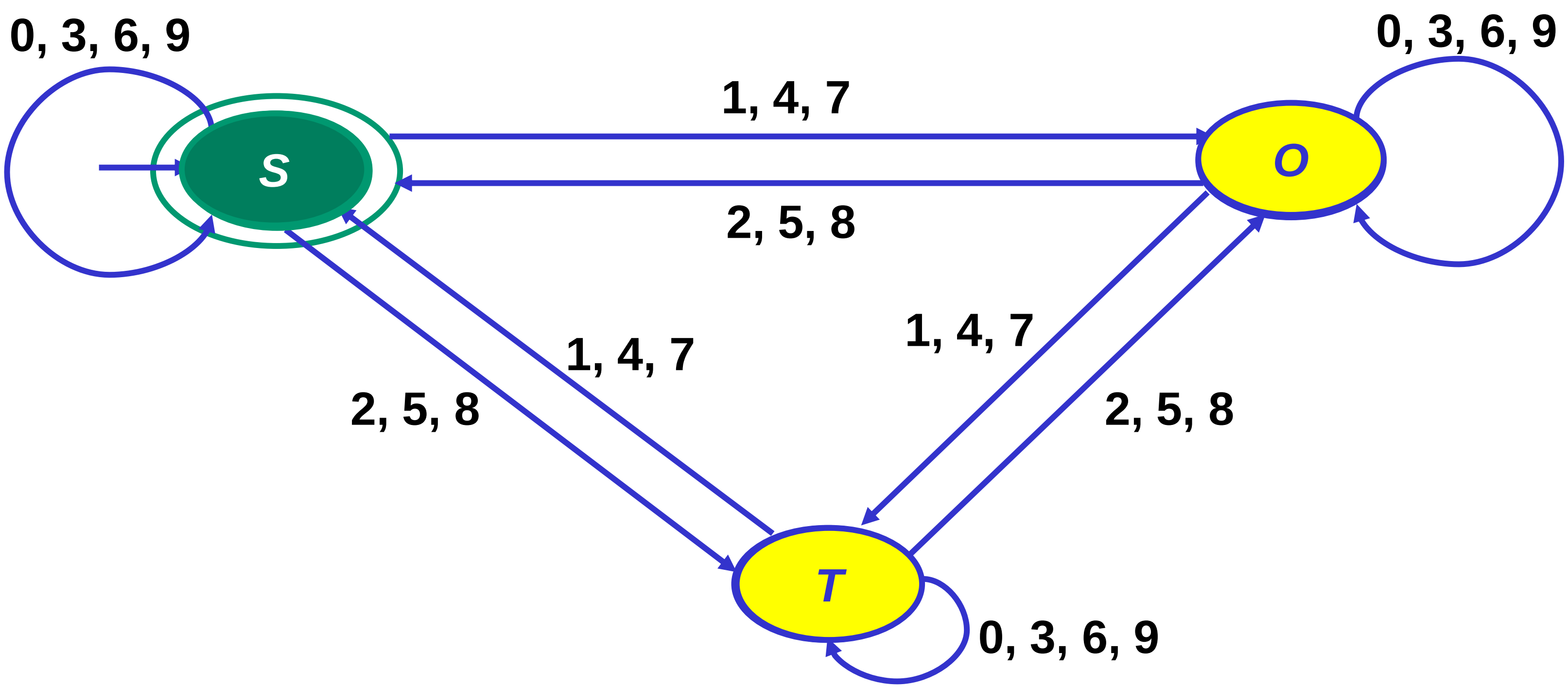
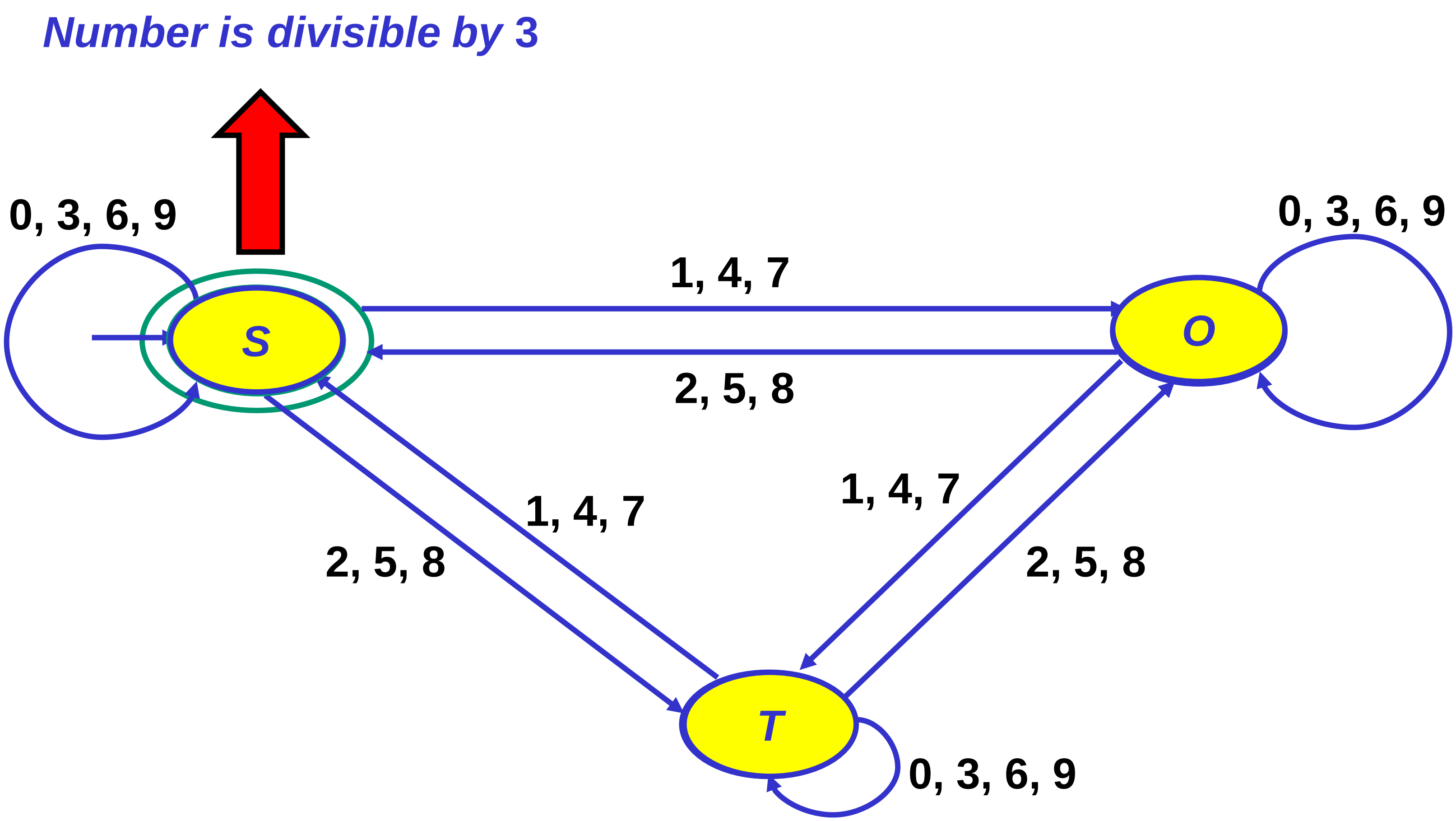
4th lecture overview

Lecture overview

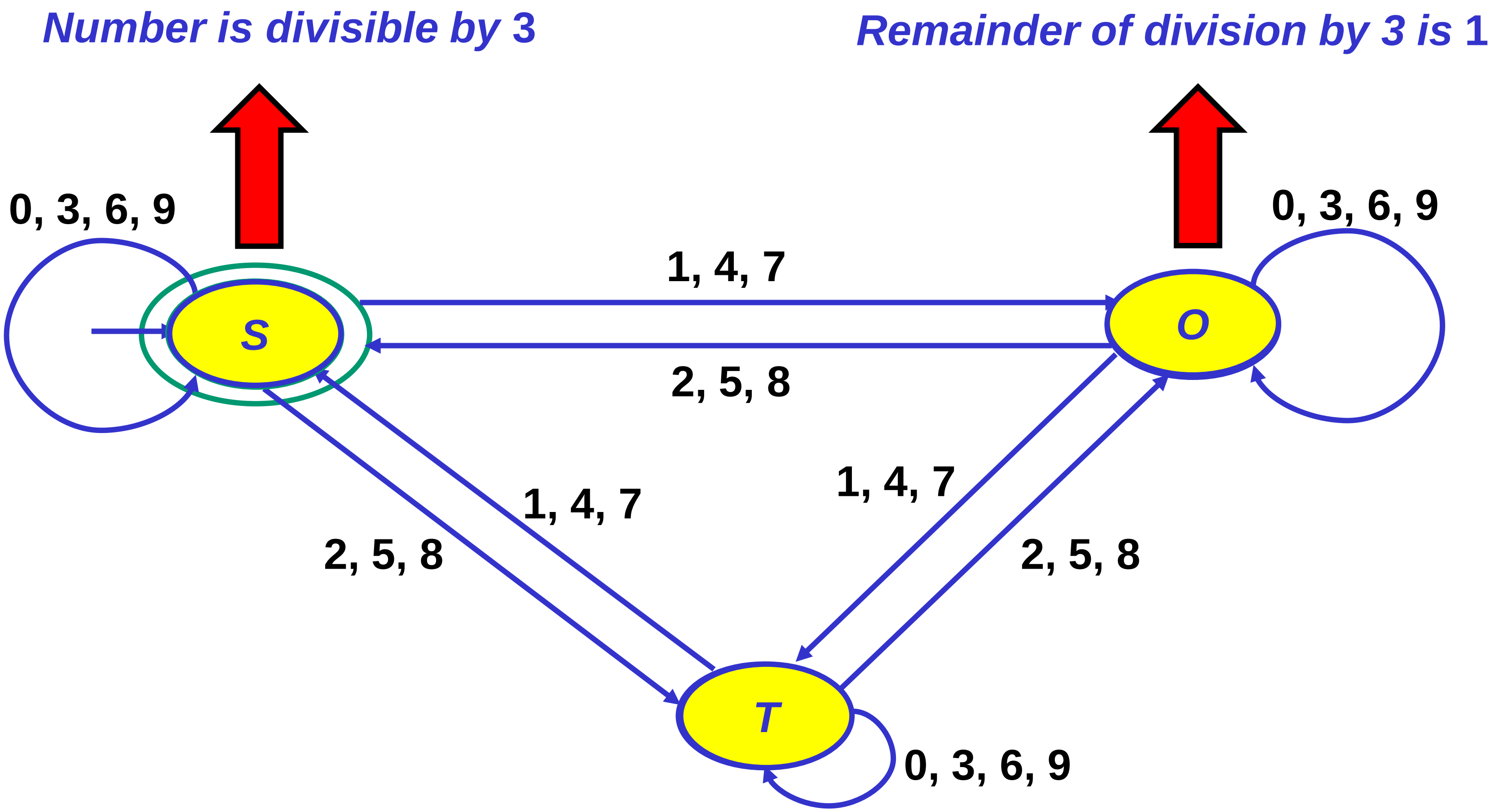
Finite state machines with output



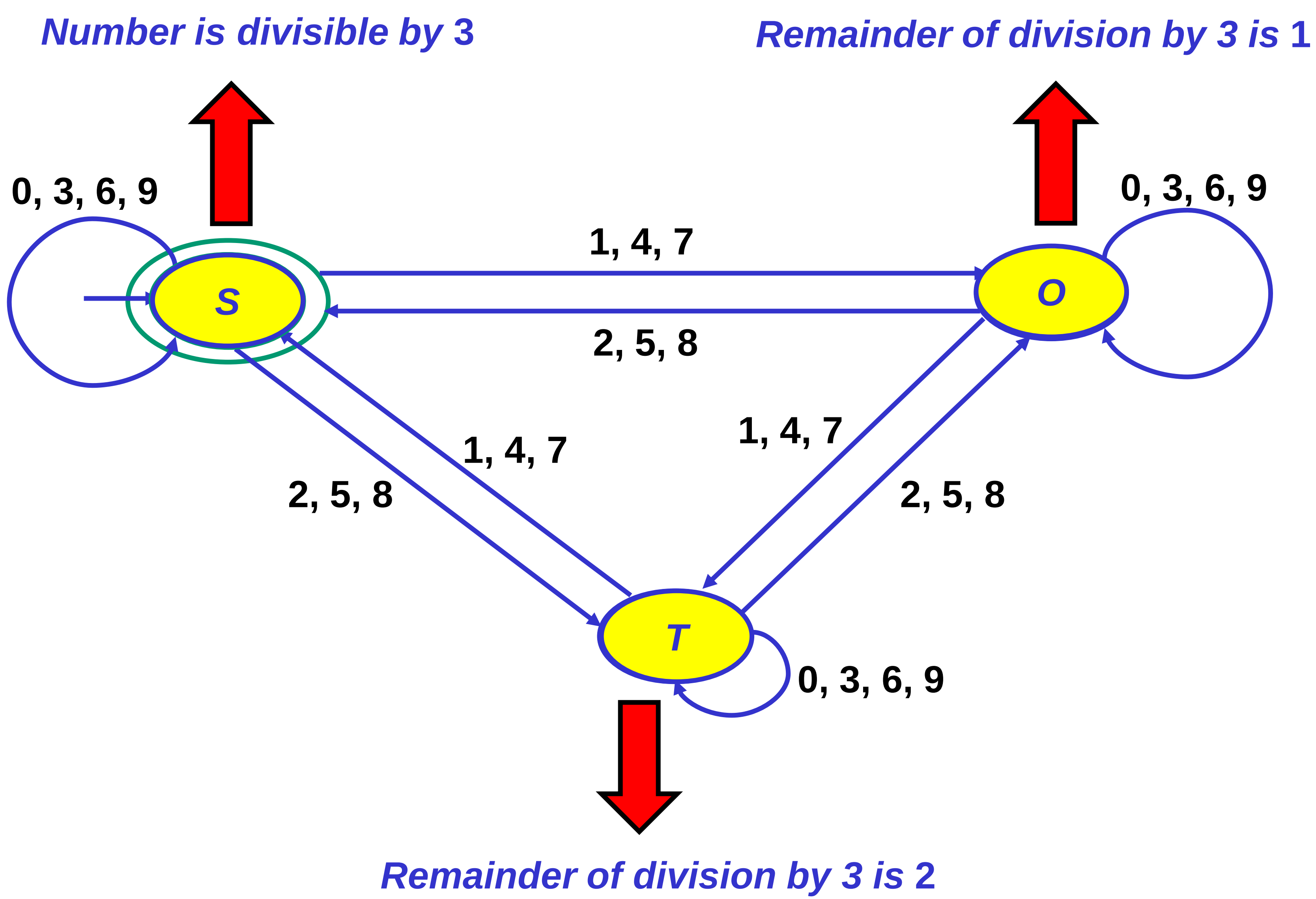
Finite state machines with output



Finite state machines with output



Finite state machines with output



Finite state machines with output

- **Moore machine**
    - **The output is a function of the state**
- **Mealy machine**
    - **The output is a function of both the state and the input symbol**

Moore machine

$$MoDfa = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

Q	- finite set of states
Σ	- finite set of input symbols
Δ	- finite set of output symbols
δ	- transition function $Q \times \Sigma \rightarrow Q$
λ	- output function $Q \rightarrow \Delta$
$q_0 \in Q$	- start state

Moore machine

q_0

$\lambda(q_0)$

Moore machine

$$\begin{array}{ccc} & a_1 & \\ q_0 & \rightarrow & q_1 \end{array}$$

$\boxed{\lambda(q_0)}$ $\boxed{\lambda(q_1)}$

Moore machine

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2$$

$$\boxed{\lambda(q_0)} \qquad \boxed{\lambda(q_1)} \qquad \boxed{\lambda(q_2)}$$

Moore machine

	a_1		a_2		a_3			
q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	- -	- -	q_{n-1}
$\lambda(q_0)$		$\lambda(q_1)$		$\lambda(q_2)$				$\lambda(q_{n-1})$

Moore machine

$$\begin{array}{ccccccccccc}
 & a_1 & & a_2 & & a_3 & & & & a_n & \\
q_0 & \rightarrow & q_1 & \rightarrow & q_2 & \rightarrow & -\,- & & -\,- & q_{n-1} & \rightarrow & q_n \\
\boxed{\lambda(q_0)} & & \boxed{\lambda(q_1)} & & \boxed{\lambda(q_2)} & & & & \boxed{\lambda(q_{n-1})} & & \boxed{\lambda(q_n)}
\end{array}$$

Moore machine

w	$w0$	$w1$
i	$2i$	$2i+1$

Moore machine

w	w0	w1
i	$2i$	$2i+1$

$i\%3$	$(2i)\%3$	$(2i+1)\%3$
0		
1		
2		

Moore machine

w	$w0$	$w1$
i	$2i$	$2i+1$

$i\%3$	$(2i)\%3$	$(2i+1)\%3$
0	0	1
1	2	0
2	1	2

Moore machine

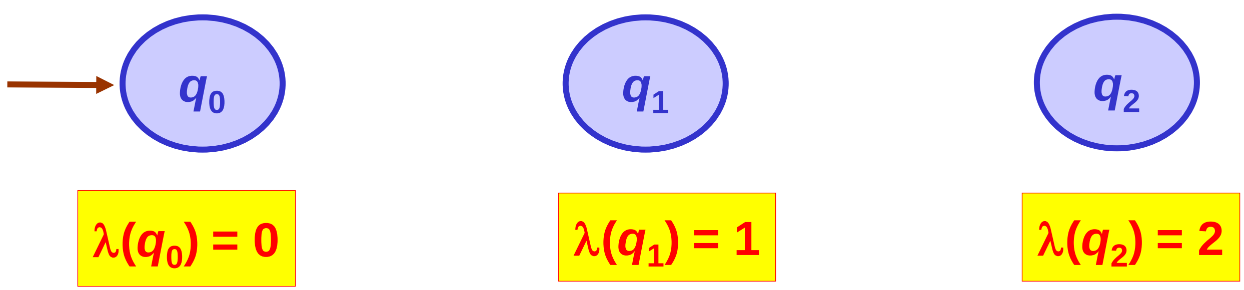
w	*w*0	*w*1
i	2*i*	2*i*+1

i%3	(2*i*)%3	(2*i*+1)%3
0	0	1
1	2	0
2	1	2

Moore machine

w	*w*0	*w*1
i	2*i*	2*i*+1

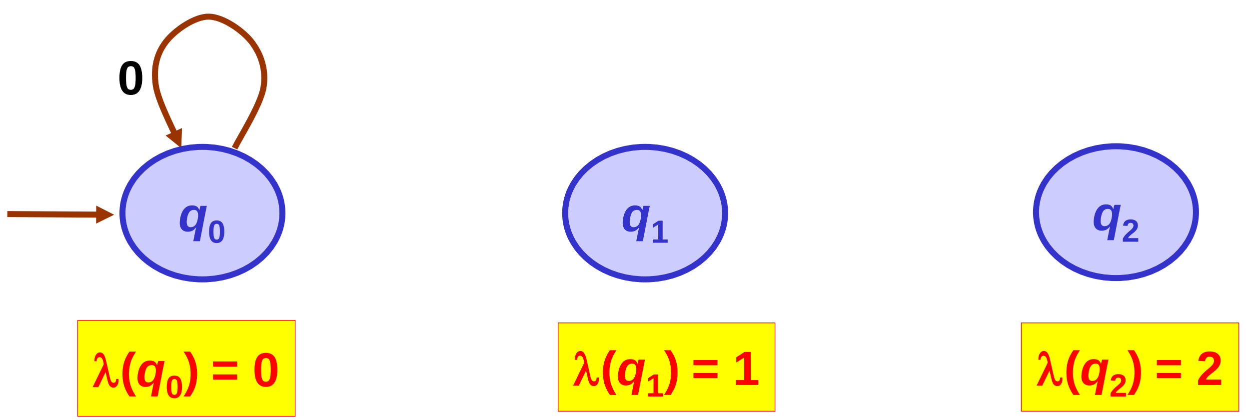
i%3	(2*i*)%3	(2*i*+1)%3
0	0	1
1	2	0
2	1	2



Moore machine

w	*w*0	*w*1
i	2*i*	2*i*+1

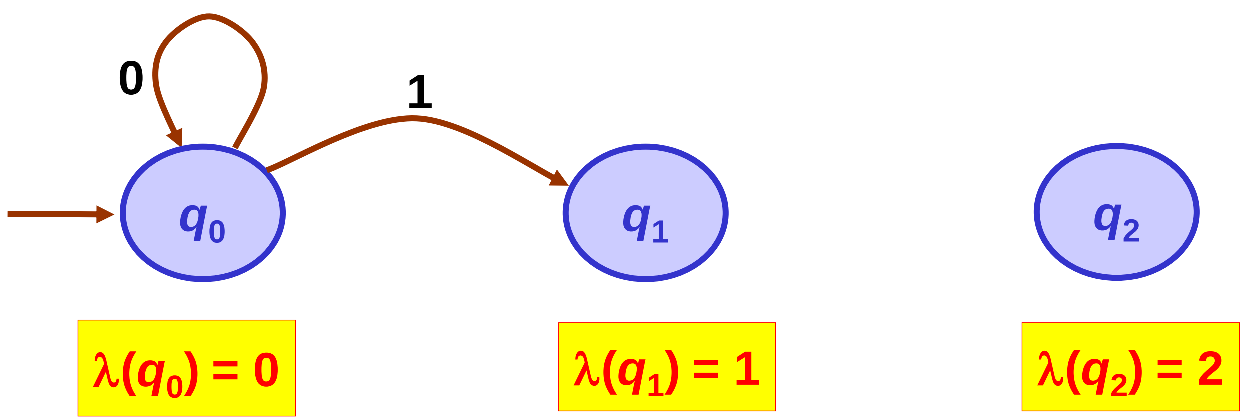
i%3	(2*i*)%3	(2*i*+1)%3
0	0	1
1	2	0
2	1	2



Moore machine

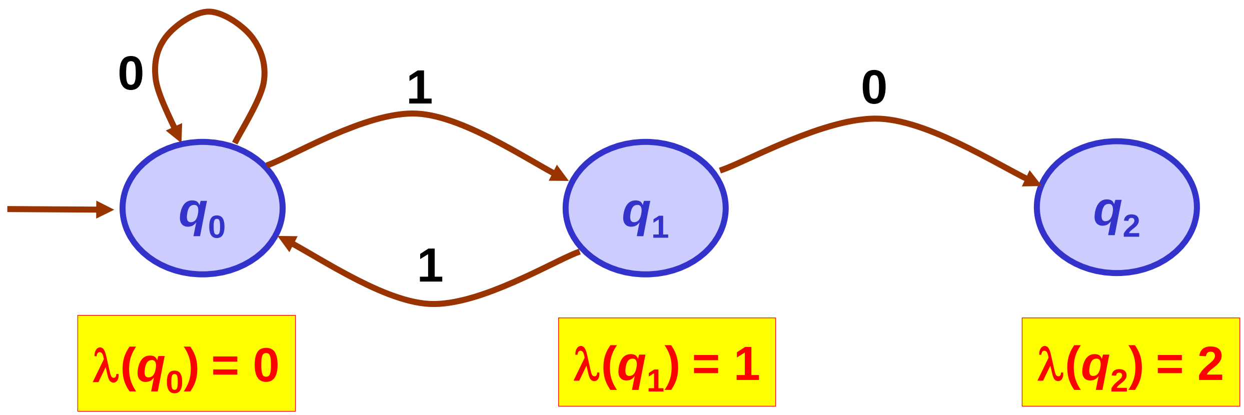
w	$w0$	$w1$
i	$2i$	$2i+1$

$i\%3$	$(2i)\%3$	$(2i+1)\%3$
0	0	1
1	2	0
2	1	2



Moore machine

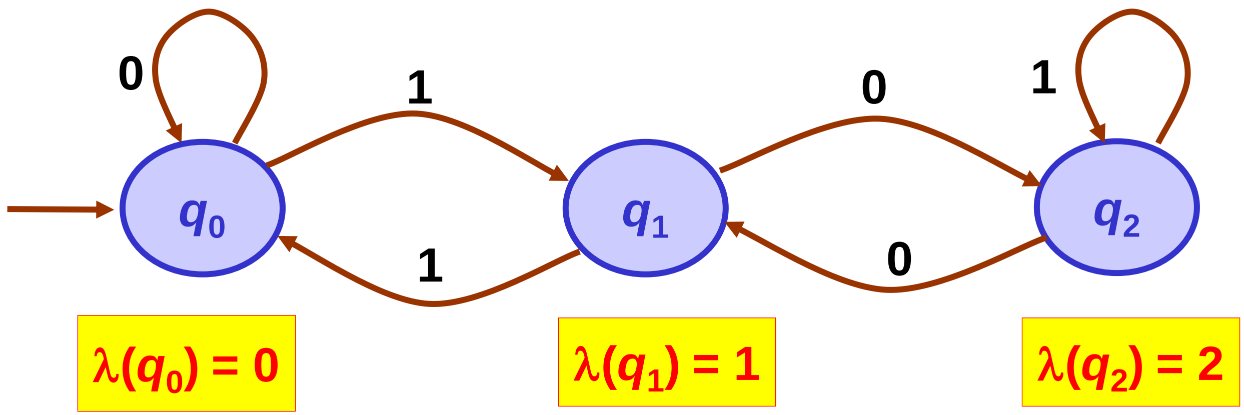
w	*w*0	*w*1
i	2*i*	2*i*+1
i%3	(2*i*)%3	(2*i*+1)%3
0	0	1
1	2	0
2	1	2



Moore machine

w	*w*0	*w*1
i	2*i*	2*i*+1

i%3	(2*i*)%3	(2*i*+1)%3
0	0	1
1	2	0
2	1	2



Moore machine

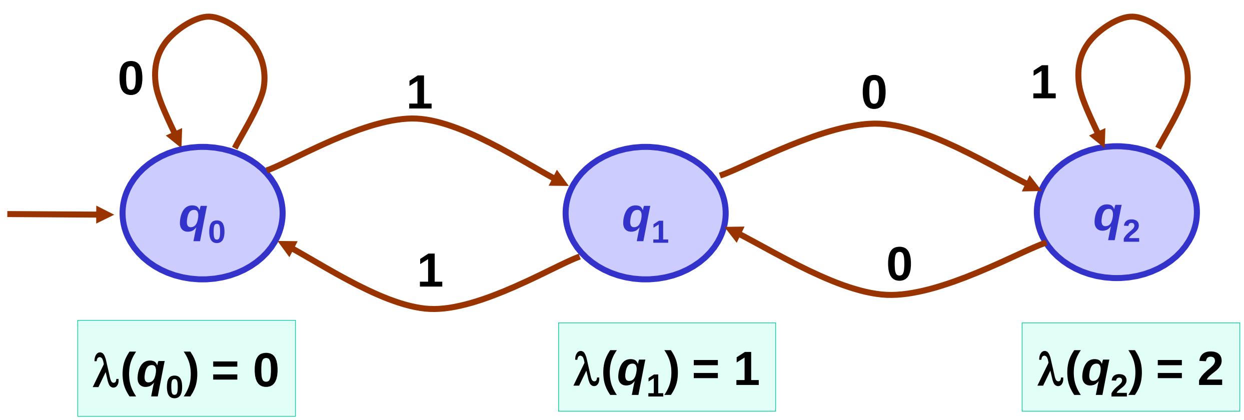
Read prefix ε

Integer value

q_0

Output sequence

$\lambda(q_0)=0$



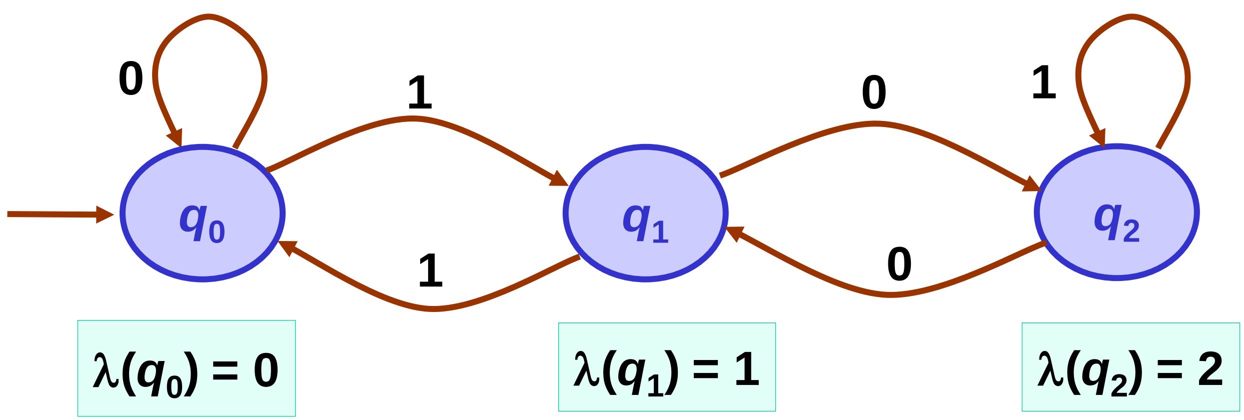
Moore machine

Read prefix	ε	**1**
Integer value		**1**

$$q_0 \xrightarrow{1} q_1$$

Output sequence

$\lambda(q_0)=0$ $\lambda(q_1)=1$

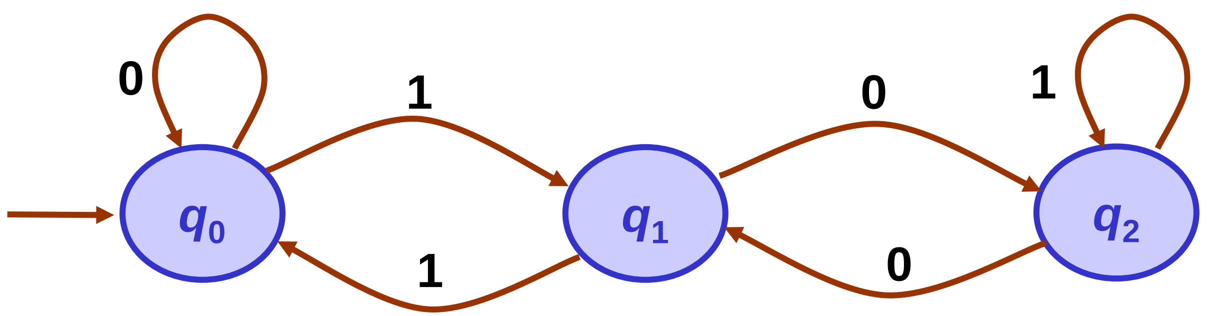


Moore machine

Read prefix	ε	**1**	**10**
Integer value		**1**	**2**

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2$$

Output sequence $\lambda(q_0)=0$ $\lambda(q_1)=1$ $\lambda(q_2)=2$



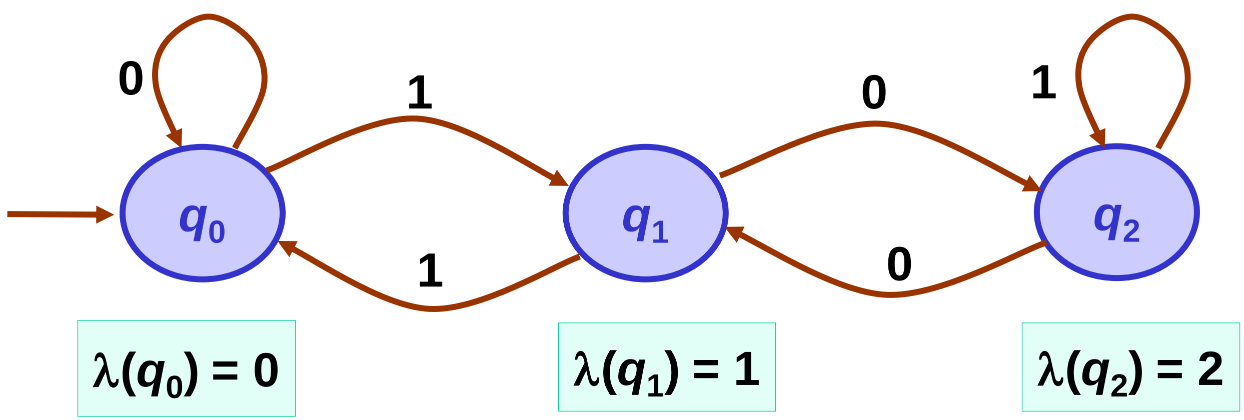
$\lambda(q_0) = 0$ $\lambda(q_1) = 1$ $\lambda(q_2) = 2$

Moore machine

Read prefix	ε	**1**	**10**	**101**
Integer value		**1**	**2**	**5**

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_2$$

Output sequence	$\lambda(q_0)=0$	$\lambda(q_1)=1$	$\lambda(q_2)=2$	$\lambda(q_2)=2$



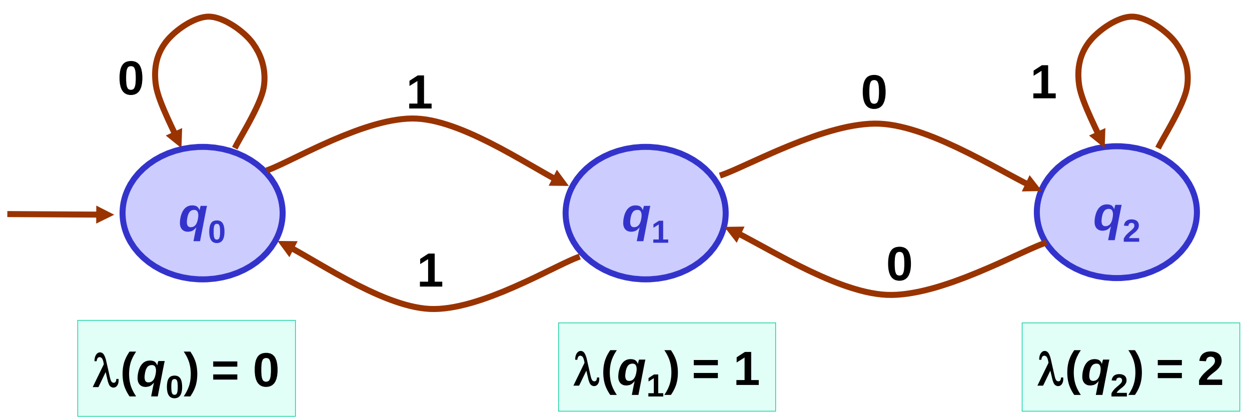
Moore machine

Read prefix	ε	**1**	**10**	**101**	**1010**
Integer value		**1**	**2**	**5**	**10**

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1$$

Output sequence

$\lambda(q_0)=0$	$\lambda(q_1)=1$	$\lambda(q_2)=2$	$\lambda(q_2)=2$	$\lambda(q_1)=1$



Mealy machine

$$MeDfa = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

Q	- finite set of states
Σ	- finite set of input symbols
Δ	- finite set of output symbols
δ	- transition function $Q \times \Sigma \rightarrow Q$
λ	- output function $Q \times \Sigma \rightarrow \Delta$
$q_0 \in Q$	- start state

Mealy machine

q_0

Mealy machine

$$\begin{array}{ccc} & a_1 & \\ q_0 & \rightarrow & q_1 \\ & \lambda(q_0, a_1) & \end{array}$$

Mealy machine

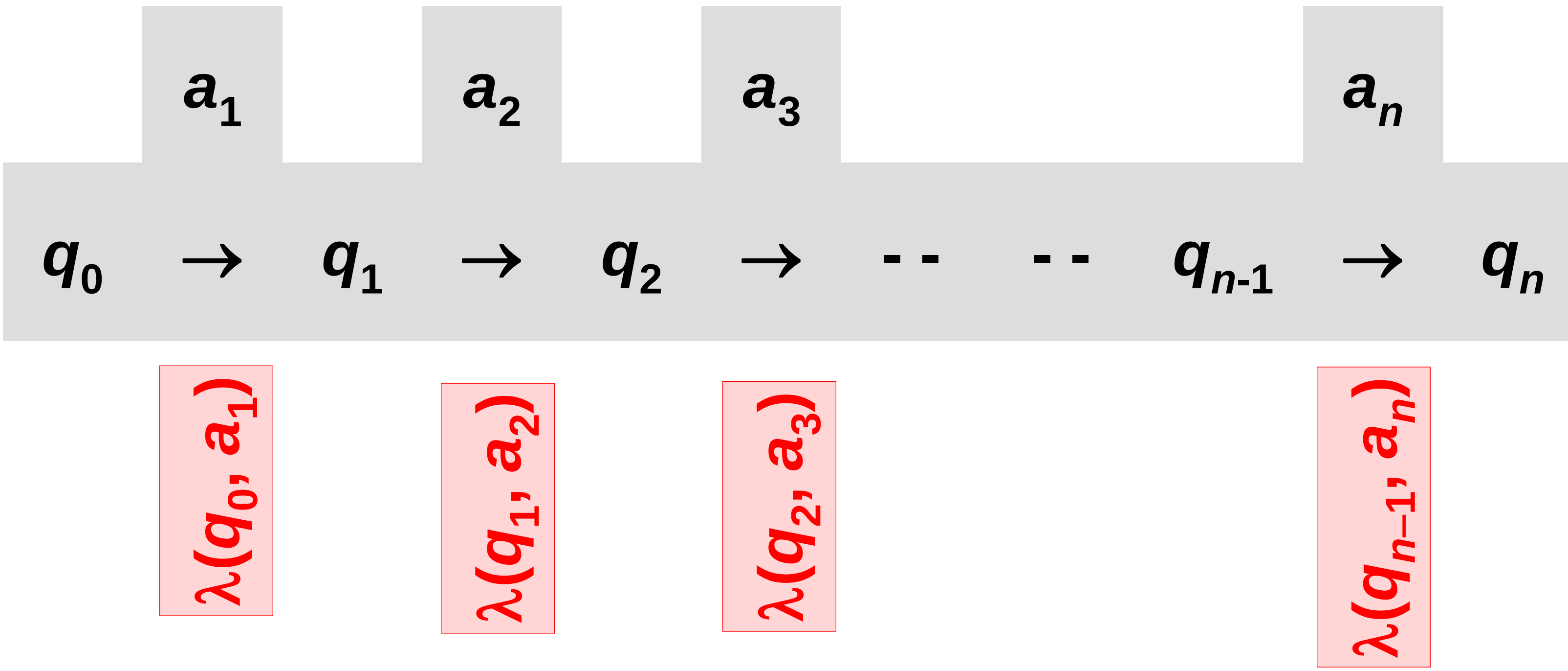
$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2$$

$\lambda(q_0, a_1)$ $\lambda(q_1, a_2)$

Mealy machine

	a_1		a_2		a_3			
q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	- -	- -	$q_{n\text{-}1}$
	$\lambda(q_0, a_1)$		$\lambda(q_1, a_2)$		$\lambda(q_2, a_3)$			

Mealy machine



Constructing Mealy machine for the given Moore machine

Mealy machine	Moore machine
$M' = (Q, \Sigma, \Delta, \delta, \lambda', q_0)$	$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$$b\, T_{M'}(w) \quad = \quad T_M(w)$$

$$1)\ \ \lambda'(q,a) = \lambda(\delta(q,a))$$

Constructing Mealy machine for the given Moore machine

Input sequence	a_1	a_2	- - -	a_n
State sequence	q_0	q_1	- - -	q_n
Output sequence	$\lambda'(q_0, a_1)$	$\lambda'(q_1, a_2)$	- - -	$\lambda'(q_{n-1}, a_n)$

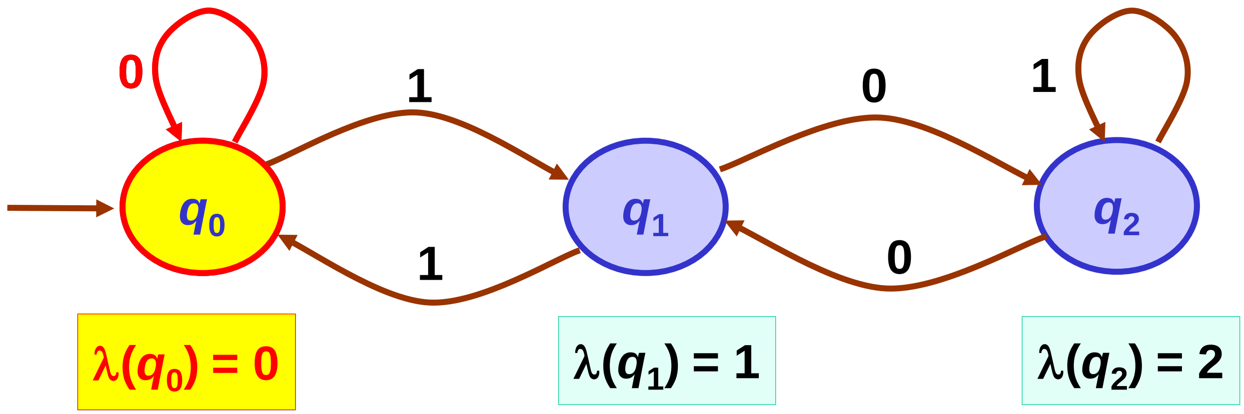
Constructing Mealy machine for the given Moore machine

Input sequence	a_1	a_2	- - -	a_n
State sequence	q_0	q_1	- - -	q_n
Output sequence	$\lambda'(q_0, a_1)$	$\lambda'(q_1, a_2)$	- - -	$\lambda'(q_{n-1}, a_n)$
	$\lambda(\delta(q_0,a_1))$	$\lambda(\delta(q_1,a_2))$	- - -	$\lambda(\delta(q_{n-1},a_n))$

Constructing Mealy machine for the given Moore machine

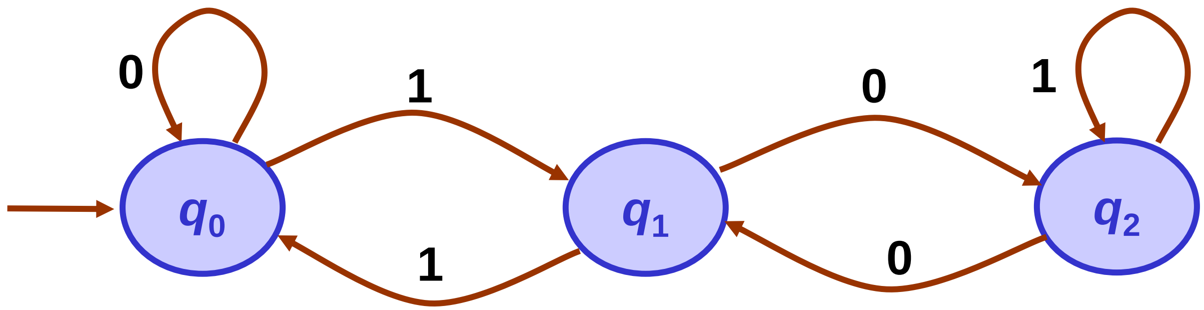
Input sequence	a_1	a_2	- - -	a_n
State sequence	q_0	q_1	- - -	q_n
Output sequence	$\lambda'(q_0, a_1)$	$\lambda'(q_1, a_2)$	- - -	$\lambda'(q_{n-1}, a_n)$
	$\lambda(\delta(q_0,a_1))$	$\lambda(\delta(q_1,a_2))$	- - -	$\lambda(\delta(q_{n-1},a_n))$
	$\lambda(q_1)$	$\lambda(q_2)$	- - -	$\lambda(q_n)$

Constructing Mealy machine for the given Moore machine



$\lambda'(q_0, 0) = 0$, **because** $\lambda'(q_0,0) = \lambda(\delta(q_0,0)) = \lambda(q_0) = 0$

Constructing Mealy machine for the given Moore machine

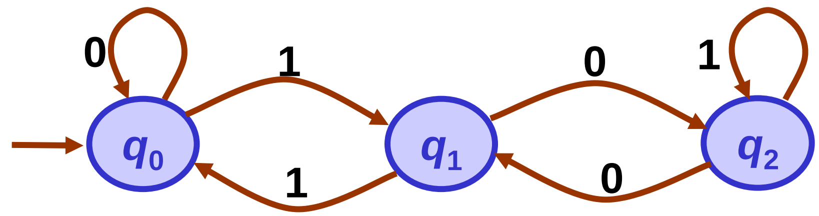


$\lambda(q_0) = 0$ $\lambda(q_1) = 1$ $\lambda(q_2) = 2$

$\lambda'(q_0, 0) = 0,$	**because**	$\lambda'(q_0,0) = \lambda(\delta(q_0,0)) = \lambda(q_0) = 0$
$\lambda'(q_0, 1) = 1,$	**because**	$\lambda'(q_0,1) = \lambda(\delta(q_0,1)) = \lambda(q_1) = 1$
$\lambda'(q_1, 0) = 2,$	**because**	$\lambda'(q_1,0) = \lambda(\delta(q_1,0)) = \lambda(q_2) = 2$
$\lambda'(q_1, 1)=0,$	**because**	$\lambda'(q_1,1) = \lambda(\delta(q_1,1)) = \lambda(q_0) = 0$
$\lambda'(q_2, 0)=1,$	**because**	$\lambda'(q_2,0) = \lambda(\delta(q_2,0)) = \lambda(q_1) = 1$
$\lambda'(q_2, 1)=2,$	**because**	$\lambda'(q_2,1) = \lambda(\delta(q_2,1)) = \lambda(q_2) = 2$

Constructing Mealy machine for the given Moore machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



Read prefix ε

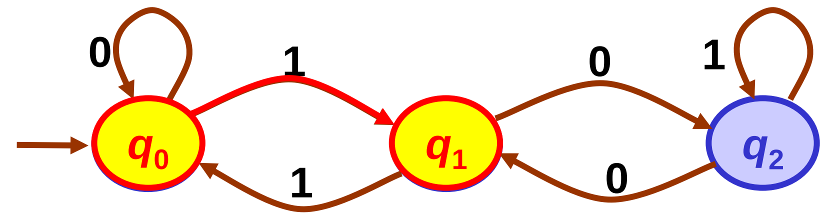
Integer value

q_0

Output sequence

Constructing Mealy machine for the given Moore machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



Read prefix ε **1**

Integer value **1**

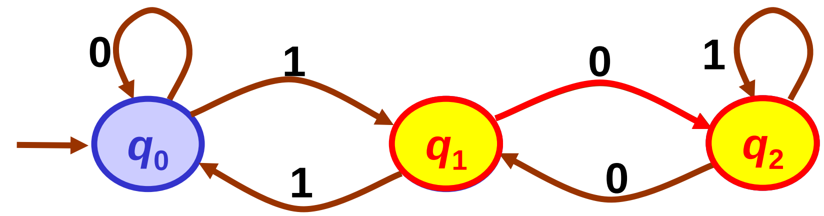
$$q_0 \xrightarrow{1} q_1$$

Output sequence

$\lambda'(q_0,1) = 1$

Constructing Mealy machine for the given Moore machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



Read prefix	ε	1	10
Integer value		1	2

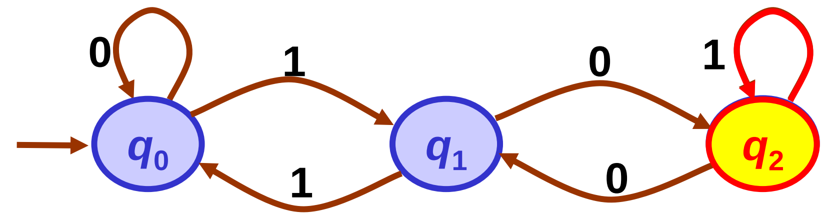
$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2$$

Output sequence

$\lambda'(q_0,1) = 1$ $\lambda'(q_1,0) = 2$

Constructing Mealy machine for the given Moore machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



Read prefix	ε	**1**	**10**	**101**
Integer value		**1**	**2**	**5**

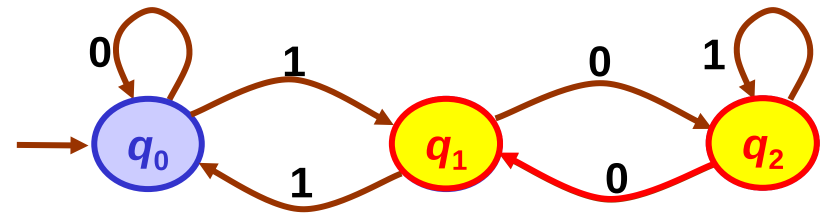
$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_2$$

Output sequence

$\lambda'(q_0,1) = 1$ $\quad$ $\lambda'(q_1,0) = 2$ $\quad$ $\lambda'(q_2,1) = 2$

Constructing Mealy machine for the given Moore machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



Read prefix	ε	1	10	101	1010
Integer value		1	2	5	10

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1$$

Output sequence

$\lambda'(q_0,1) = 1$ $\lambda'(q_1,0) = 2$ $\lambda'(q_2,1) = 2$ $\lambda'(q_2,0) = 1$

Constructing Moore machine for the given Mealy machine

Moore machine	Mealy machine
$M = (Q', \Sigma, \Delta, \delta', \lambda', q_0')$	$M' = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

1) $Q' = Q \times \Delta$, where state $[q,b] \in Q'$, $q \in Q$ i $b \in \Delta$

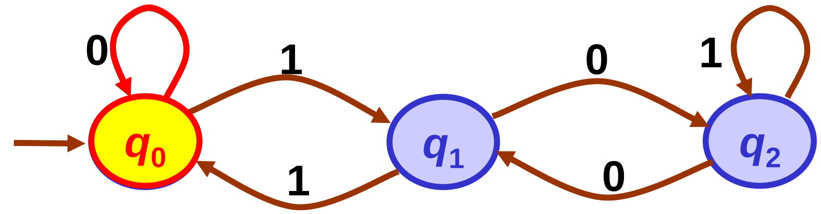
2) $q_0' = [q_0, b_0]$, where b_0 is an arbitrary element of the set Δ

3) $\delta'([q,b], a) = [\delta(q,a), \lambda(q,a)]$, where $q \in Q$, $b \in \Delta$ and $a \in \Sigma$

4) $\lambda'([q,b]) = b$, where $q \in Q$ i $b \in \Delta$

Constructing Moore machine for the given Mealy machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



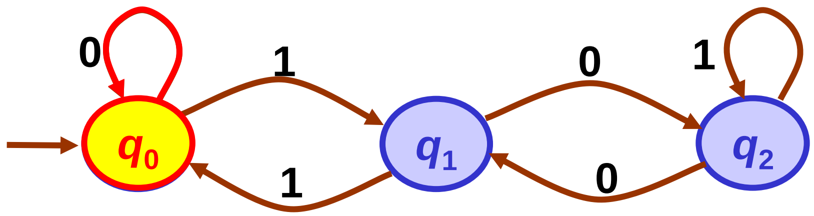
1) $Q' = \{[q_0,0], [q_0,1], [q_0,2], [q_1,0], [q_1,1], [q_1,2], [q_2,0], [q_2,1],[q_2,2]\}$

2) $q_0' = [q_0,0]$

3) $\delta'([q_0,0], 0) = [\ \delta(q_0,0), \lambda(q_0,0)\ ] = [q_0,$

Constructing Moore machine for the given Mealy machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



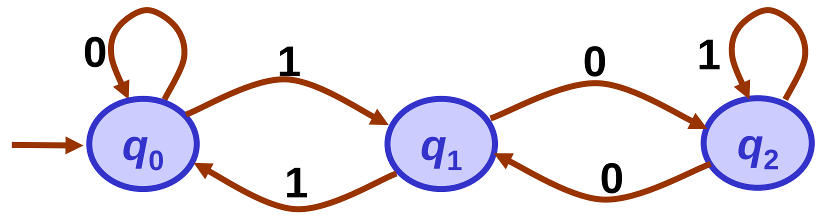
1) $Q' = \{[q_0,0], [q_0,1], [q_0,2], [q_1,0], [q_1,1], [q_1,2], [q_2,0], [q_2,1], [q_2,2]\}$

2) $q_0' = [q_0,0]$

3) $\delta'([q_0,0], 0) = [\ \delta(q_0,0), \lambda(q_0,0)\ ] = [q_0, 0]$

Constructing Moore machine for the given Mealy machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



1) $Q' = \{[q_0,0], [q_0,1], [q_0,2], [q_1,0], [q_1,1], [q_1,2], [q_2,0], [q_2,1],[q_2,2]\}$

2) $q_0' = [q_0,0]$

3) $\delta'([q_0,0], 0) = [\ \delta(q_0,0), \lambda(q_0,0)\ ] = [q_0, 0]$

$\delta'([q_0,0], 1) = [\ \delta(q_0,1), \lambda(q_0,1)\ ] = [q_1, 1]$

$\delta'([q_1,1], 0) = [\ \delta(q_1,0), \lambda(q_1,0)\ ] = [q_2, 2]$

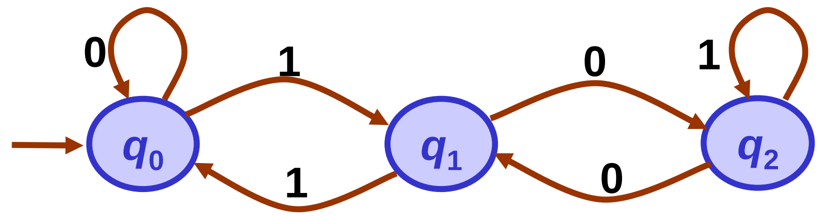
$\delta'([q_1,1], 1) = [\ \delta(q_1,1), \lambda(q_1,1)\ ] = [q_0, 0]$

$\delta'([q_2,2], 0) = [\ \delta(q_2,0), \lambda(q_2,0)\ ] = [q_1, 1]$

$\delta'([q_2,2], 1) = [\ \delta(q_2,1), \lambda(q_2,1)\ ] = [q_2, 2]$

Constructing Moore machine for the given Mealy machine

$\lambda'(q_0, 0) = 0$	$\lambda'(q_1, 0) = 2$	$\lambda'(q_2, 0) = 1$
$\lambda'(q_0, 1) = 1$	$\lambda'(q_1, 1) = 0$	$\lambda'(q_2, 1) = 2$



1) $Q' = \{[q_0,0], [q_0,1], [q_0,2], [q_1,0], [q_1,1], [q_1,2], [q_2,0], [q_2,1], [q_2,2]\}$

2) $q_0' = [q_0,0]$

3) $\delta'([q_0,0], 0) = [\ \delta(q_0,0), \lambda(q_0,0)\ ] = [q_0, 0]$

$\delta'([q_0,0], 1) = [\ \delta(q_0,1), \lambda(q_0,1)\ ] = [q_1, 1]$

$\delta'([q_1,1], 0) = [\ \delta(q_1,0), \lambda(q_1,0)\ ] = [q_2, 2]$

$\delta'([q_1,1], 1) = [\ \delta(q_1,1), \lambda(q_1,1)\ ] = [q_0, 0]$

$\delta'([q_2,2], 0) = [\ \delta(q_2,0), \lambda(q_2,0)\ ] = [q_1, 1]$

$\delta'([q_2,2], 1) = [\ \delta(q_2,1), \lambda(q_2,1)\ ] = [q_2, 2]$

4)

$\lambda'([q_0, 0]) = 0$	$\lambda'([q_1, 0]) = 0$	$\lambda'([q_2, 0]) = 0$
$\lambda'([q_0, 1]) = 1$	$\lambda'([q_1, 1]) = 1$	$\lambda'([q_2, 1]) = 1$
$\lambda'([q_0, 2]) = 2$	$\lambda'([q_1, 2]) = 2$	$\lambda'([q_2, 2]) = 2$

Lecture overview

Regular expressions

$$N = \{\ wcw^R\ \}$$

Regular expressions

$$N = \{\ wcw^R\ \}$$

***N* is not a regular language**

It is not possible to construct a finite state machine accepting *N*

Regular expressions

Regular language ***K***

→

Definition of language K by regular expressions r
$L(r) = K$

↓

ε–NFA ***M such that***
$L(M)=L(r)$

↓

NFA ***M' such that***
$L(M')=L(M)$

↓

DFA ***M'' such that***
$L(M'')=L(M')$

↓

DFA ***M''' with minimum number of states such that***
$L(M''')=L(M'')=L(M')=L(M)=L(r)=K$

Definition of regular expressions

1)	$\emptyset$	- Language $L(\emptyset) = \{ \}$
2)	ε	- Language $L(\varepsilon) = \{ \varepsilon \}$
3)	a	- Language $L(a) = \{ a \}$
4)	$(r)+(s)$ $(r) \mid (s)$	- Language $L((r)+(s)) = L(r) \cup L(s)$
5)	$(r)(s)$	- Language $L((r)(s)) = L(r)L(s)$
6)	$(r)^*$	- Language $L((r)^*) = L(r)^*$

Examples of regular expressions and languages

1) **Regular expression:** 01
   Language: $L(01) = \{01\}$

2) **Regular expression :** $0+1$
   Language : $L(0+1) = \{0, 1\}$

3) **Regular expression :** $(0+1)(0+1)$
   Language : $L((0+1)(0+1)) = \{00, 01, 10, 11\}$

4) **Regular expression :** 1^*
   Language : $L(1^*) = \{\varepsilon, 1, 11, 111, ..., 1111111, ...\}$

Examples of regular expressions and languages

5) **Regular expression :** $(0+1)^*$

Language : $L(0+1)^*=\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, ..., 01111101, ...\}$

6) **Regular expression :** $(0+1)^*00(0+1)^*$

Language : Any string having at least two consecutive 0s at any position.

7) **Regular expression :** 0^*1^*

Language : $L(0^*1^*)=\{\varepsilon, 0, 1, 00, 01, 000, 001, 011, 111, ..., 000111111, ...\}$

Operator associativity and precedence

1) *

Unary operator
Left associative
Highest precedence

2) Concatenation operator

Left associative
Higher precendence than +

3) +

Left associative
Lowest precedence

Algebraic laws

$r+s = s+r$	+ *is commutative*
$r+(s+t) = (r+s)+t$	+ *is associative*
$(rs)t = r(st)$	*concatenation is associative*
$r(s+t) = rs+rt$ $(s+t)r = sr+tr$	*distributivity of concatenation over* +
$\varepsilon r = r\varepsilon = r$	ε is *neutral element for concatenation*
$r^* = (r+\varepsilon)^*$	*relation between* + and *
$r^{**} = r^*$	*idempotence*

Lecture overview

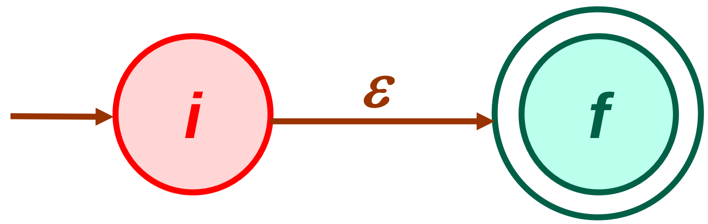
Construction of ε–NFA for the given regular expressions

***p*1)** $\emptyset$ **- Language** $L(\emptyset) = \{\ \}$

ε**-NFA** $M=(\{i, f\}, \Sigma, \{\ \}, i, \{f\})$

Construction of ε–NFA for the given regular expressions

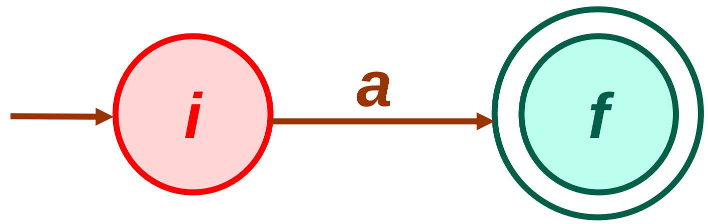
***p*2)** ε **- Language** $L(\varepsilon) = \{ \varepsilon \}$



ε**-NFA** $M=(\{ i, f \}, \Sigma, \{ \delta(i, \varepsilon)=f \}, i, \{ f \})$

Construction of ε–NFA for the given regular expressions

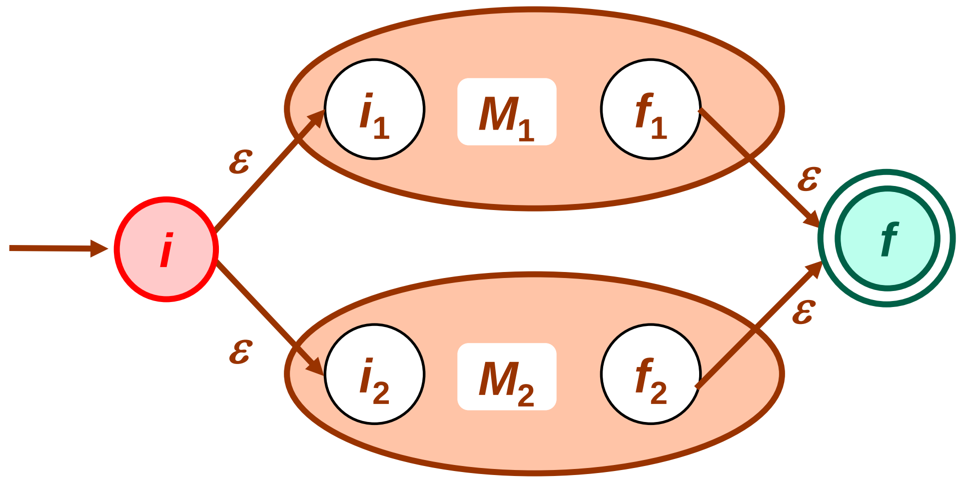
p*3)** ***a **- Language $L(a) = \{ a \}$**



ε-NFA M=({ i, f } , Σ, { $\delta(i, a)=f$ }, i , { f })

Construction of ε–NFA for the given regular expressions

***p*4)** $(r)+(s)$ **- Language** $L((r)+(s)) = L(r) \cup L(s)$

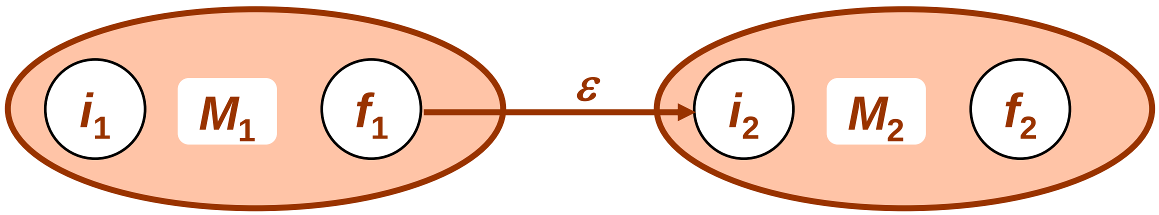


ε–NFA $M=(Q_1 \cup Q_2 \cup \{i, f\}, \Sigma_1 \cup \Sigma_2, \delta, i, \{f\})$

***a*)** $\delta(i, \varepsilon) = \{i_1, i_2\}$
***b*)** $\delta(q, a) = \delta_1(q, a), \forall q \in Q_1, \forall a \in (\Sigma_1 \cup \{\varepsilon\})$
***c*)** $\delta(q, b) = \delta_2(q, b), \forall q \in Q_2, \forall b \in (\Sigma_2 \cup \{\varepsilon\})$
***d*)** $\delta(f_1, \varepsilon) = \delta(f_2, \varepsilon) = \{f\}$

Construction of ε–NFA for the given regular expressions

***p*5)** $(r)(s)$ - **Language** $L((r)(s)) = L(r)L(s)$



$$\varepsilon\text{–NFA } M=(Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, i_1, \{f_2\})$$

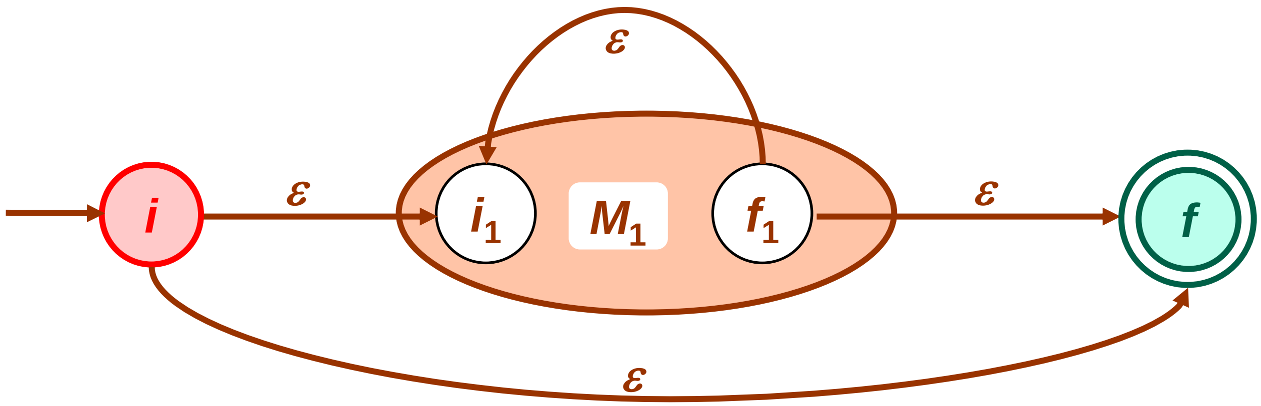
a) $\delta(q, a) = \delta_1(q, a), \forall q \in Q_1, \forall a \in (\Sigma_1 \cup \{\varepsilon\})$

b) $\delta(q, b) = \delta_2(q, b), \forall q \in Q_2, \forall b \in (\Sigma_2 \cup \{\varepsilon\})$

c) $\delta(f_1, \varepsilon) = i_2$

Construction of ε–NFA for the given regular expressions

***p*6)** $(r)^*$ **- Language** $L((r)^*) = L(r)^*$



ε–NFA $M=(Q_1 \cup Q_2 \cup \{i, f\}, \Sigma_1 \cup \Sigma_2, \delta, i, \{f\})$

***a*)** $\delta(i, \varepsilon) = \delta(f_1, \varepsilon) = \{i_1, f\}$

***b*)** $\delta(q, a) = \delta_1(q, a), \forall q \in Q_1, \forall a \in (\Sigma_1 \cup \{\varepsilon\})$

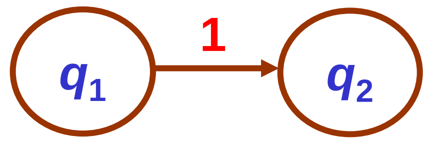
Construction of ε–NFA for the given regular expressions

$$r = 01^*+1$$

Construction of ε–NFA for the given regular expressions

$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

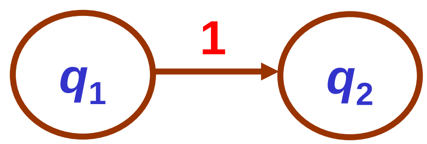


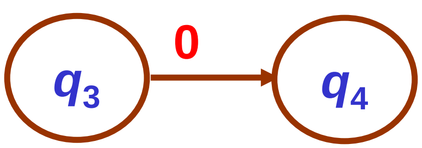
Construction of ε–NFA for the given regular expressions

$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

$$r_1 = r_3 r_4,\ \ r_3 = 0,\ r_4 = 1^*$$





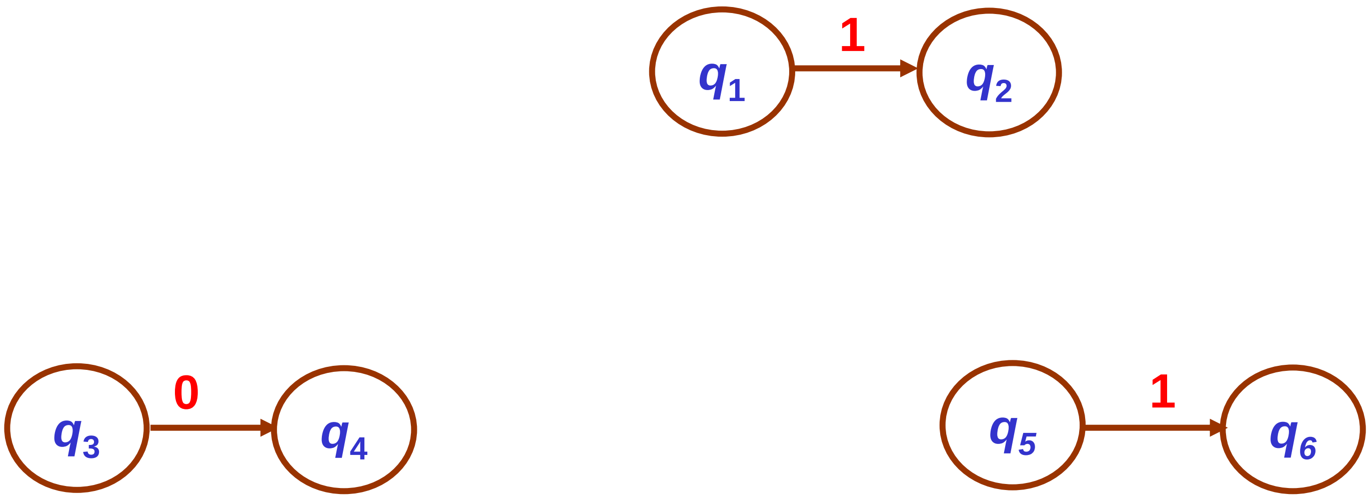
Construction of ε–NFA for the given regular expressions

$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

$$r_1 = r_3 r_4,\ \ r_3 = 0,\ r_4 = 1^*$$

$$r_4 = r_5^*,\ r_5 = 1$$



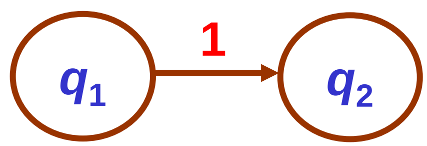
Construction of ε–NFA for the given regular expressions

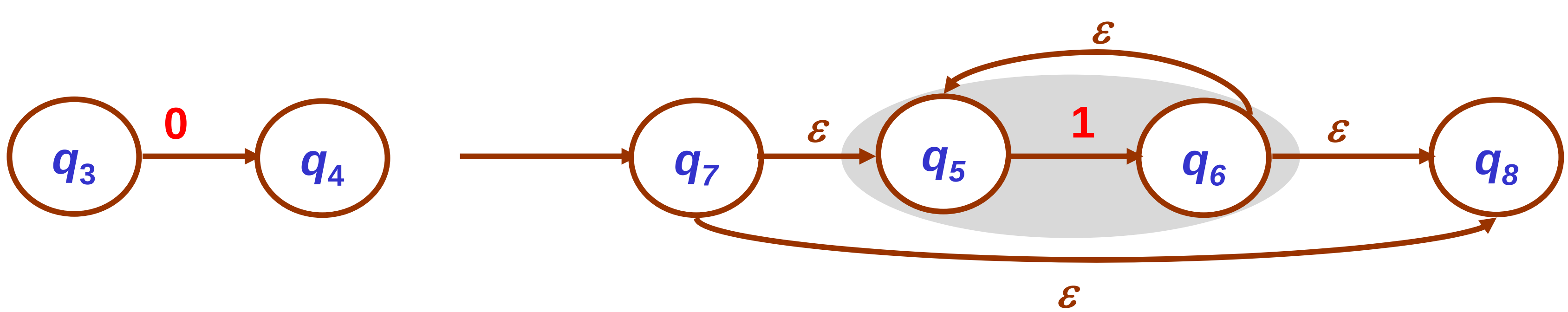
$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

$$r_1 = r_3 r_4,\ \ r_3 = 0,\ r_4 = 1^*$$

$$r_4 = r_5^*,\ r_5 = 1$$





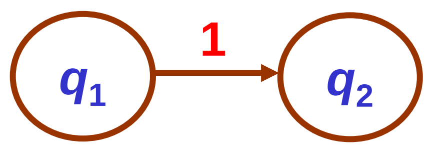
Construction of ε–NFA for the given regular expressions

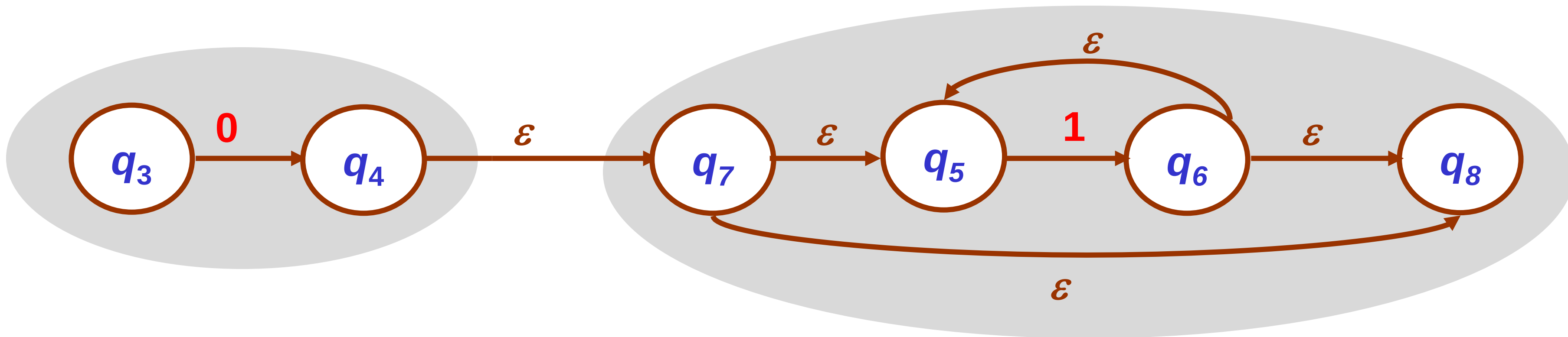
$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

$$r_1 = r_3 r_4,\ \ r_3 = 0,\ r_4 = 1^*$$

$$r_4 = r_5^*,\ r_5 = 1$$





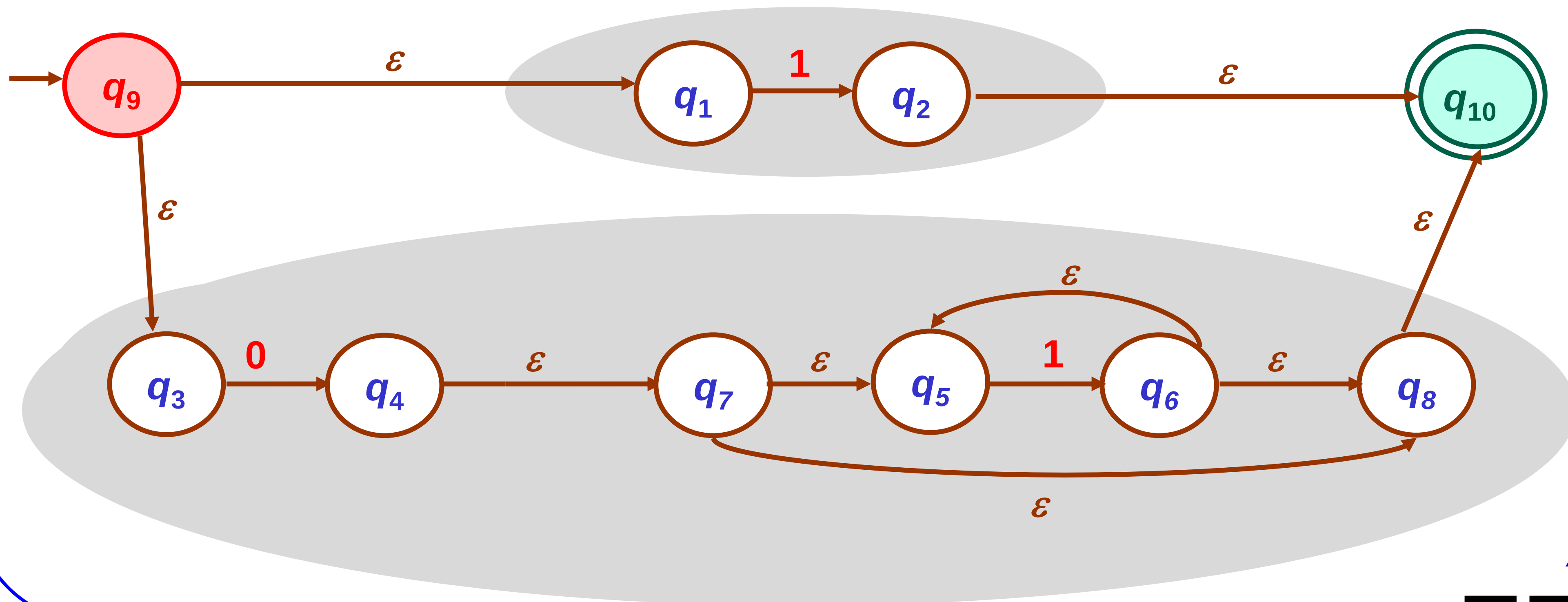
Construction of ε–NFA for the given regular expressions

$$r = 01^*+1$$

$$r = r_1+r_2,\ r_1 = 01^*,\ r_2 = 1$$

$$r_1 = r_3 r_4,\ \ r_3 = 0,\ r_4 = 1^*$$

$$r_4 = r_5^*,\ r_5 = 1$$



Construction of ε–NFA for the given regular expressions

- The number of states in the constructed ε–NKA is never larger than $2|r|$, where $|r|$ is the number of symbols in regular expression r.

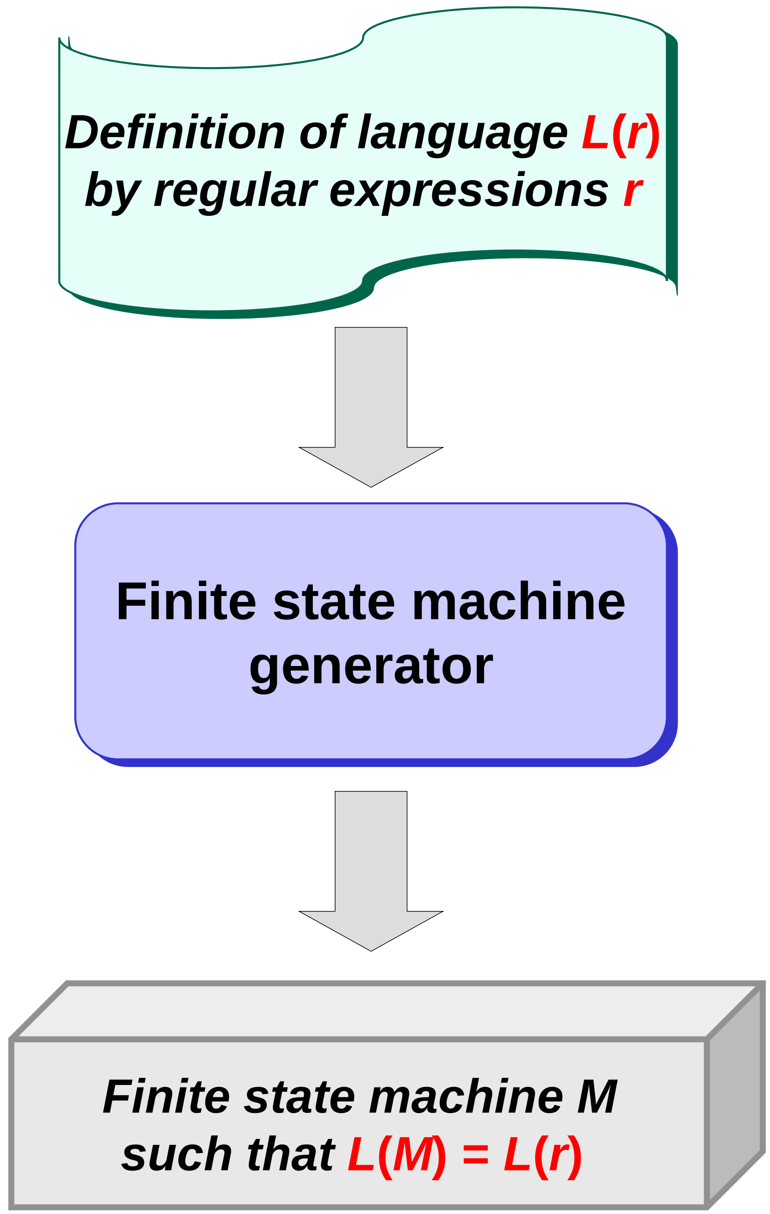
- ε–NKA has only one accepting state f for which

  $$\delta(f, a) = \varnothing$$

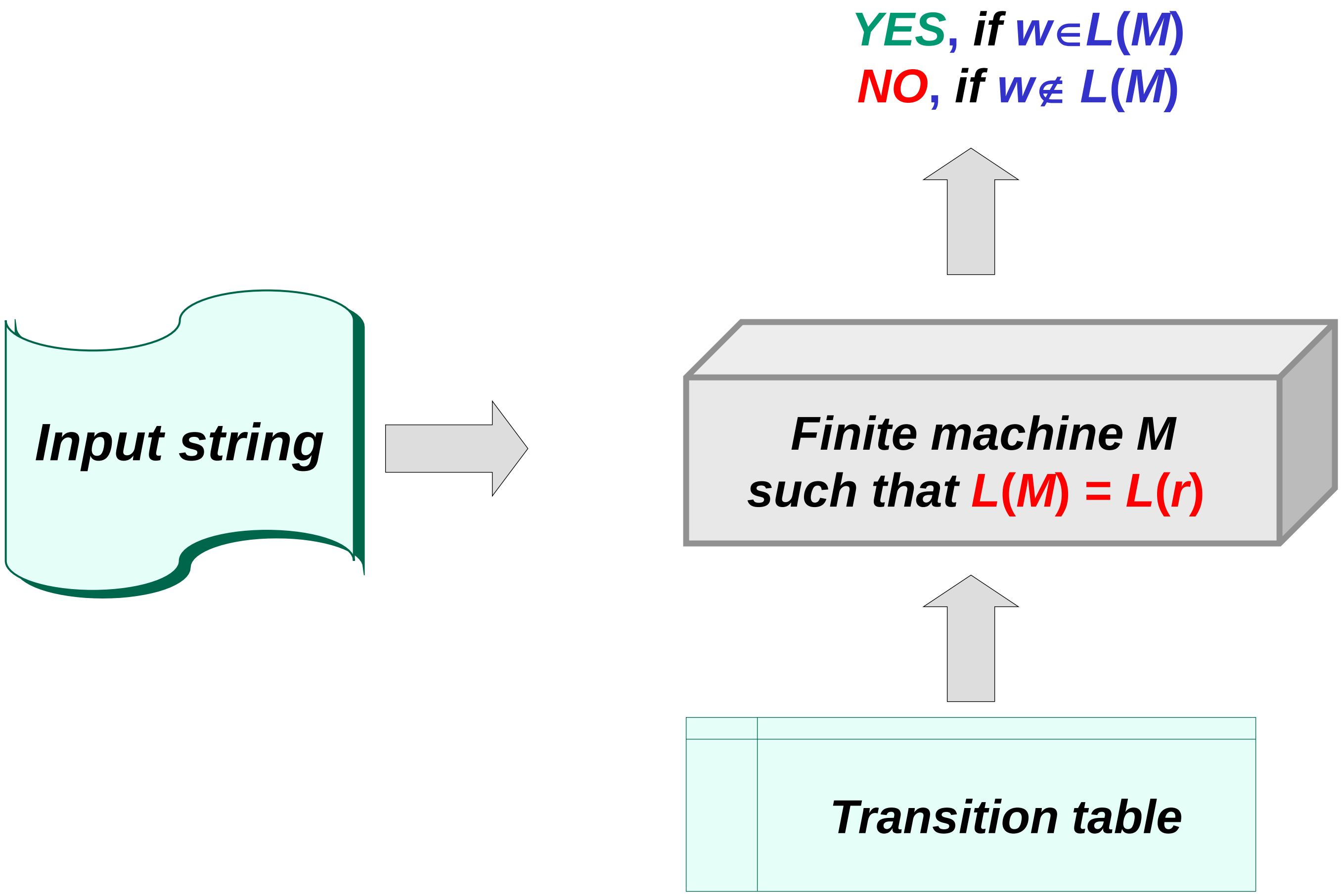
- Set $\delta(q, a)$ contains at most one state for each input symbol a from alphabet Σ, whereas set $\delta(q, \varepsilon)$ contains at most two states.

Lecture overview

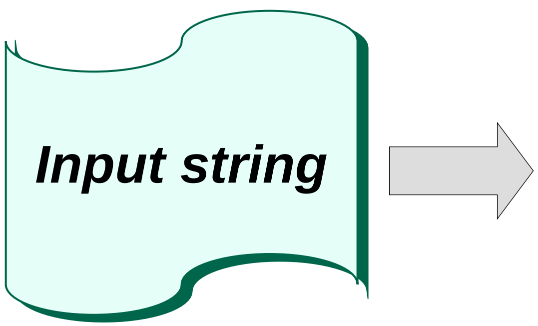
Finite state machine generator



Finite state machine generator

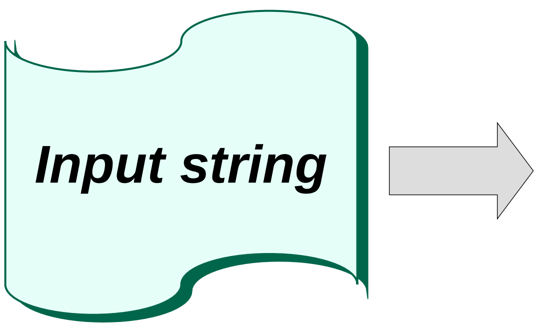


Finite state machine generator




```
Table[PP, 0] = NP;
Table[PP, 1] = PN;
Table[PP, ⊥] = 1;
Table[NP, 0] = PP;
Table[NP, 1] = NN;
Table[NP, ⊥] = 0;
Table[PN, 0] = NN;
Table[PN, 1] = PP;
Table[PN, ⊥] = 0;
Table[NN, 0] = PN;
Table[NN, 1] = NP;
Table[NN, ⊥] = 1;

State = PP;

Read(Symbol);

while (Symbol != ⊥)
{
    State = Table[State, Symbol];
    Read(Symbol);
}

Print(Table[State, ⊥], State);
```

Finite state machine generator

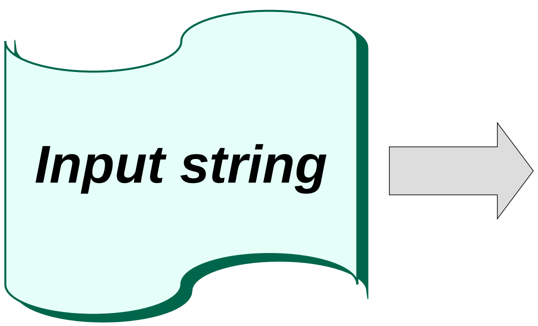



```
Table[PP, 0] = NP;
Table[PP, 1] = PN;
Table[PP, ⊥] = 1;
Table[NP, 0] = PP;
Table[NP, 1] = NN;
Table[NP, ⊥] = 0;
Table[PN, 0] = NN;
Table[PN, 1] = PP;
Table[PN, ⊥] = 0;
Table[NN, 0] = PN;
Table[NN, 1] = NP;
Table[NN, ⊥] = 1;

State = PP;

Read(Symbol);

while (Symbol != ⊥)
{
    State = Table[State, Symbol];
    Read(Symbol);
}

Print(Table[State, ⊥], State);
```

TRANSITION TABLE

contains variable information that depend on the transition function of the finite state machine

Finite state machine generator




```
Table[PP, 0] = NP;
Table[PP, 1] = PN;
Table[PP, ⊥] = 1;
Table[NP, 0] = PP;
Table[NP, 1] = NN;
Table[NP, ⊥] = 0;
Table[PN, 0] = NN;
Table[PN, 1] = PP;
Table[PN, ⊥] = 0;
Table[NN, 0] = PN;
Table[NN, 1] = NP;
Table[NN, ⊥] = 1;

State = PP;
```

TRANSITION TABLE

contains variable information that depend on the transition function of the finite state machine

```
Read(Symbol);

while (Symbol != ⊥)
{
    State = Table[State, Symbol];
    Read(Symbol);
}

Print(Table[State, ⊥], State);
```

SIMULATOR PROGRAM

the program code is the same for all finite state machines

Finite state machine generator