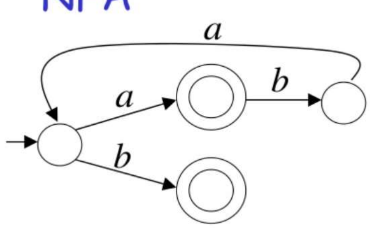
Intersection: $L_1 \cap L_2$





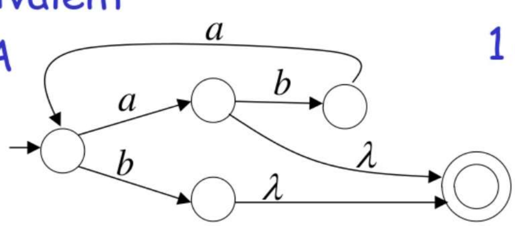
A useful transformation: use one accept state

NFA




2 accept states

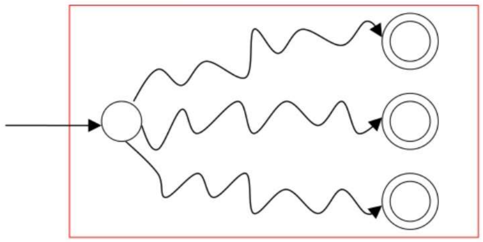
Equivalent NFA



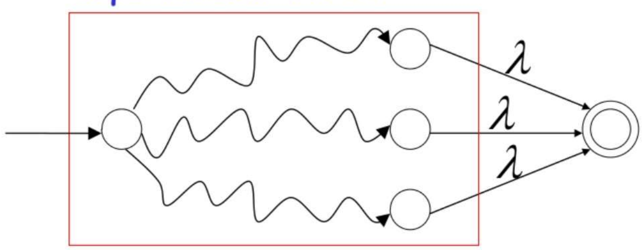

1 accept state

## In General

NFA



Equivalent NFA



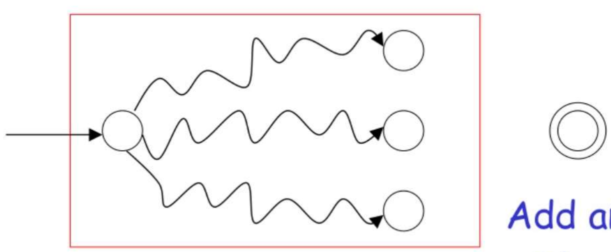

Single accepting state





## Extreme case

NFA without accepting state



Add an accepting state without transitions

## Take two languages

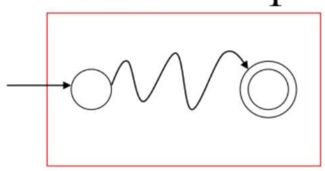
Regular language $L_1$

Regular language $L_2$

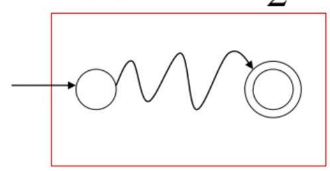
$L(M_1) = L_1$

$L(M_2) = L_2$

NFA $M_1$



NFA $M_2$



Single accepting state

Single accepting state



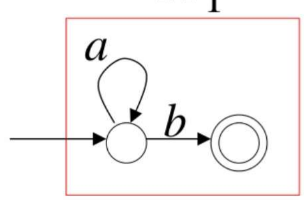
7

## Example

$n \geq 0$

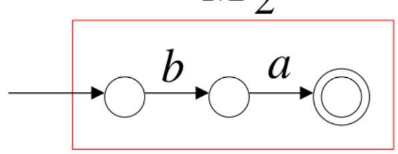
$L_1 = \{a^n b\}$

$M_1$




$L_2 = \{ba\}$

$M_2$

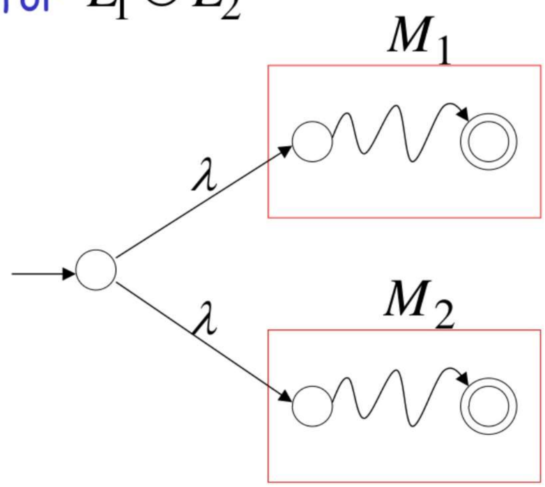





8

# Union

NFA for $L_1 \cup L_2$




$w \in L_1 \cup L_2 \iff w \in L_1$ or $w \in L_2$

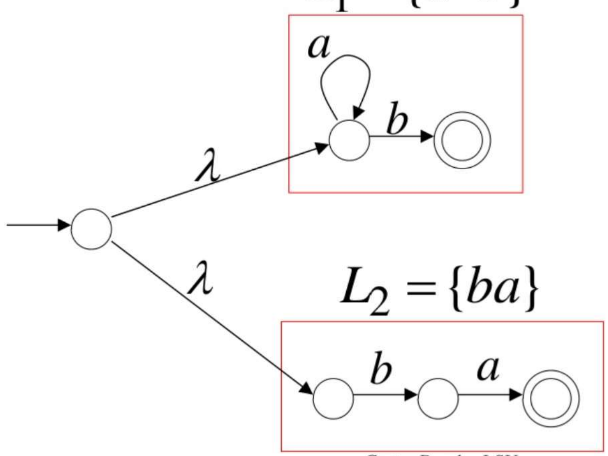


9

# Example

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

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

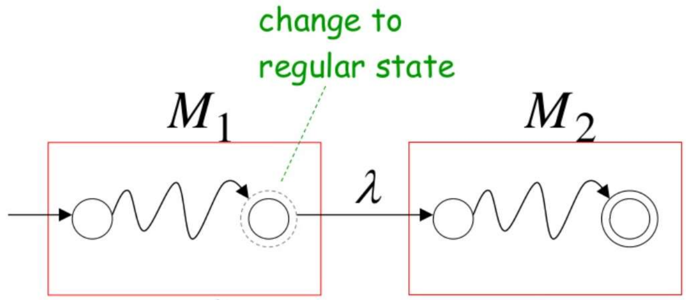





10

# Concatenation

NFA for $L_1L_2$



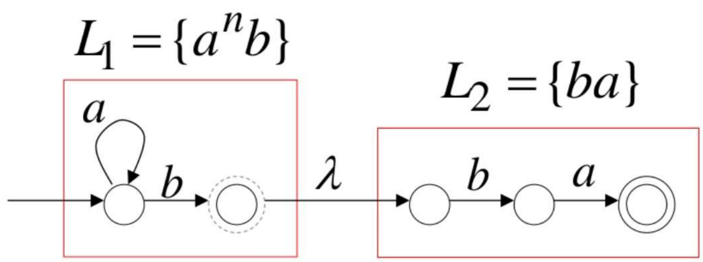

$w \in L_1L_2 \Leftrightarrow w = w_1w_2 : w_1 \in L_1$ and $w_2 \in L_2$

Costas Busch - LSU 11

11

# Example

NFA for $L_1L_2 = \{a^nb\}\{ba\} = \{a^nbba\}$

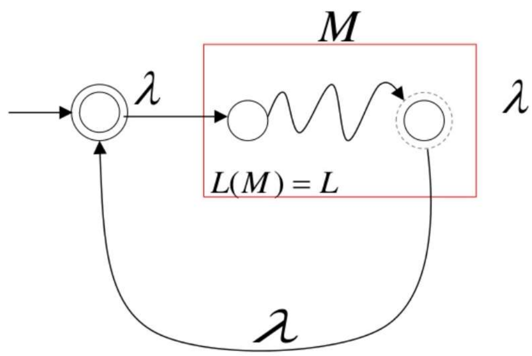



Costas Busch - LSU 12

12

## Star Operation

NFA for $L^*$




$w \in L^*$ $\Leftrightarrow$ $w = w_1 w_2 \Lambda\ w_k :\ w_i \in L$
or $w = \lambda$

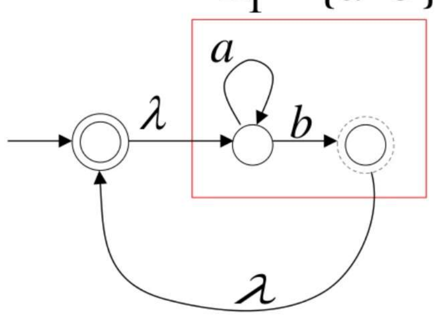
Costas Busch - LSU 13

13

## Example

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

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

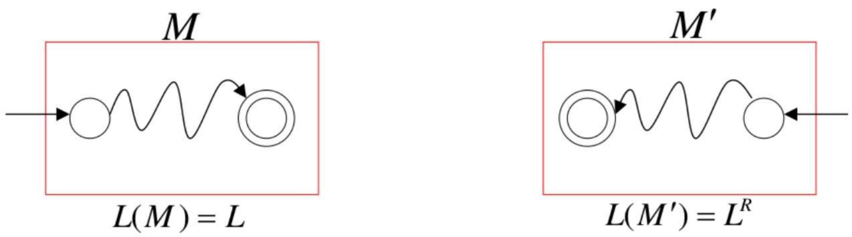



Costas Busch - LSU 14

14

## Reverse

NFA for $L^R$




1. Reverse all transitions

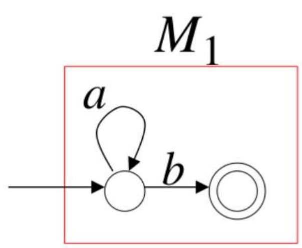
2. Make the initial state accept state
   and the accept state initial state



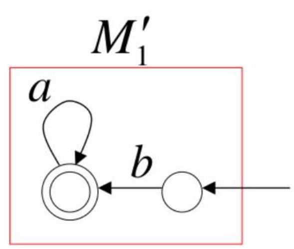


## Example

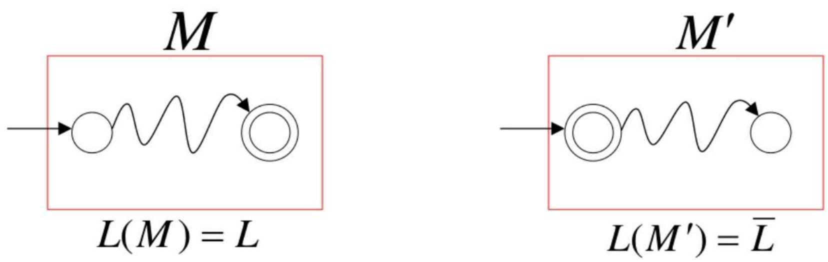
$L_1 = \{a^n b\}$




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



## Complement




1. Take the **DFA** that accepts $L$

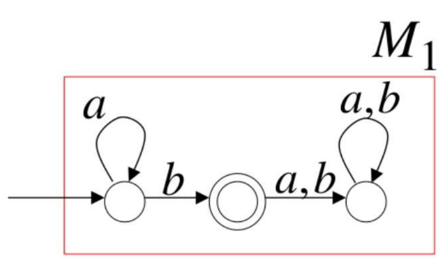
2. Make accept states regular
   and vice-versa



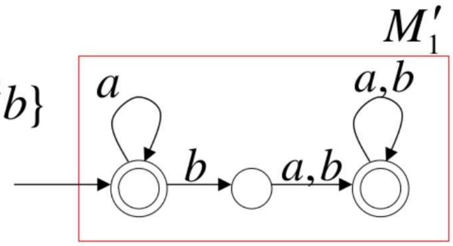
17

## Example

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




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



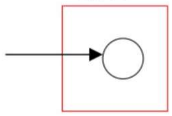



18

## NFAs cannot be used for complement

Make accept states regular
and vice-versa

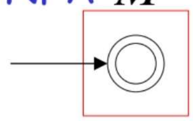
NFA $M$



$L(M) = \{\ \}$

$\overline{L(M)} = \Sigma^* = \{a,b\}^*$

NFA $M'$



$L(M') = \{\lambda\} \neq \overline{L(M)}$

it is not the
complement

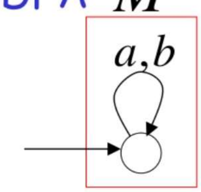
Costas Busch - LSU 19

19

## Same example with DFAs

Make accept states regular
and vice-versa

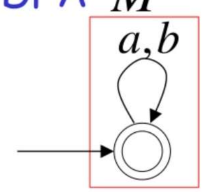
DFA $M$




$L(M) = \{\ \}$

$\overline{L(M)} = \Sigma^* = \{a,b\}^*$

DFA $M'$




$L(M') = \{a,b\}^* = \overline{L(M)}$

it is the
complement

Costas Busch - LSU 20

20

## Intersection

$L_1$ regular

$L_2$ regular

We show $L_1 \cap L_2$ regular





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

$L_1$ , $L_2$ regular, regular

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

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

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

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





## Another Proof for Intersection Closure

Machine $M_1$

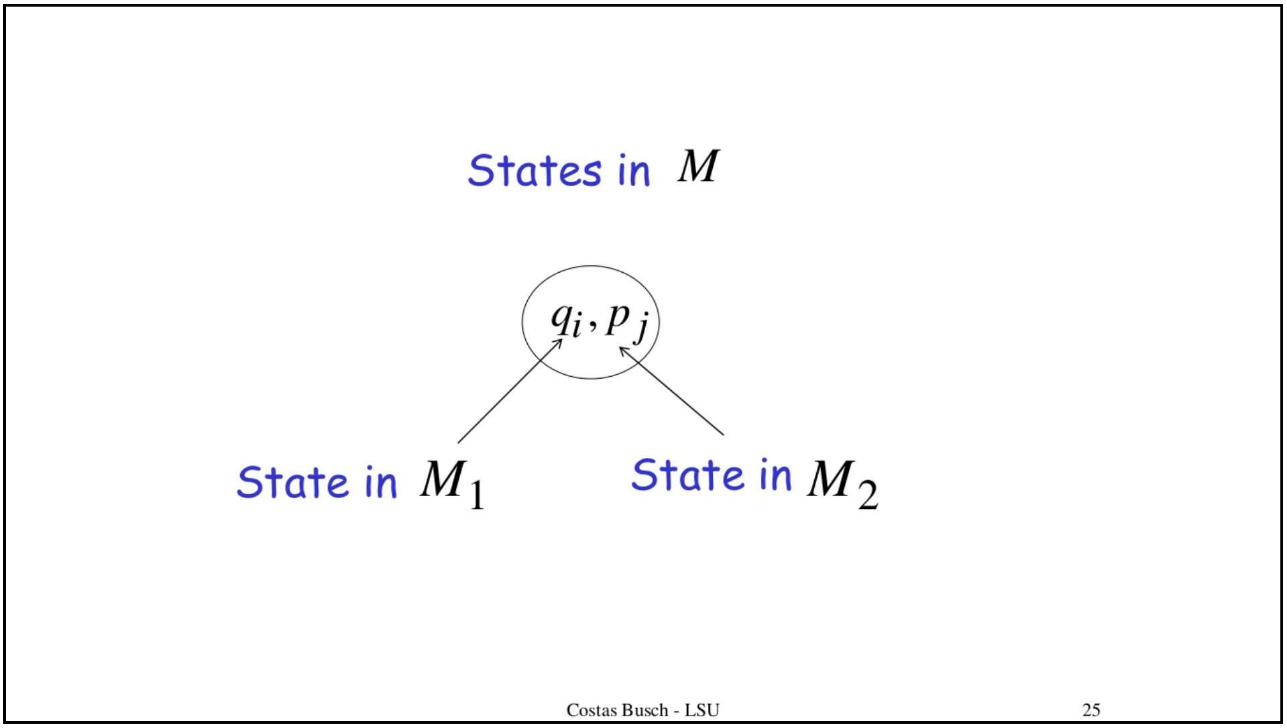
DFA for $L_1$

Machine $M_2$

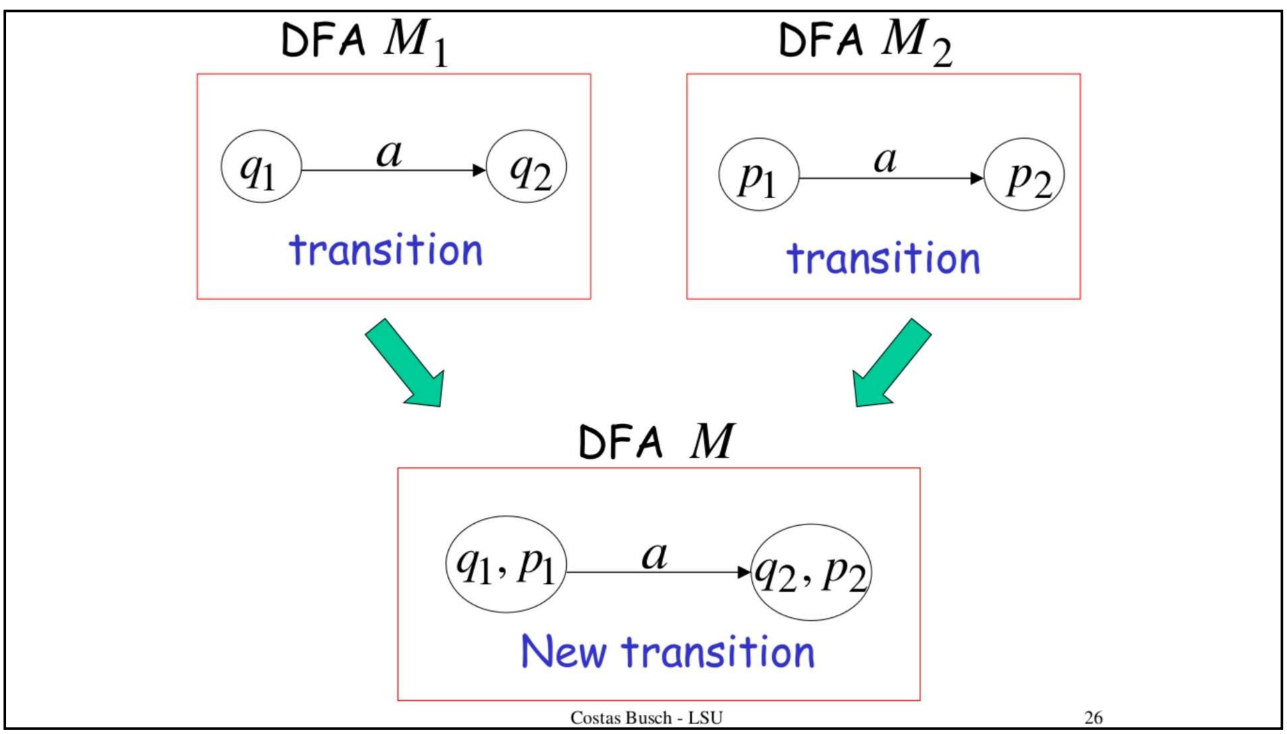
DFA for $L_2$

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

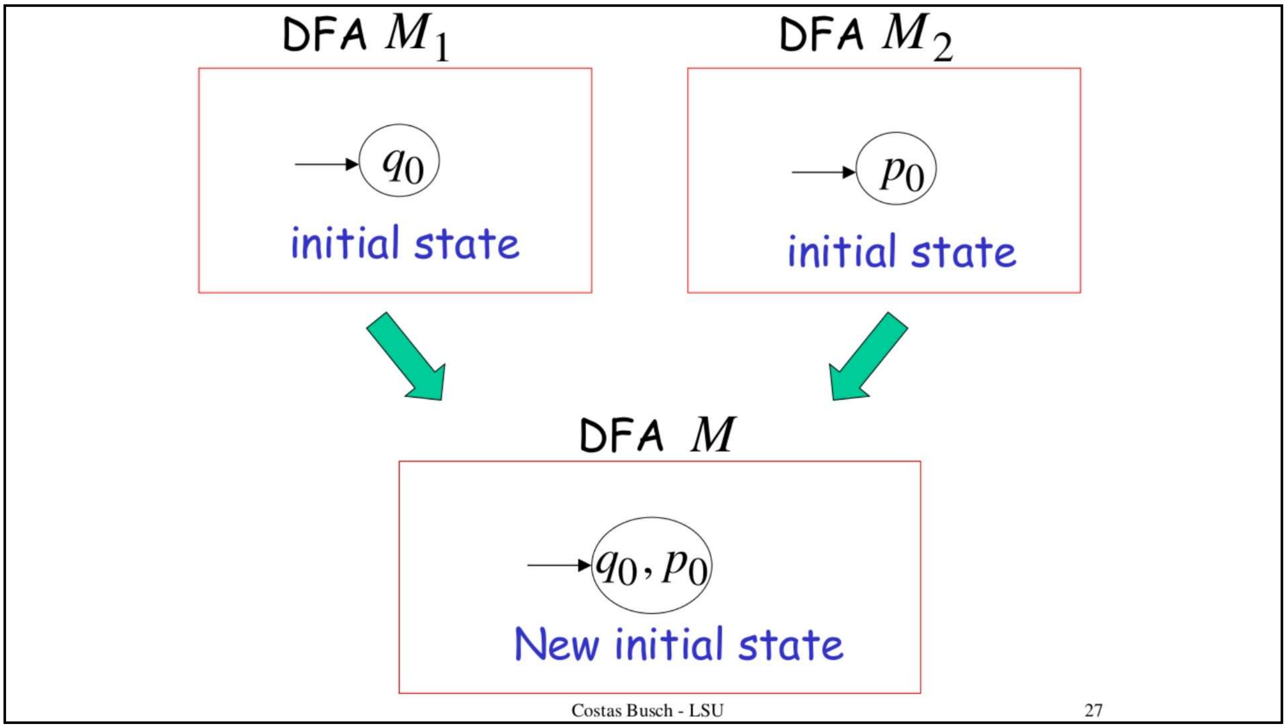
$M$ simulates in parallel $M_1$ and $M_2$


States in $M$
$q_i, p_j$
State in $M_1$
State in $M_2$
Costas Busch - LSU
25

25

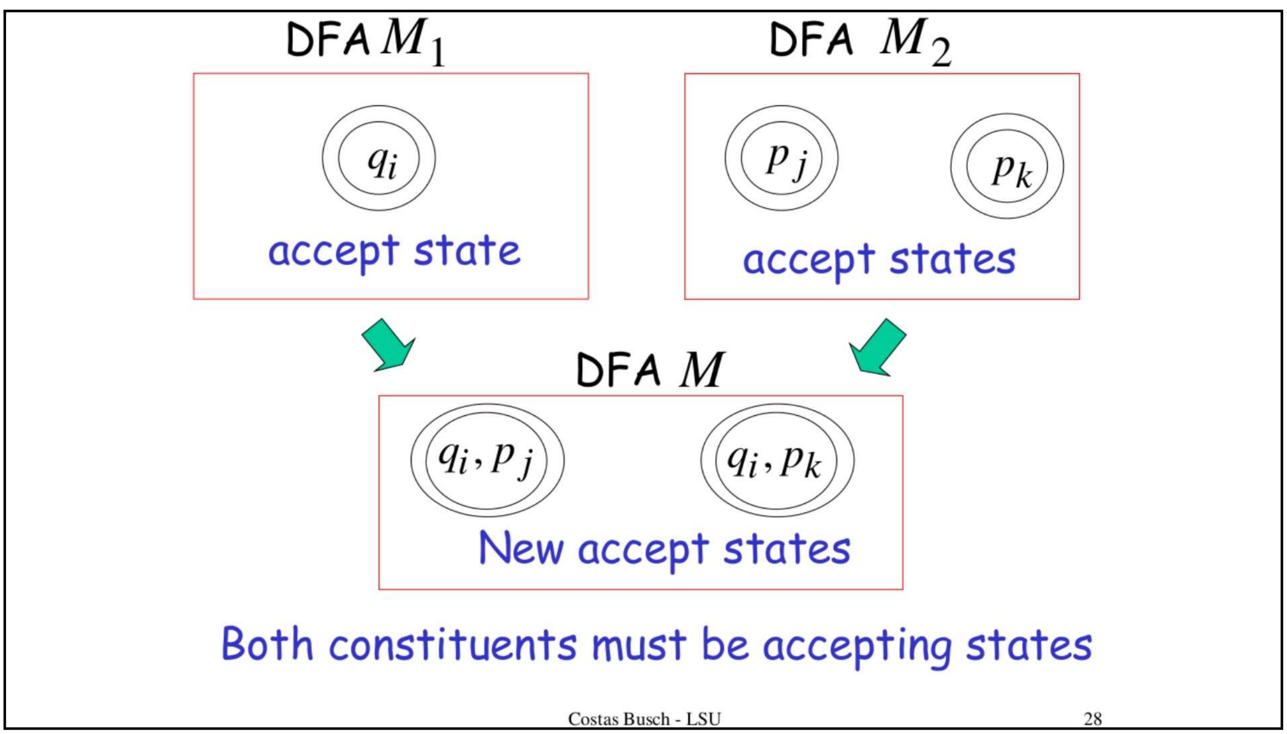

DFA $M_1$
$q_1$
$a$
$q_2$
transition
DFA $M_2$
$p_1$
$a$
$p_2$
transition
DFA $M$
$q_1, p_1$
$a$
$q_2, p_2$
New transition
Costas Busch - LSU
26

26



DFA $M_1$
$q_0$
initial state
DFA $M_2$
$p_0$
initial state
DFA $M$
$q_0, p_0$
New initial state
Costas Busch - LSU
27


27



DFA $M_1$
$q_i$
accept state
DFA $M_2$
$p_j$
$p_k$
accept states
DFA $M$
$q_i, p_j$
$q_i, p_k$
New accept states
Both constituents must be accepting states
Costas Busch - LSU
28

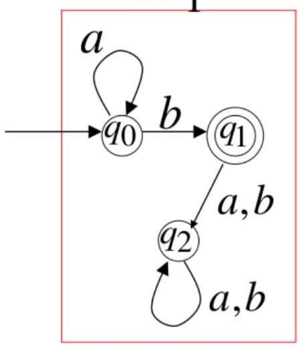

28

**Example:**

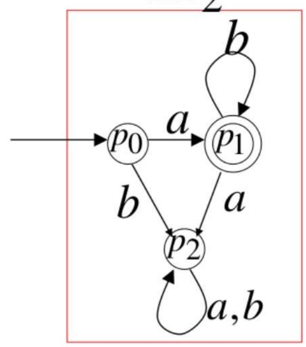
$$L_1 = \{a^n b\} \quad n \geq 0$$

$$L_2 = \{ab^m\} \quad m \geq 0$$

$M_1$




$M_2$

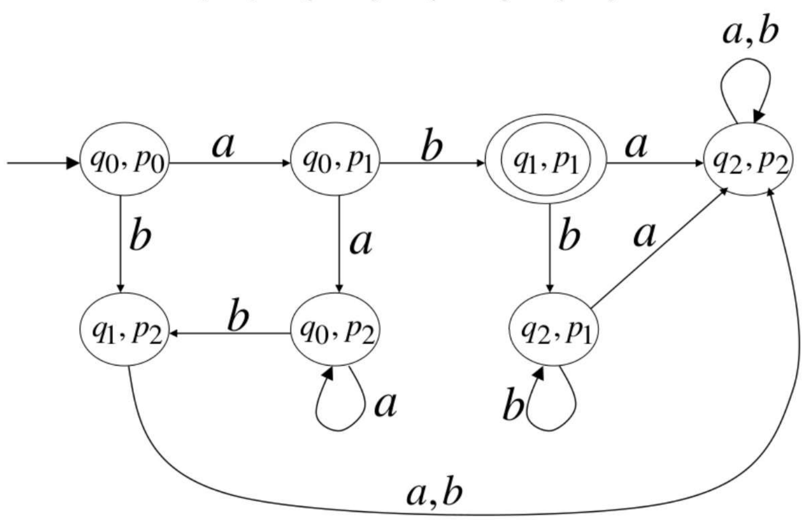







DFA $M$ for intersection

$$L(M) = \{a^n b\} \cap \{ab^m\} = \{ab\}$$



## Construction procedure for intersection

1. Build Initial State

2. For each new state and for each symbol
   add transition to either an existing state
   or create a new state and point to it

3. Repeat step 3 until no new states
   are added

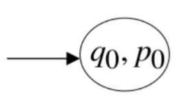
4. Designate accept states



31

## Automaton for intersection

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



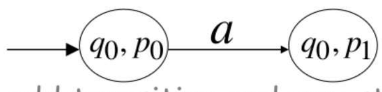

initial state



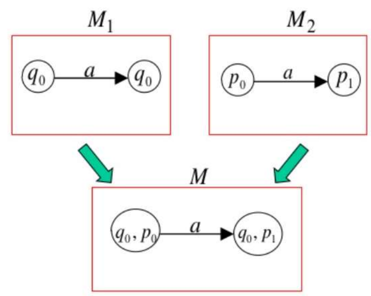
32

## Automaton for intersection

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




add transition and new state
for symbol a

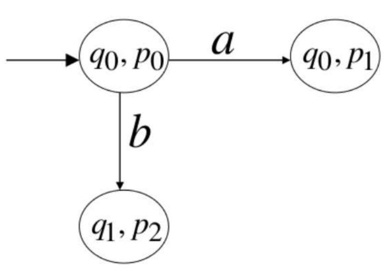




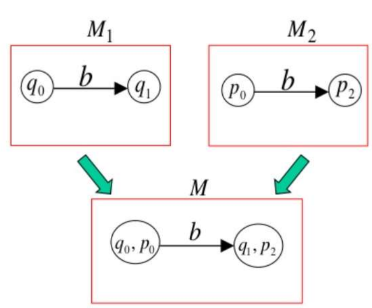



## Automaton for intersection

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



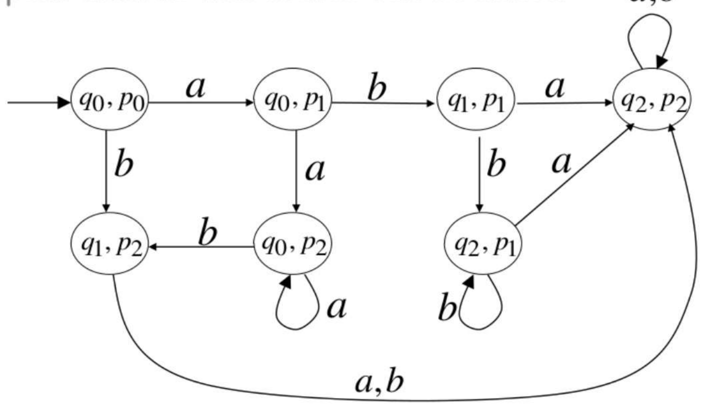

add transition and new state
for symbol b



## Automaton for intersection

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

Repeat until no new states can be added





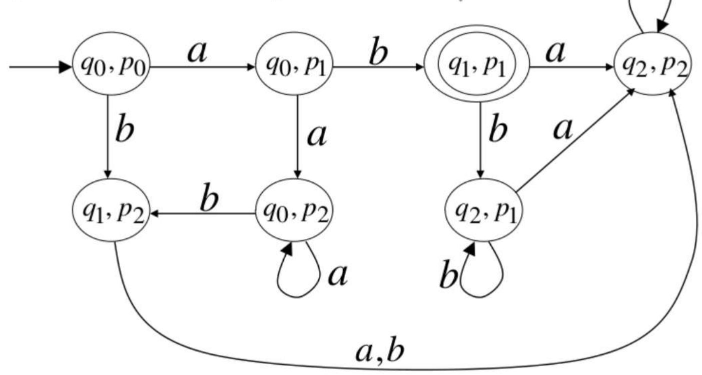

35

## Automaton for intersection

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

$q_1$ accept state for $M_1$

$p_1$ accept state for $M_2$ ⟹ add Accept state






36

Intersection DFA $M$:

simulates in parallel $M_1$ and $M_2$

accepts string $w$ if and only if:

$M_1$ accepts string $w$

**and** $M_2$ accepts string $w$

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