

Lecture 2

2 REGULAR LANGUAGES

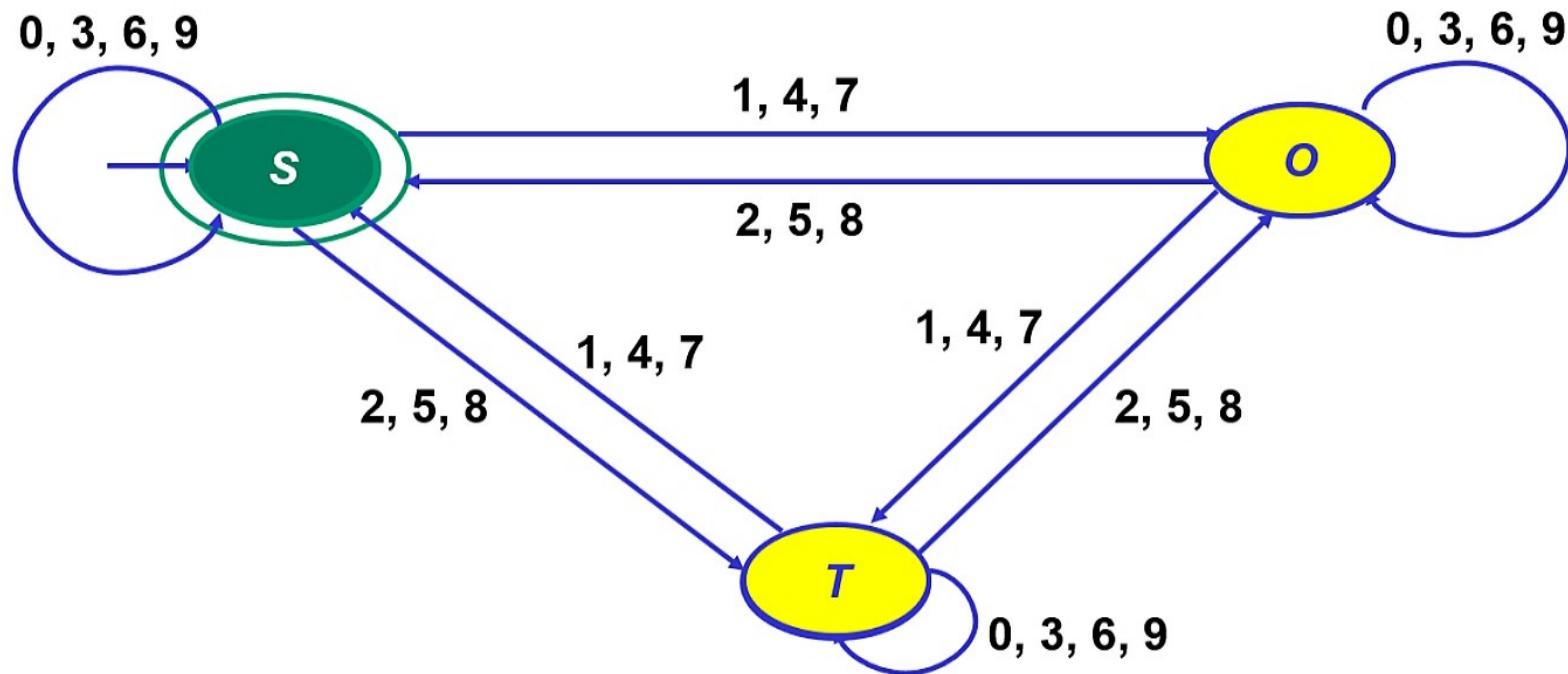
2.1 FINITE AUTOMATA

2.1.1 Deterministic finite automaton (DFA)

2.1.2 Minimization of deterministic finite automaton

Deterministic Finite Automaton (DFA)

The number is divisible by 3



Remainder is 1

Remainder is 2

Lecture 2

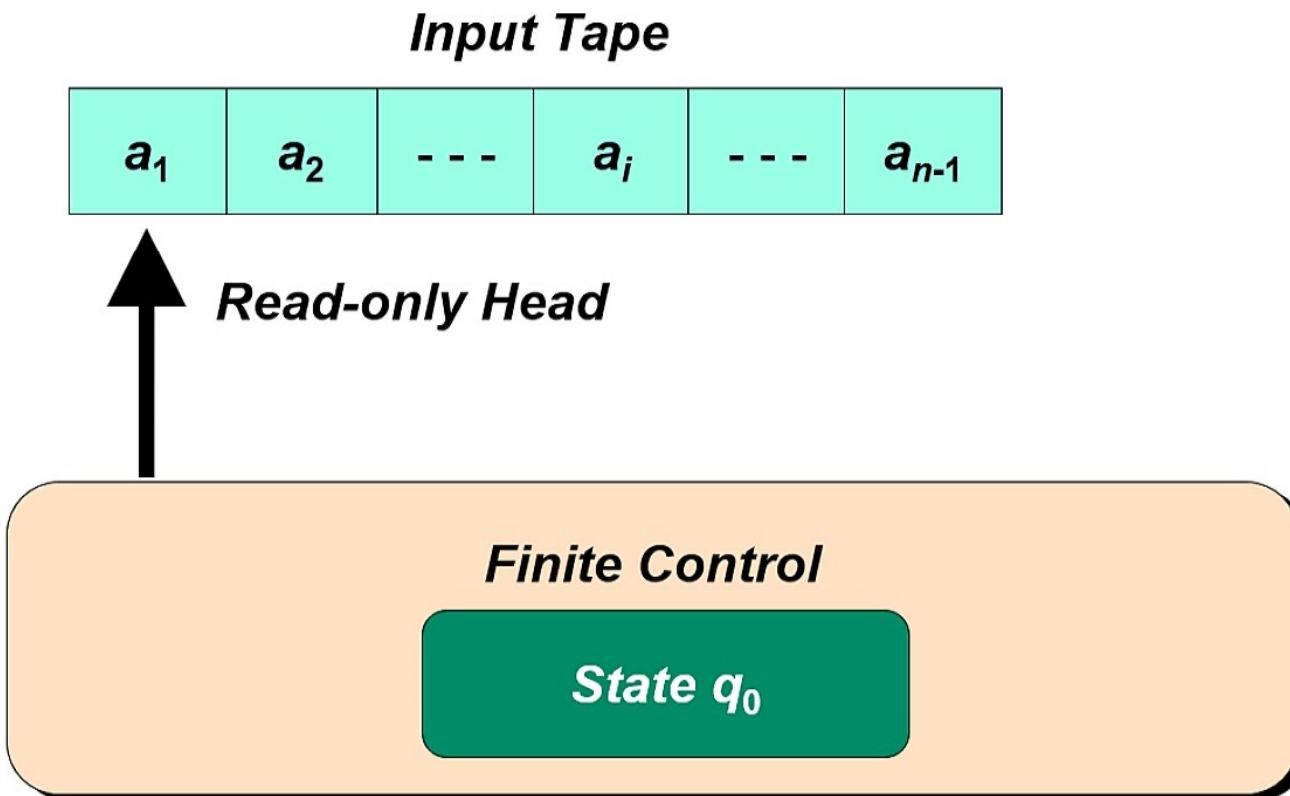
2 REGULAR LANGUAGES

2.1 FINITE AUTOMATA

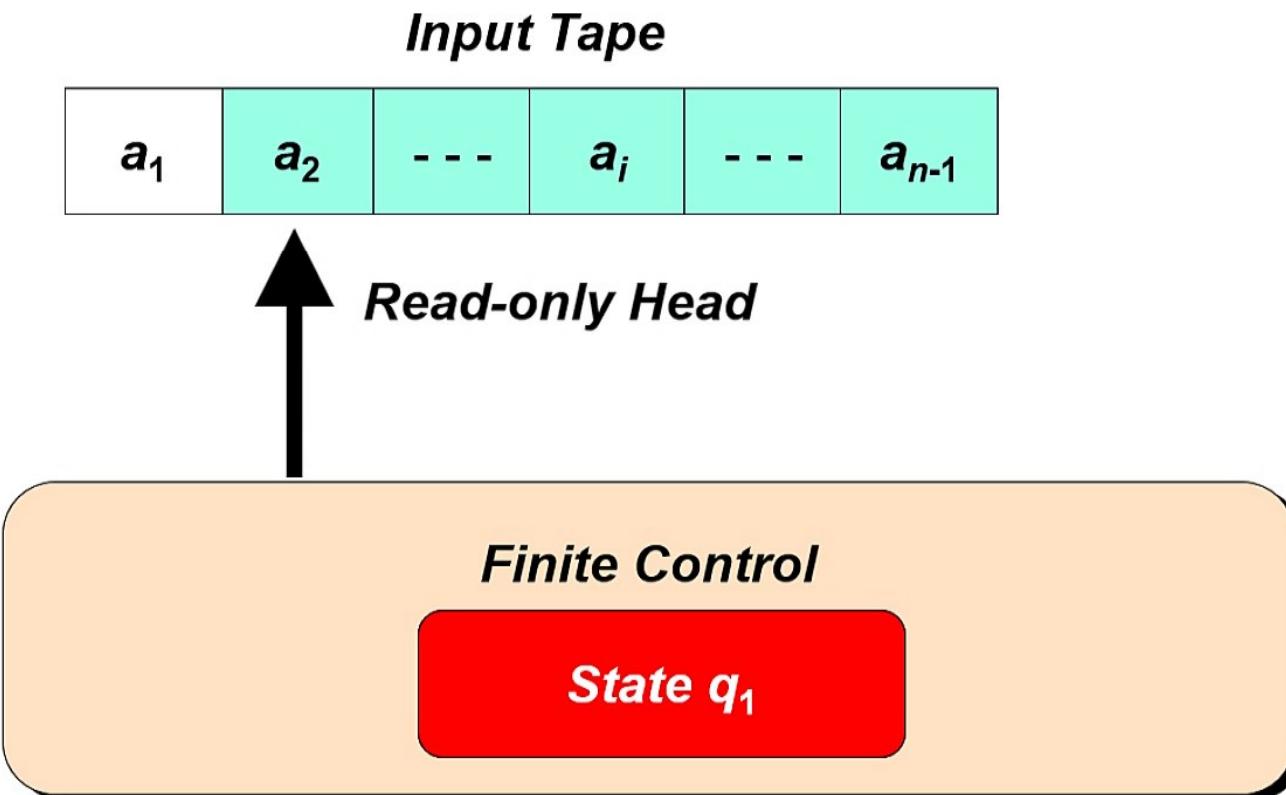
2.1.1 Deterministic finite automaton (DFA)

2.1.2 Minimization of deterministic finite automaton

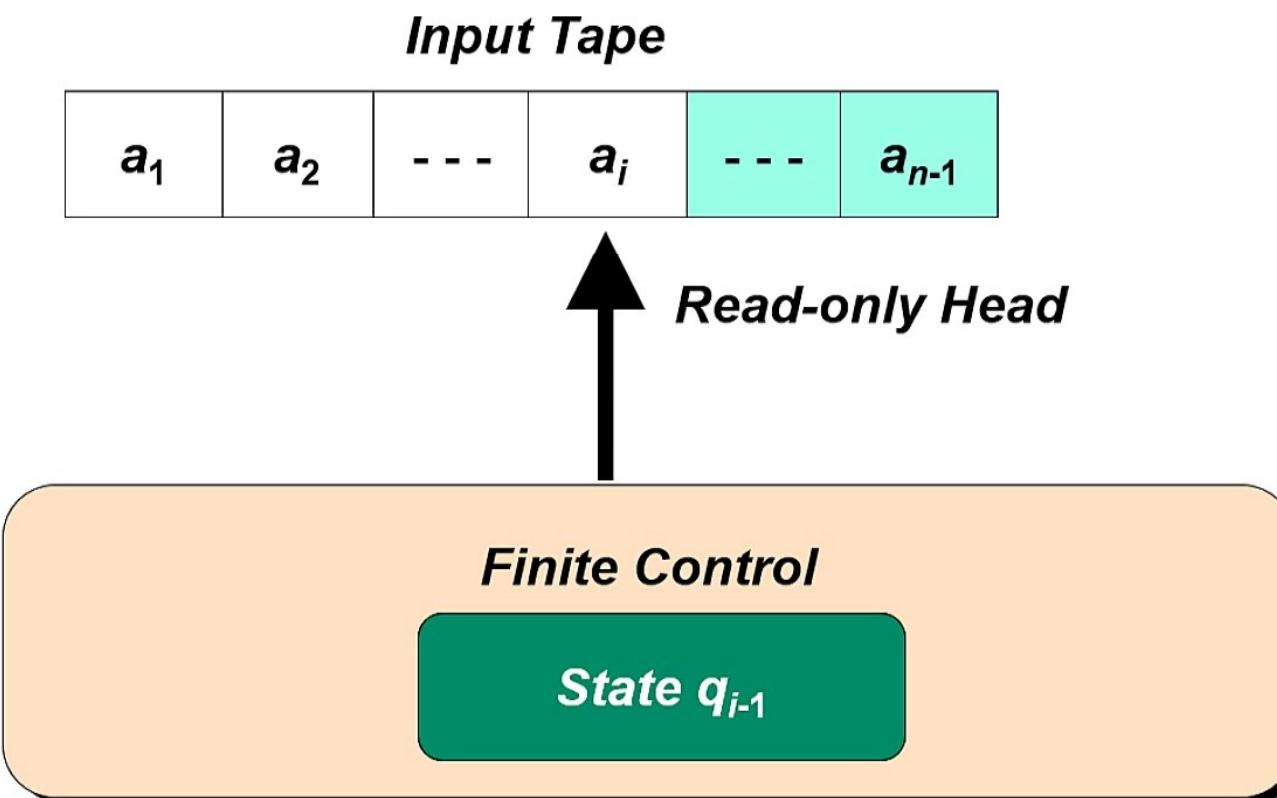
Deterministic Finite Automaton (DFA)



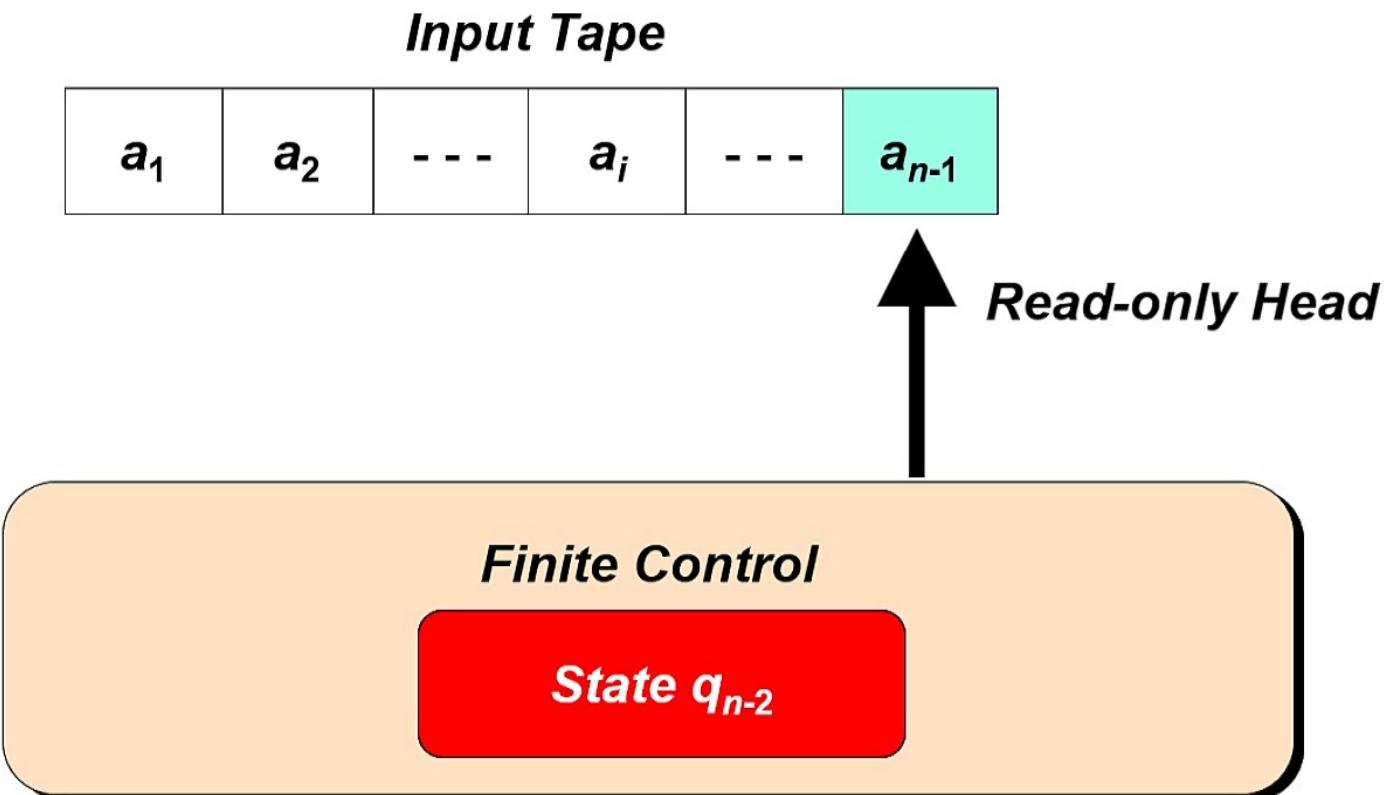
Deterministic Finite Automaton (DFA)



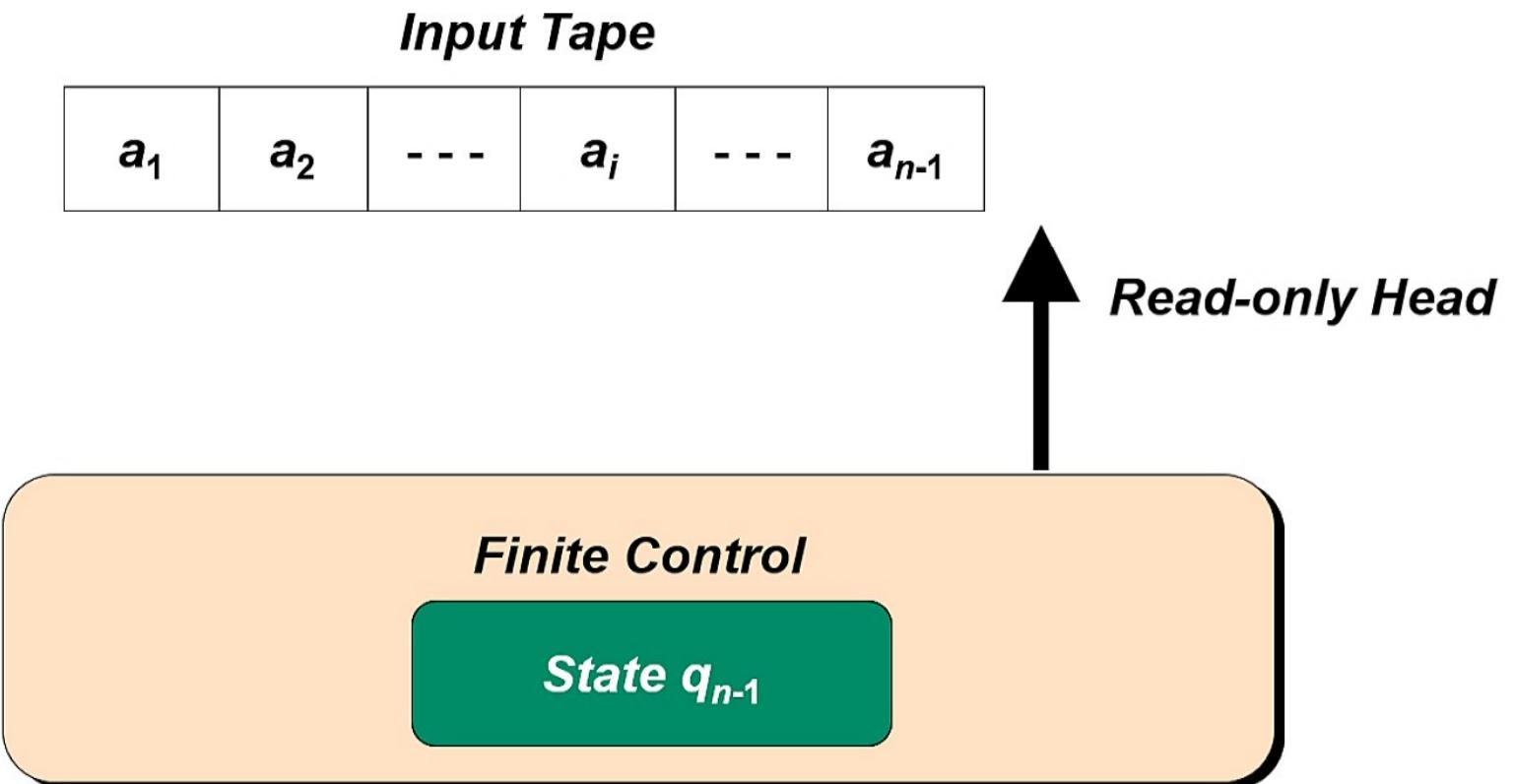
Deterministic Finite Automaton (DFA)



Deterministic Finite Automaton (DFA)



Deterministic Finite Automaton (DFA)



Deterministic Finite Automaton (DFA)

$$dfa = (Q, \Sigma, \delta, q_0, F)$$

Q

- a finite set of states

Σ

- a finite set of input symbols

δ

- a transition function $Q \times \Sigma \rightarrow Q$

$q_0 \in Q$

- an initial (start) state

$F \subseteq Q$

- a set of accepting states

$$\text{NextState} = \delta(\text{CurrentState}, \text{InputSymbol})$$

Deterministic Finite Automaton (DFA)

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

$$(1) \quad \hat{\delta}(q, \varepsilon) = q$$

$$(2) \quad \hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$$

$$\hat{\delta}(q, a) = \delta(\hat{\delta}(q, \varepsilon), a) = \delta(q, a)$$

String and Language Acceptance (DFA)

$$DFA = (Q, \Sigma, \delta, q_0, F)$$

DFA accepts the string w if $\delta(q_0, w)=p$ and $p \in F$

DFA accepts the language $L(DFA) = \{w \mid \delta(q_0, w) \in F\} \subseteq \Sigma^*$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo} \quad \delta(\text{eo}, 1) = \text{ee} \quad \delta(\text{oe}, 1) = \text{oo} \quad \delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{ee}, 0) = \text{oe} \quad \delta(\text{eo}, 0) = \text{oo} \quad \delta(\text{oe}, 0) = \text{ee} \quad \delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\text{ee}, \varepsilon), 1) =$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) =$$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) =$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\text{ee}, 1101)$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\delta(\text{ee}, 110), 1)$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\delta(\delta(\text{ee}, 11), 0), 1)$$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\delta(\delta(\delta(\text{ee}, 1), 1), 0), 1)$$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\delta(\delta(\text{eo}, 1), 0), 1)$$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\delta(\text{ee}, 0), 1)$$

Example of a DFA

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

$$\delta(\text{oe}, 1)$$

Example of a DFA

$$\begin{array}{llll} \delta(\text{ee}, 1) = \text{eo} & \delta(\text{eo}, 1) = \text{ee} & \delta(\text{oe}, 1) = \text{oo} & \delta(\text{oo}, 1) = \text{oe} \\ \delta(\text{ee}, 0) = \text{oe} & \delta(\text{eo}, 0) = \text{oo} & \delta(\text{oe}, 0) = \text{ee} & \delta(\text{oo}, 0) = \text{eo} \end{array}$$

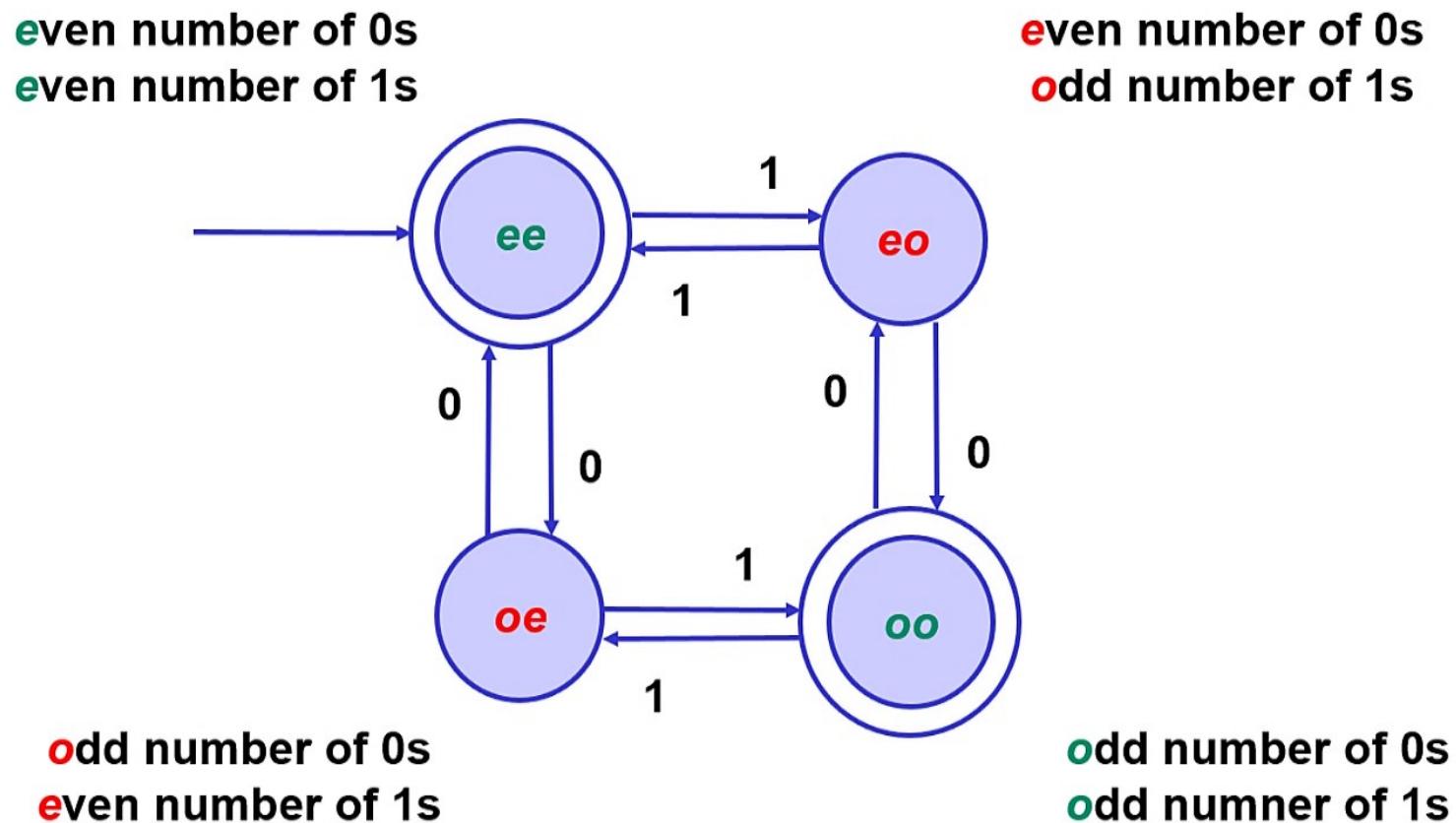
$$\delta(\text{ee}, \varepsilon) = \text{ee}$$

$$\delta(\text{ee}, 1) = \delta(\delta(\text{ee}, \varepsilon), 1) = \delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{ee}, 1101) =$$

oo

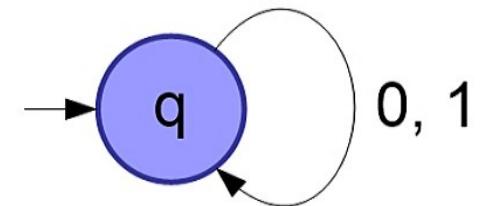
Example of a DFA



Example of a DFA

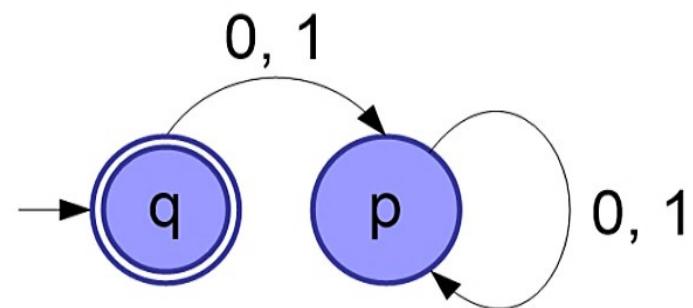
$M_1 = (\{q\}, \{0, 1\}, \{\delta(q, 0) = q, \delta(q, 1) = q\}, q, \{ \ })$

$L(M_1) = \{ \ }$



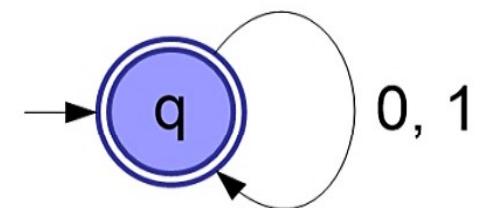
$M_2 = (\{q, p\}, \{0, 1\}, \{\delta(q, 0) = p, \delta(q, 1) = p, \delta(p, 0) = p, \delta(p, 1) = p\}, q, \{ q \})$

$L(M_2) = \{ \varepsilon \}$

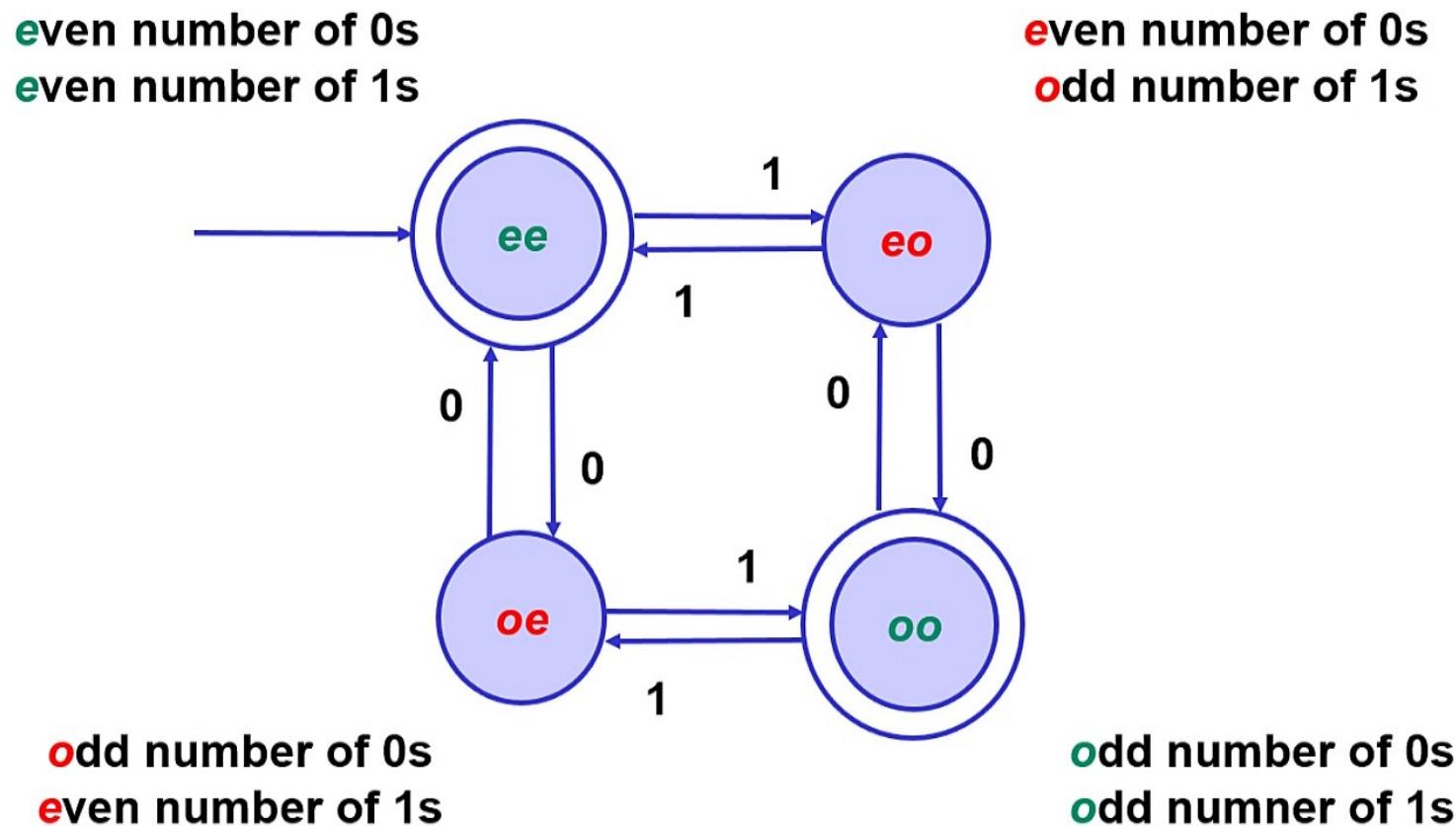


$M_3 = (\{q\}, \{0, 1\}, \{\delta(q, 0) = q, \delta(q, 1) = q\}, q, \{ q \})$

$L(M_3) = \Sigma^*$



Software Implementation of DFA



State Representation Method – EXPLICIT

$\delta(\text{ee}, 0) = \text{oe}$
 $\delta(\text{ee}, 1) = \text{eo}$
 $\delta(\text{eo}, 0) = \text{oo}$
 $\delta(\text{eo}, 1) = \text{ee}$
 $\delta(\text{oe}, 0) = \text{ee}$
 $\delta(\text{oe}, 1) = \text{oo}$
 $\delta(\text{oo}, 0) = \text{eo}$
 $\delta(\text{oo}, 1) = \text{oe}$

Table [EE, 0] = OE;
Table [EE, 1] = EO;
Table [EE, ⊥] = 1;
Table [OE, 0] = EE;
Table [OE, 1] = OO;
Table [OE, ⊥] = 0;
Table [EO, 0] = OO;
Table [EO, 1] = EE;
Table [EO, ⊥] = 0;
Table [OO, 0] = EO;
Table [OO, 1] = OE;
Table [OO, ⊥] = 1;

State = *EE*;
Read(Symbol);

while (*Symbol* != \perp)
{
 State = *Table* [**State**, *Symbol*];
 Read(Symbol);
}

Print(Table [State, ⊥], State);

State Representation Method – IMPLICIT

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

EE: *Read(Symbol);*

```
if (Symbol == ⊥)
    Print("STRING IS ACCEPTABLE, EVEN:0, EVEN:1");
if (Symbol == 0)
    goto OE;
else
    goto EO;
```

OE: *Read(Symbol);*

```
if (Symbol == ⊥)
    Print("STRING IS NOT ACCEPTABLE, ODD:0, EVEN:1");
if (Symbol == 0)
    goto EE;
else
    goto OO;
```

State Representation Method – IMPLICIT

$$\delta(\text{ee}, 1) = \text{eo}$$

$$\delta(\text{eo}, 1) = \text{ee}$$

$$\delta(\text{oe}, 1) = \text{oo}$$

$$\delta(\text{oo}, 1) = \text{oe}$$

$$\delta(\text{ee}, 0) = \text{oe}$$

$$\delta(\text{eo}, 0) = \text{oo}$$

$$\delta(\text{oe}, 0) = \text{ee}$$

$$\delta(\text{oo}, 0) = \text{eo}$$

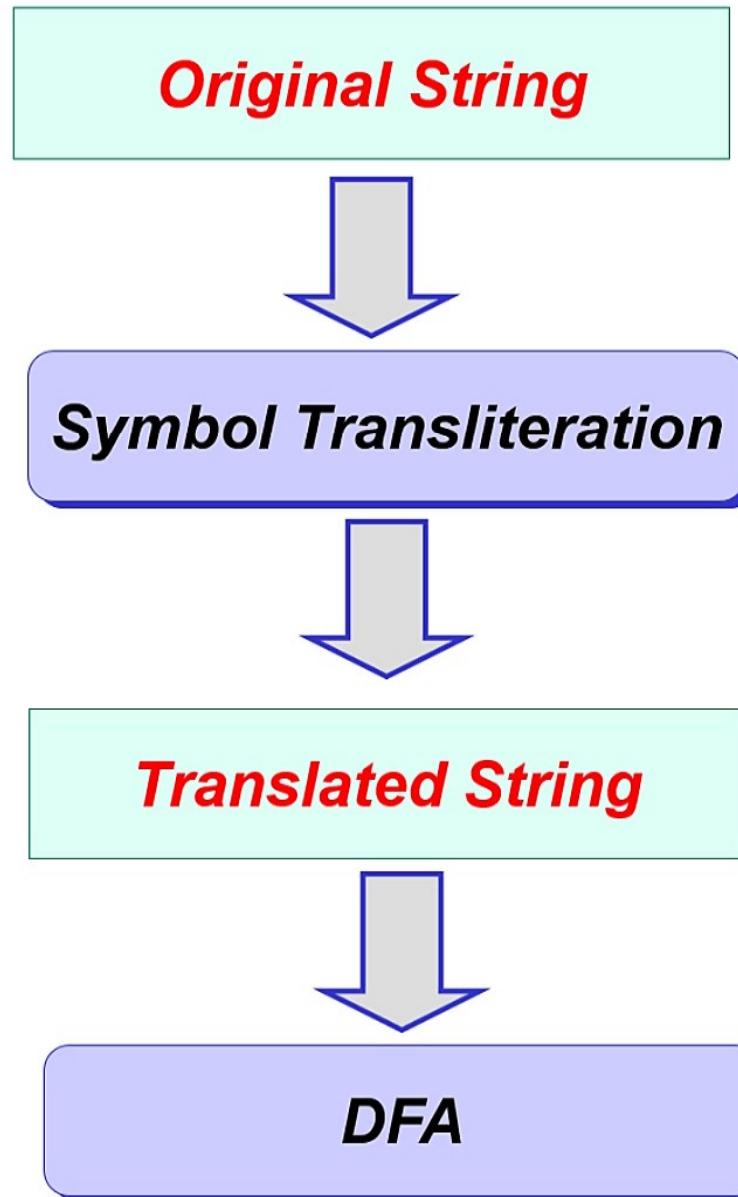
EO: *Read(Symbol);*

```
if (Symbol == ⊥)
    Print("STRING IS NOT ACCEPTABLE, EVEN:0, ODD:1");
if (Symbol == 0)
    goto OO;
else
    goto EE;
```

OO: *Read(Symbol);*

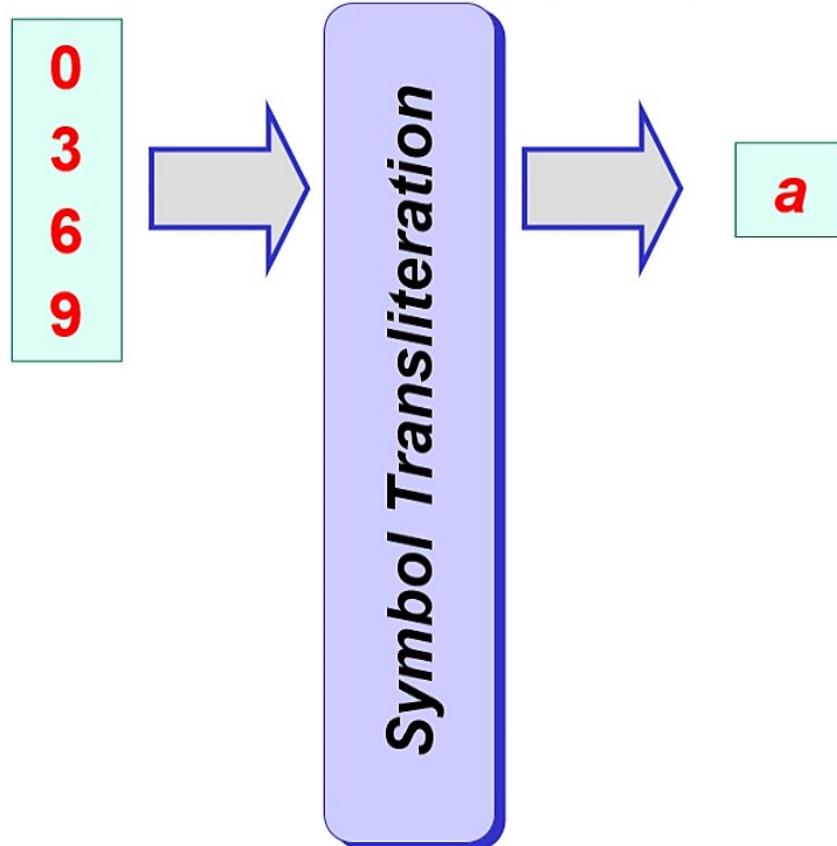
```
if (Symbol == ⊥)
    Print("STRING IS ACCEPTABLE, ODD:0, ODD:1");
if (Symbol == 0)
    goto EO;
else
    goto OE;
```

Symbol Transliteration



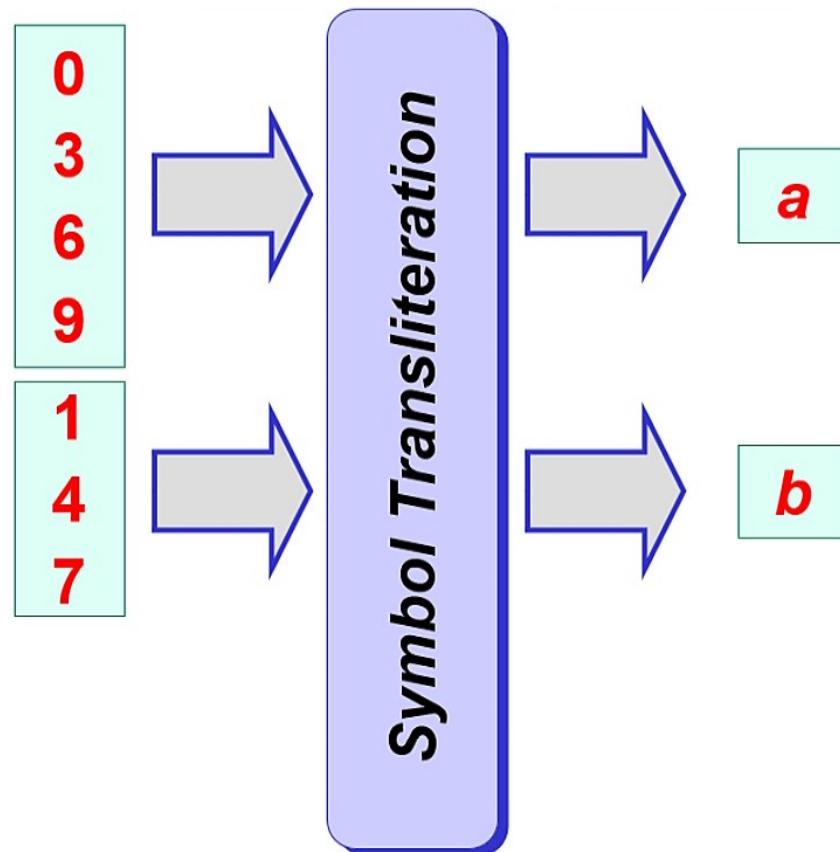
Example of Symbol Transliteration

	0	1	2	3	4	5	6	7	8	9	
S	S	O	T	S	O	T	S	O	T	S	1
O	O	T	S	O	T	S	O	T	S	O	0
T	T	S	O	T	S	O	T	S	O	T	0

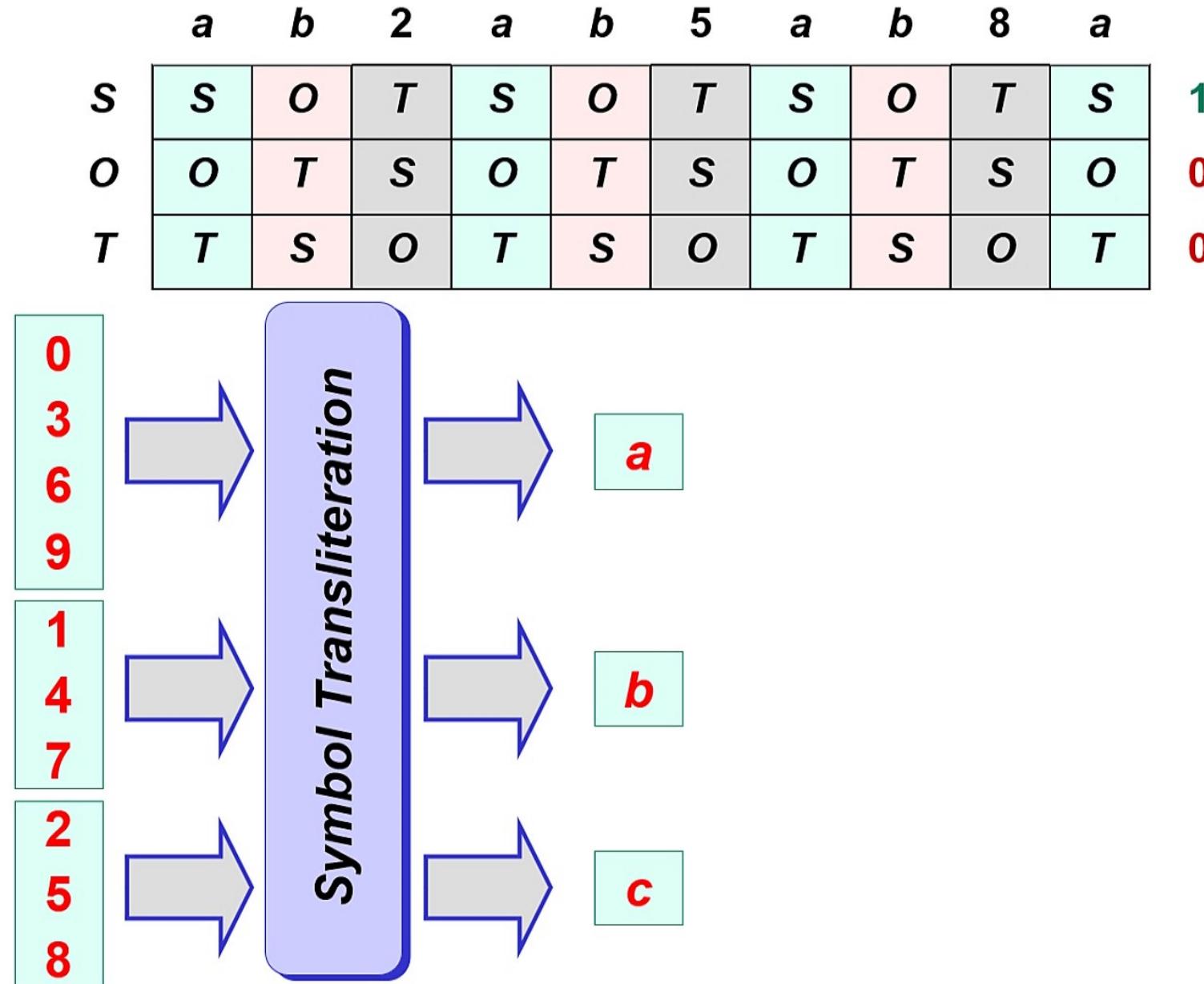


Example of Symbol Transliteration

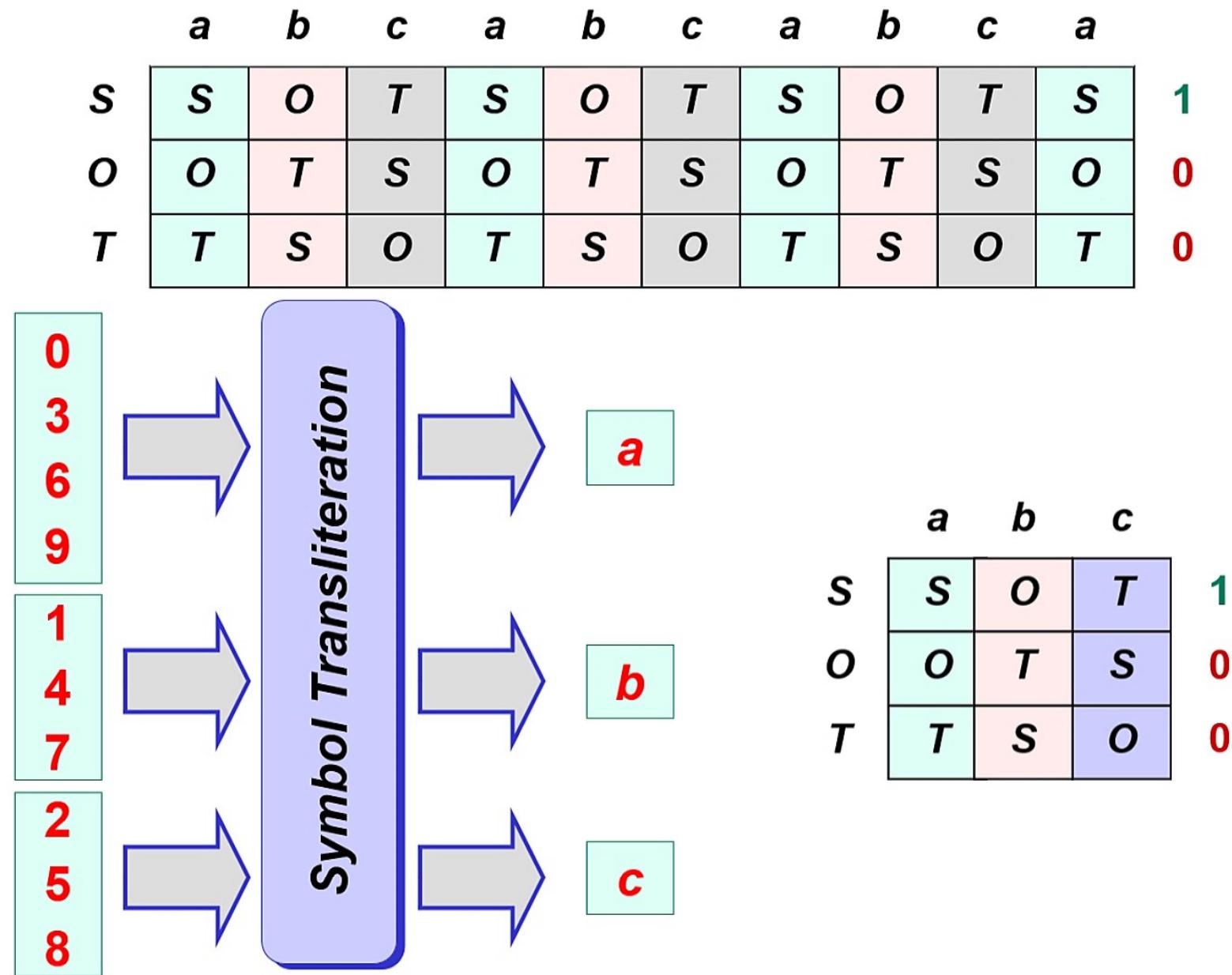
	a	1	2	a	4	5	a	7	8	a	
s	S	O	T	S	O	T	S	O	T	S	1
o	O	T	S	O	T	S	O	T	S	O	0
t	T	S	O	T	S	O	T	S	O	T	0



Example of Symbol Transliteration



Example of Symbol Transliteration



Lecture 2

2 REGULAR LANGUAGES

2.1 FINITE AUTOMATA

2.1.1 Deterministic finite automaton (DFA)

2.1.2 Minimization of deterministic finite automaton

Minimization of DFA

- Efficient software implementation
 - Requires DFA with a minimum possible number of states

For a regular language L there exists a DFA M with less or equal number of states than any other DFA M' that accepts that same language L .

- Equivalence of states

State p of a DFA $M=(Q, \Sigma, \delta, q_0, F)$ is equivalent to the state p' of a DFA $M'=(Q', \Sigma, \delta', q_0', F')$ if and only if DFA M from the state p accepts the same set of strings as DFA M' accepts from the state p' .

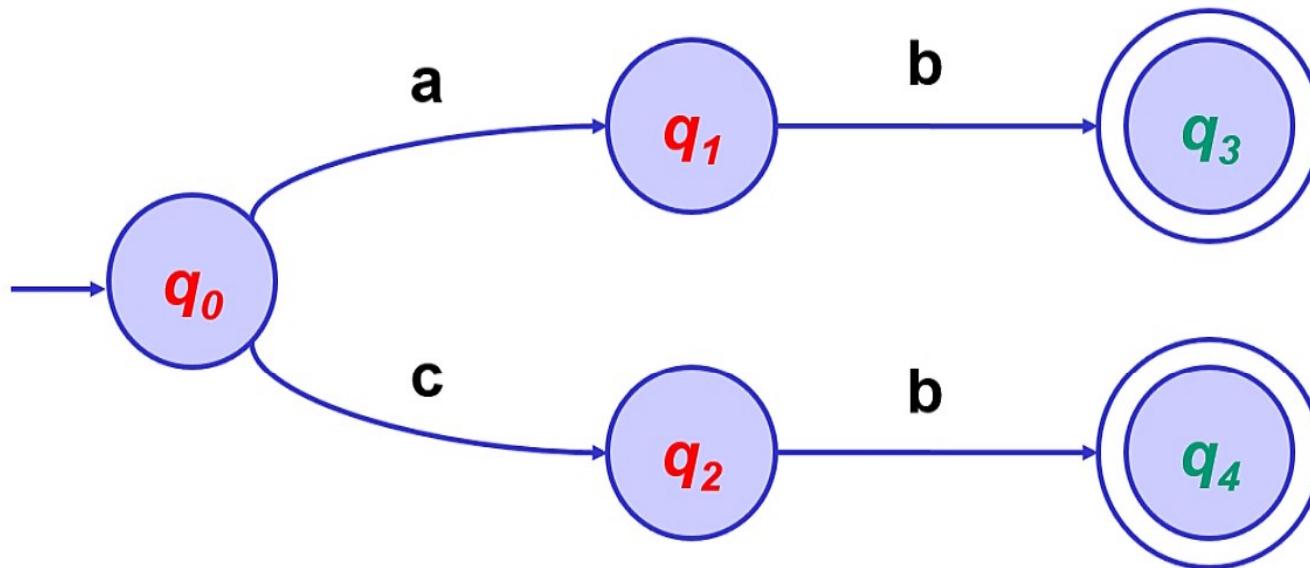
For any string w in set Σ^* holds the following:

$$\delta(p, w) \in F \wedge \delta'(p', w) \in F'$$

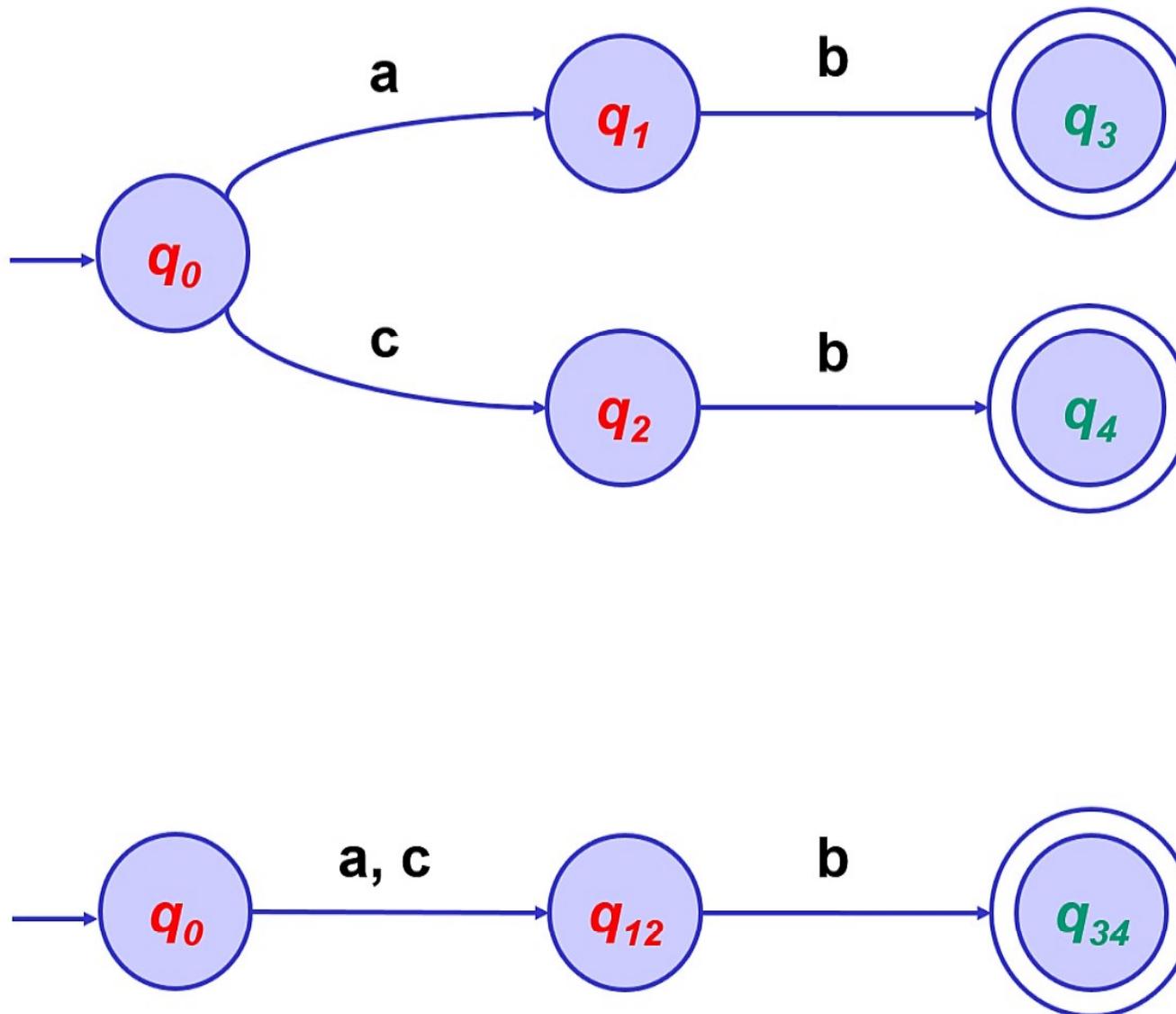
or

$$\delta(p, w) \notin F \wedge \delta'(p', w) \notin F'$$

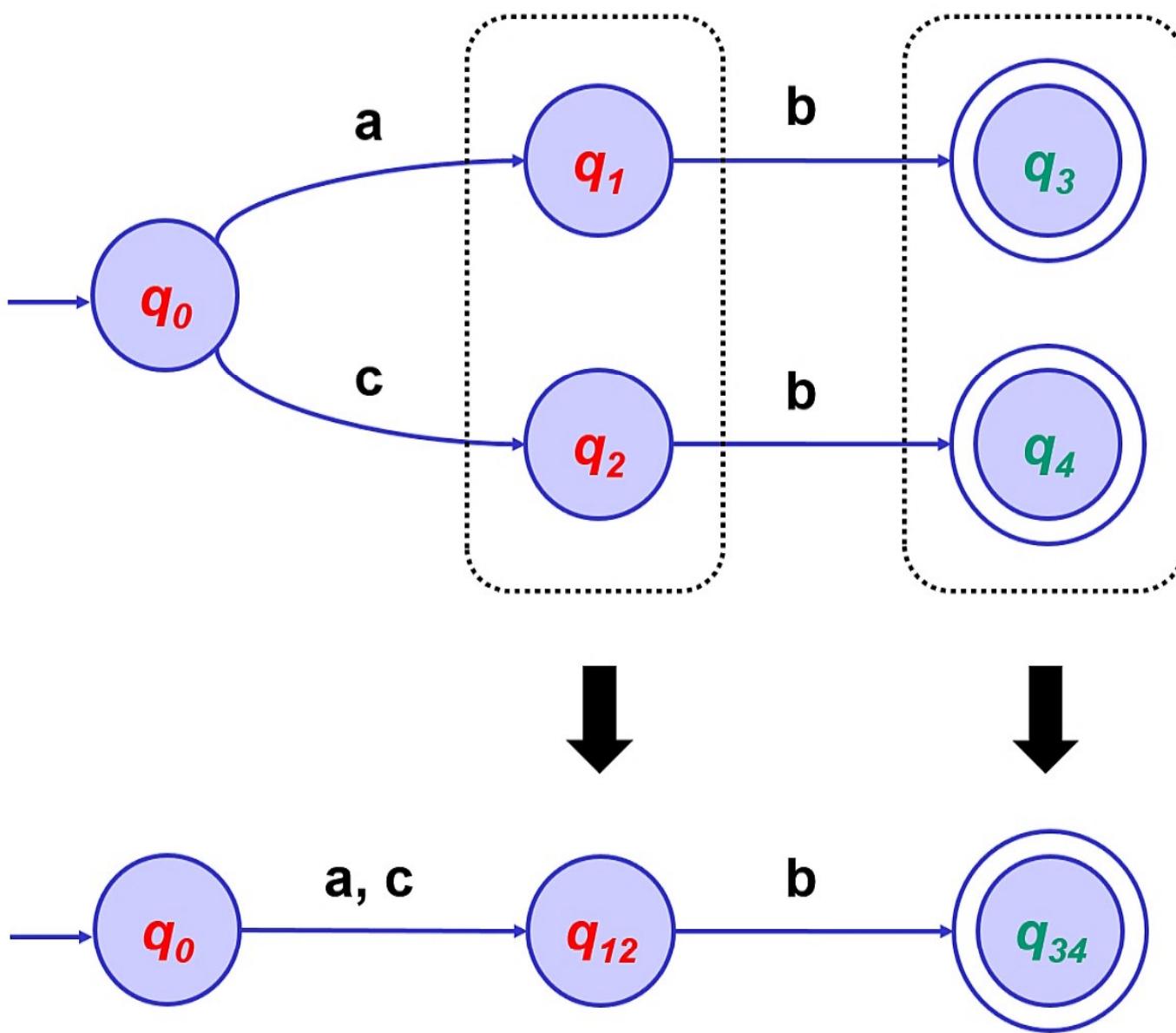
Minimization of DFA



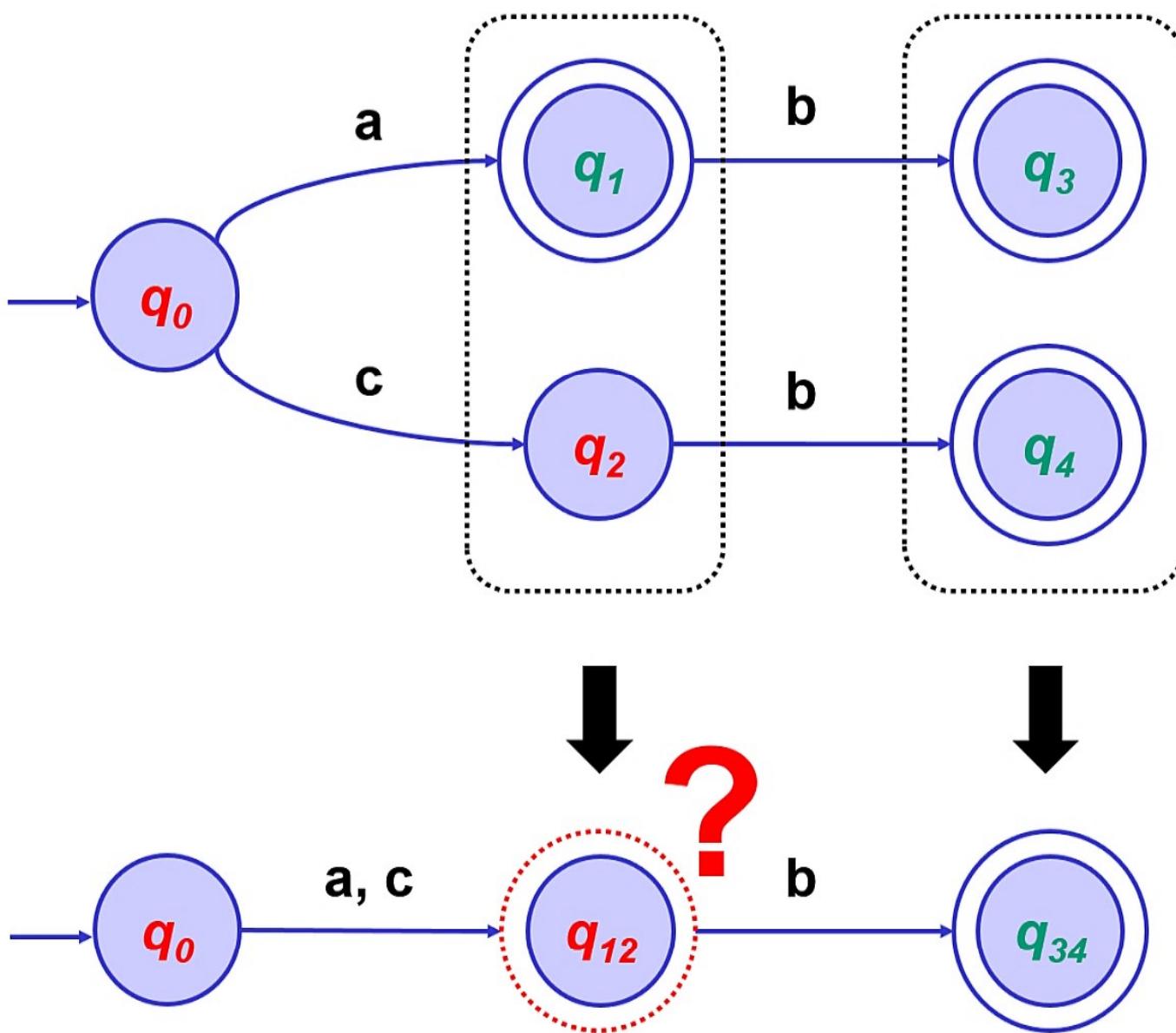
Minimization of DFA



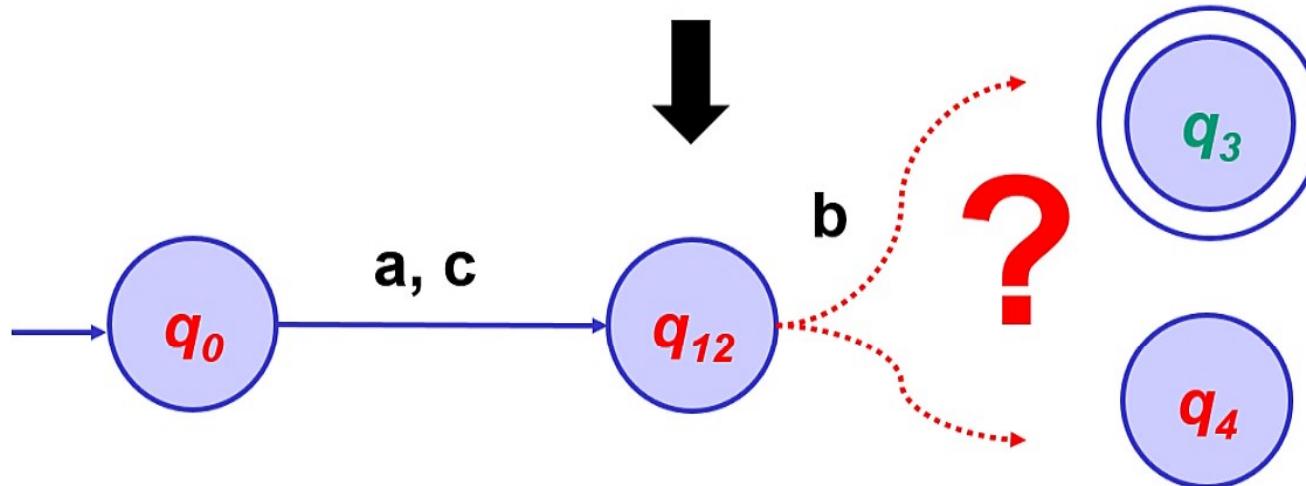
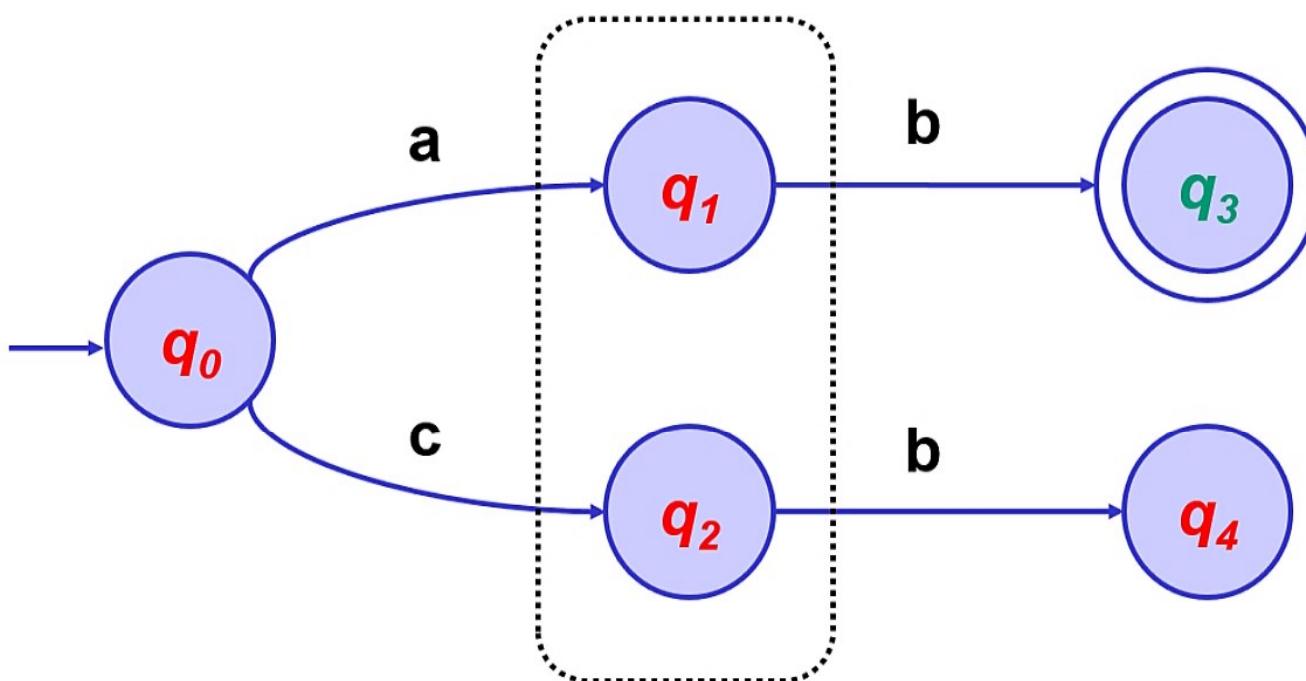
q_1 and q_2 are equivalent
 q_3 and q_4 are equivalent



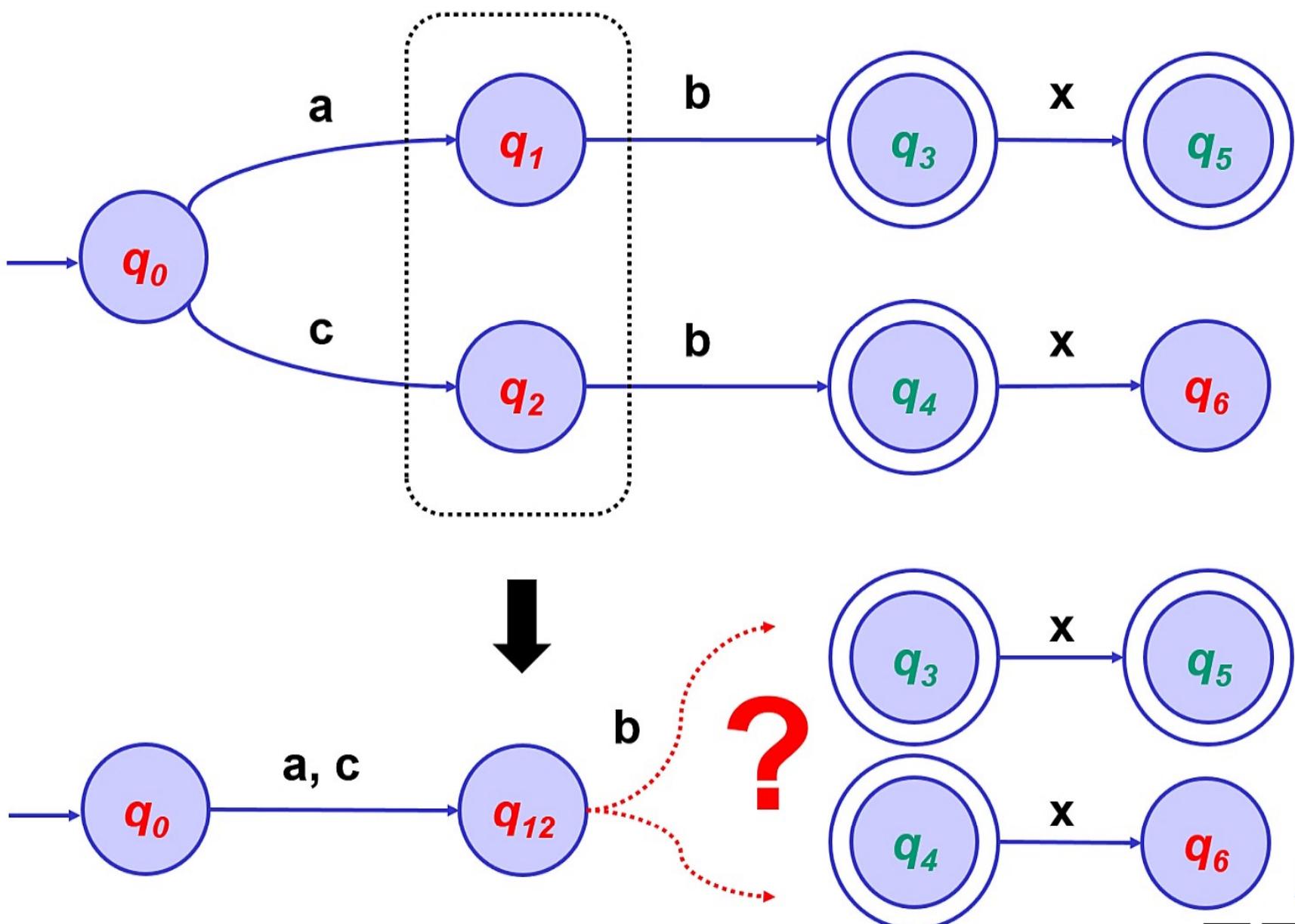
q_1 and q_2 are NOT equivalent
 q_3 and q_4 are equivalent



q_3 and q_4 are NOT equivalent \Rightarrow
 q_1 and q_2 are NOT equivalent



q_5 and q_6 are NOT equivalent \Rightarrow
 q_3 and q_4 are NOT equivalent \Rightarrow
 q_1 and q_2 are NOT equivalent



Minimization of DFA

- Equivalence test for states p and q

- Testing of two conditions:

Compatibility condition:

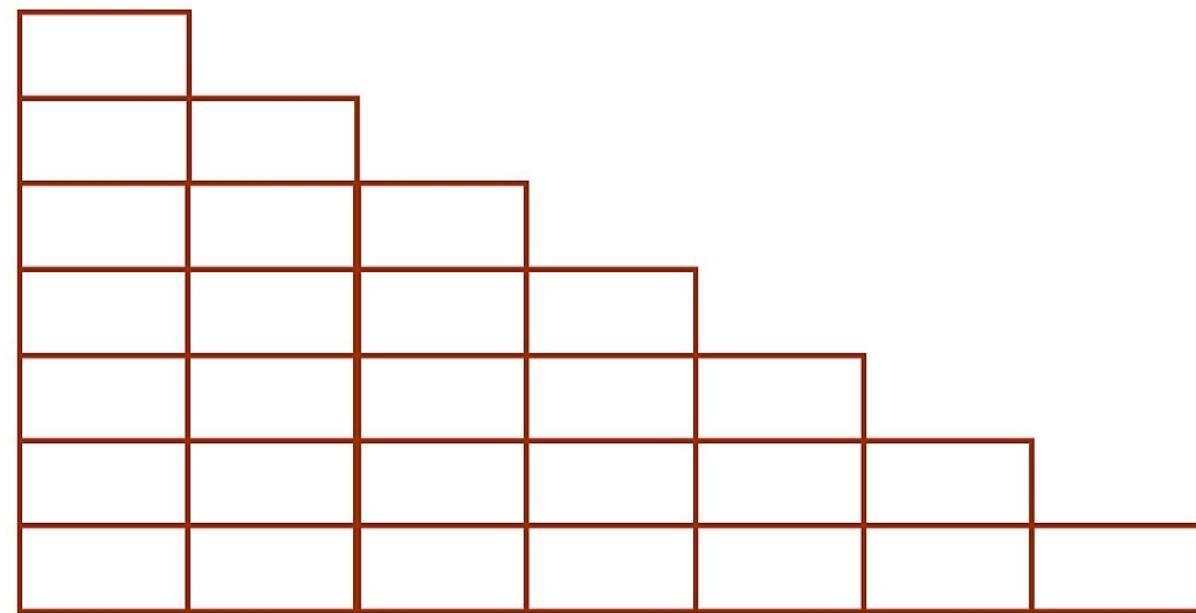
States p and q are either both accepting ($p \in F \wedge q \in F$) or both nonaccepting ($p \notin F \wedge q \notin F$).

Propagation condition:

For each input symbol a , states $\delta(p, a)$ and $\delta(q, a)$ are equivalent.

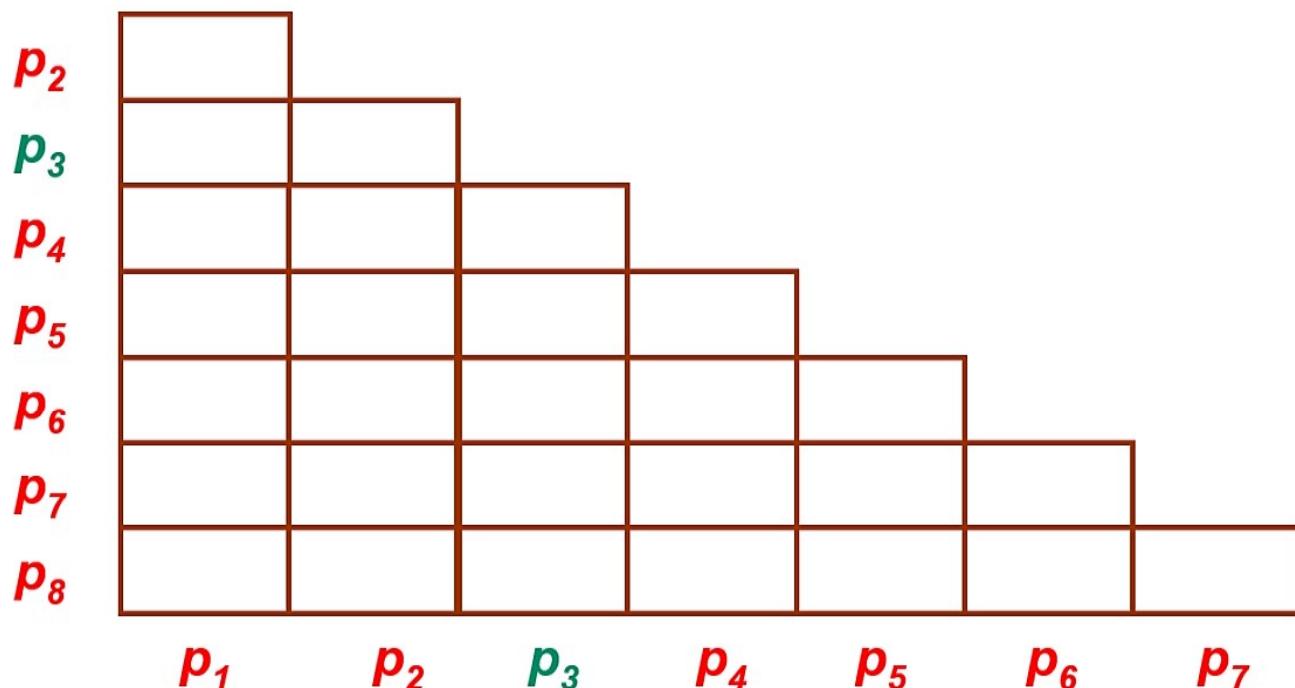
Minimization of DFA – *table-filling algorithm*

	0	1	\perp
p_1	p_2	p_6	0
p_2	p_7	p_3	0
p_3	p_1	p_3	1
p_4	p_3	p_7	0
p_5	p_8	p_6	0
p_6	p_3	p_7	0
p_7	p_7	p_5	0
p_8	p_7	p_3	0



Minimization of DFA – table-filling algorithm

	0	1	\perp
p_1	p_2	p_6	0
p_2	p_7	p_3	0
p_3	p_1	p_3	1
p_4	p_3	p_7	0
p_5	p_8	p_6	0
p_6	p_3	p_7	0
p_7	p_7	p_5	0
p_8	p_7	p_3	0



1

Mark all pairs (p, q) for which holds that $p \in F$ and $q \notin F$

Minimization of DFA – *table-filling algorithm*

	0	1
p_1	p_2	p_6
p_2	p_7	p_3
p_3	p_1	p_3
p_4	p_3	p_7
p_5	p_8	p_6
p_6	p_3	p_7
p_7	p_7	p_5
p_8	p_7	p_3

A diagram illustrating a sequence of 8 horizontal bars, each labeled p_i at the bottom. The bars are colored red, orange, or yellow. The first bar (p_1) is red. The second bar (p_2) is orange and contains two red 'X' marks. The third bar (p_3) is yellow. The fourth bar (p_4) is orange and contains one red 'X' mark. The fifth bar (p_5) is red. The sixth bar (p_6) is orange and contains one red 'X' mark. The seventh bar (p_7) is yellow. The eighth bar (p_8) is orange and contains one red 'X' mark.

1

Mark all pairs (p, q) for which holds that $p \in F$ and $q \notin F$

Minimization of DFA – table-filling algorithm

	0	1	\perp	p_2							
p_1	p_2	p_6	0	p_3	X	X					
p_2	p_7	p_3	0	p_4			X				
p_3	p_1	p_3	1	p_5			X				
p_4	p_3	p_7	0	p_6			X				
p_5	p_8	p_6	0	p_7			X				
p_6	p_3	p_7	0	p_8			X				
p_7	p_7	p_5	0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	
p_8	p_7	p_3	0								

2

For any unmarked pair of states (p, q)
 if the pair $(\delta(p,a), \delta(q,a))$ is already marked
 Mark (p,q) ;
 Recursively mark all the pairs in the list assigned to the marked pair
 else
 if $\delta(p,a) \neq \delta(q,a)$
 put the pair (p,q) into the list assigned to the pair $(\delta(p,a), \delta(q,a))$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3	0	

0	p_2					
0	p_3	X	X			
1	p_4			X		
0	p_5			X		
0	p_6		X			
0	p_7		X			
0	p_8		X			

2

For any unmarked pair of states (p, q)
 if the pair $(\delta(p,a), \delta(q,a))$ is already marked
 Mark (p,q) ;
 Recursively mark all the pairs in the list assigned to the marked pair
 else
 if $\delta(p,a) \neq \delta(q,a)$
 put the pair (p,q) into the list assigned to the pair $(\delta(p,a), \delta(q,a))$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3	0	p_1

	0	1	\perp	
p_2	X			p_2
p_3	X	X		p_3
p_4			X	p_4
p_5			X	p_5
p_6			X	p_6
p_7			X	p_7
p_8			X	

2

For any unmarked pair of states (p, q)
 if the pair $(\delta(p,a), \delta(q,a))$ is already marked
 Mark (p,q) ;
 Recursively mark all the pairs in the list assigned to the marked pair
 else
 if $\delta(p,a) \neq \delta(q,a)$
 put the pair (p,q) into the list assigned to the pair $(\delta(p,a), \delta(q,a))$

Minimization of DFA – table-filling algorithm

	0	1
	0	1
p_1	p_2	p_6
p_2	p_7	p_3
p_3	p_1	p_3
p_4	p_3	p_7
p_5	p_8	p_6
p_6	p_3	p_7
p_7	p_7	p_5
p_8	p_7	p_3

	0	1	0	0	0	0	0	0	0
	0	1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_1			X						
p_2				X					
p_3					X				
p_4						X			
p_5							X		
p_6								X	
p_7									X
p_8									

Minimization of DFA – table-filling algorithm

	0	1	\perp
p_1	p_2	p_6	
p_2	p_7	p_3	
p_3	p_1	p_3	
p_4	p_3	p_7	
p_5	p_8	p_6	
p_6	p_3	p_7	
p_7	p_7	p_5	
p_8	p_7	p_3	

	0	1	\perp	p_2	p_3	p_4	p_5	p_6	p_7
p_2	0			X					
p_3	0			X	X				
p_4	1			X		X			
p_5	0					X			
p_6	0					X			
p_7	0					X			
p_8	0					X			

Minimization of DFA – *table-filling algorithm*

	0	1
p_1	p_2	p_6
p_2	p_7	p_3
p_3	p_1	p_3
p_4	p_3	p_7
p_5	p_8	p_6
p_6	p_3	p_7
p_7	p_7	p_5
p_8	p_7	p_3

A diagram illustrating a sequence of points p_2 through p_9 . The points are arranged in a staircase pattern, where each point p_i has a horizontal range from p_{i-1} to p_{i+1} . The points are marked with red 'X' symbols at specific positions:

- p_2 is marked at its range boundary.
- p_3 is marked at both its range boundaries.
- p_4 is marked at its range boundaries.
- p_5 is unmarked.
- p_6 is marked at its range boundary.
- p_7 is unmarked.
- p_8 is marked at its range boundary.

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3	0	p_1

	0	1	\perp	
p_2	X			p_2
p_3	X	X		p_3
p_4	X		X	p_4
p_5			X	p_5
p_6			X	p_6
p_7			X	p_7
p_8			X	p_8

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Minimization of DFA – *table-filling algorithm*

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		

0	p_2	X					
0	p_3	X	X				
1	p_4	X		X			
0	p_5			X			
0	p_6	X		X			
0	p_7			X			
0	p_8	X		X			

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Minimization of DFA – *table-filling algorithm*

The figure displays two sets of 8x8 matrices, each with a specific color scheme and labeling.

- Left Set:** Labeled with indices p_1 through p_8 . The first row (p_1) has values p_2 and p_6 . The second row (p_2) has values p_7 and p_3 . The third row (p_3) has values p_1 and p_3 . The fourth row (p_4) has values p_3 and p_7 . The fifth row (p_5) has values p_8 and p_6 . The sixth row (p_6) has values p_3 and p_7 . The seventh row (p_7) has values p_7 and p_5 . The eighth row (p_8) has values p_7 and p_3 .
- Right Set:** Labeled with indices p_2 through p_8 . The first row (p_2) has value X . The second row (p_3) has values X and X . The third row (p_4) has value X . The fourth row (p_5) has value X . The fifth row (p_6) has value X . The sixth row (p_7) has value X . The seventh row (p_8) has values X and X .

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_2, p_7) = \{(p_1, p_7)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – *table-filling algorithm*

The diagram illustrates a 2D convolution operation with a stride of 2. The input layer consists of 8 green units labeled p_1 through p_8 . The output layer consists of 8 orange units labeled p_2 through p_8 . Red 'X' marks indicate the receptive fields of each output unit. Specifically:

- p_2 receives input from p_1 and p_3 .
- p_3 receives input from p_2 and p_4 .
- p_4 receives input from p_3 , p_5 , and p_7 .
- p_5 receives input from p_4 and p_6 .
- p_6 receives input from p_5 , p_7 , and p_8 .
- p_7 receives input from p_6 and p_8 .
- p_8 receives input from p_7 .

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_2, p_7) = \{(p_1, p_7)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		

	0	1	\perp	
p_2	X			p_2
p_3	X	X		p_3
p_4	X	X	X	p_4
p_5		X	X	p_5
p_6	X	X	X	p_6
p_7		X	X	p_7
p_8	X		X	

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_2, p_7) = \{(p_1, p_7)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp	
p_2	X			p_2
p_3	X	X		p_3
p_4	X	X	X	
p_5		X	X	
p_6	X	X	X	
p_7	X	X	X	
p_8	X		X	

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
0							

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_2, p_7) = \{(p_1, p_7)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp		
p_2	X			p_2	
p_3	X	X		p_3	
p_4	X	X	X	p_4	
p_5		X	X	X	p_5
p_6	X	X	X		p_6
p_7	X	X	X	X	p_7
p_8	X		X	X	0

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp		
p_2	X			p_2	
p_3	X	X		p_3	
p_4	X	X	X	p_4	
p_5		X	X	X	p_5
p_6	X	X	X		p_6
p_7	X	X	X	X	p_7
p_8	X		X	X	0

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp		p_1	p_2	p_3	p_4	p_5	p_6	p_7
p_2	X										
p_3	X	X									
p_4	X	X	X								
p_5		X	X	X	X						
p_6	X	X	X						X		
p_7	X	X	X	X							
p_8	X		X	X							

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

List assigned to $(p_5, p_6) = \{(p_1, p_7)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp		
p_2	X			p_2	
p_3	X	X		p_3	
p_4	X	X	X	p_4	
p_5		X	X	X	p_5
p_6	X	X	X		p_6
p_7	X	X	X	X	p_7
p_8	X		X	X	p_8
					0

$p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_6 \quad p_7$

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Minimization of DFA – table-filling algorithm

	0	1	\perp	
p_1	p_2	p_6		p_2
p_2	p_7	p_3		p_3
p_3	p_1	p_3		p_4
p_4	p_3	p_7		p_5
p_5	p_8	p_6		p_6
p_6	p_3	p_7		p_7
p_7	p_7	p_5		p_8
p_8	p_7	p_3		0

	0	1	\perp		p_1	p_2	p_3	p_4	p_5	p_6	p_7
p_2	X										
p_3	X	X									
p_4	X	X	X								
p_5		X	X	X	X						
p_6	X	X	X						X		
p_7	X	X	X	X	X	X	X				
p_8	X		X	X	X	X	X	X	X	X	X

List assigned to $(p_2, p_8) = \{(p_1, p_5)\}$

Equivalent states:

$(p_1, p_5), (p_2, p_8), (p_4, p_6)$

Minimization of DFA – *table-filling algorithm*

	0	1	\perp
p_1	p_2	p_6	0
p_2	p_7	p_3	0
p_3	p_1	p_3	1
p_4	p_3	p_7	0
p_5	p_8	p_6	0
p_6	p_3	p_7	0
p_7	p_7	p_5	0
p_8	p_7	p_3	0

0	1	\perp
$[p_1, p_5]$	$[p_4, p_6]$	0
p_7	p_3	0
$[p_1, p_5]$	p_3	1
p_3	p_7	0
p_7	$[p_1, p_5]$	0

Unreachable States in DFA

	c	d	⊥	RS	
q_0	q_1	q_5	0	q_0	
q_1	q_2	q_7	1		
q_2	q_2	q_5	1		
q_3	q_5	q_7	0		
q_4	q_5	q_6	0		
q_5	q_3	q_1	0		
q_6	q_8	q_0	1		
q_7	q_0	q_1	1		
q_8	q_3	q_6	0		

- 1) Put the initial state q_0 into the list of reachable states RS
- 2) Extend the list RS with the states $\{p \mid p = \delta(q_0, a), \text{ za sve } a \in \Sigma\}$
- 3) $q_i \in RS, RS = RS \cup \{p \mid p = \delta(q_i, a), \text{ for any } a \in \Sigma\}$

Unreachable States in DFA

	c	d	\perp	RS	q_0	q_1	q_5
q_0	q_1	q_5	0				
q_1	q_2	q_7	1				
q_2	q_2	q_5	1				
q_3	q_5	q_7	0				
q_4	q_5	q_6	0				
q_5	q_3	q_1	0				
q_6	q_8	q_0	1				
q_7	q_0	q_1	1				
q_8	q_3	q_6	0				

- 1) Put the initial state q_0 into the list of reachable states RS
- 2) Extend the list RS with the states $\{p \mid p = \delta(q_0, a), \text{ za sve } a \in \Sigma\}$
- 3) $q_i \in RS, RS = RS \cup \{p \mid p = \delta(q_i, a), \text{ for any } a \in \Sigma\}$

Unreachable States in DFA

	c	d	⊥	RS	q_0	q_1	q_5	q_2	q_7	q_3
q_0	q_1	q_5		0						
q_1	q_2	q_7		1						
q_2	q_2	q_5		1						
q_3	q_5	q_7		0						
q_4	q_5	q_6		0						
q_5	q_3	q_1		0						
q_6	q_8	q_0		1						
q_7	q_0	q_1		1						
q_8	q_3	q_6		0						

- 1) Put the initial state q_0 into the list of reachable states RS
- 2) Extend the list RS with the states $\{p \mid p = \delta(q_0, a), \text{ za sve } a \in \Sigma\}$
- 3) $q_i \in RS, RS = RS \cup \{p \mid p = \delta(q_i, a), \text{ for any } a \in \Sigma\}$

Unreachable States in DFA

	c	d	⊥	RS	q_0	q_1	q_5	q_2	q_7	q_3
q_0	q_1	q_5		0						
q_1	q_2	q_7		1						
q_2	q_2	q_5		1						
q_3	q_5	q_7		0						
q_4	q_5	q_6		0						
q_5	q_3	q_1		0						
q_6	q_8	q_0		1						
q_7	q_0	q_1		1						
q_8	q_3	q_6		0						

- 1) Put the initial state q_0 into the list of reachable states RS
- 2) Extend the list RS with the states $\{p \mid p = \delta(q_0, a), \text{ za sve } a \in \Sigma\}$
- 3) $q_i \in RS, RS = RS \cup \{p \mid p = \delta(q_i, a), \text{ for any } a \in \Sigma\}$

Unreachable States in DFA

	<i>c</i>	<i>d</i>	\perp	<i>RS</i>	q_0	q_1	q_5	q_2	q_7	q_3
q_0	q_1	q_5		0						
q_1	q_2	q_7		1						
q_2	q_2	q_5		1						
q_3	q_5	q_7		0						
q_4	q_5	q_6		0						
q_5	q_3	q_1		0						
q_6	q_8	q_0		1						
q_7	q_0	q_1		1						
q_8	q_3	q_6		0						

	<i>c</i>	<i>d</i>	\perp
q_0	q_1	q_5	0
q_1	q_2	q_7	1
q_2	q_2	q_5	1
q_3	q_5	q_7	0
q_5	q_3	q_1	0
q_7	q_0	q_1	1
q_8	q_3	q_6	1

- 1) Put the initial state q_0 into the list of reachable states *RS*
- 2) Extend the list *RS* with the states $\{p \mid p = \delta(q_0, a), \text{ za sve } a \in \Sigma\}$
- 3) $q_i \in RS, RS = RS \cup \{p \mid p = \delta(q_i, a), \text{ for any } a \in \Sigma\}$

Minimal DFA

	<i>c</i>	<i>d</i>	\perp
p_0	p_0	p_3	0
p_1	p_2	p_5	0
p_2	p_2	p_7	0
p_3	p_6	p_7	0
p_4	p_1	p_6	1
p_5	p_6	p_5	0
p_6	p_6	p_3	1
p_7	p_6	p_3	0

	<i>c</i>	<i>d</i>	\perp
A	A	B	0
B	C	B	0
C	C	B	1

	<i>c</i>	<i>d</i>
p_0, A	p_0, A	p_3, B
p_3, B	p_6, C	p_7, B
p_6, C	p_6, C	p_3, B
p_7, B	p_6, C	p_3, B