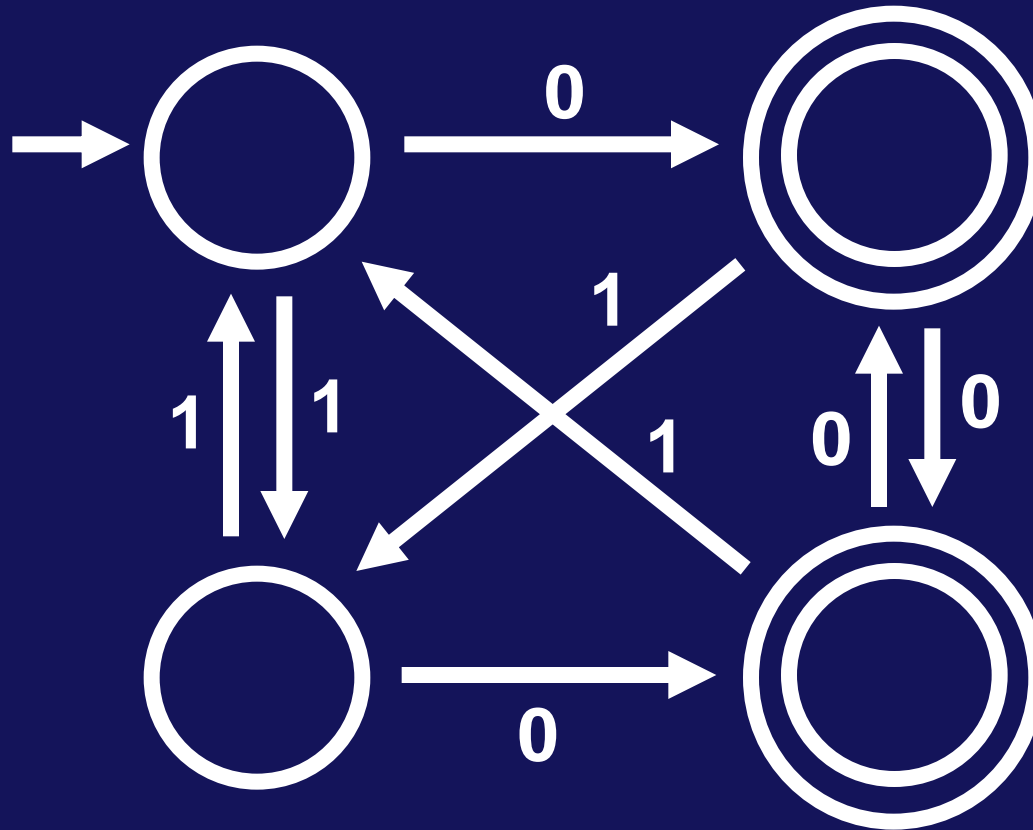


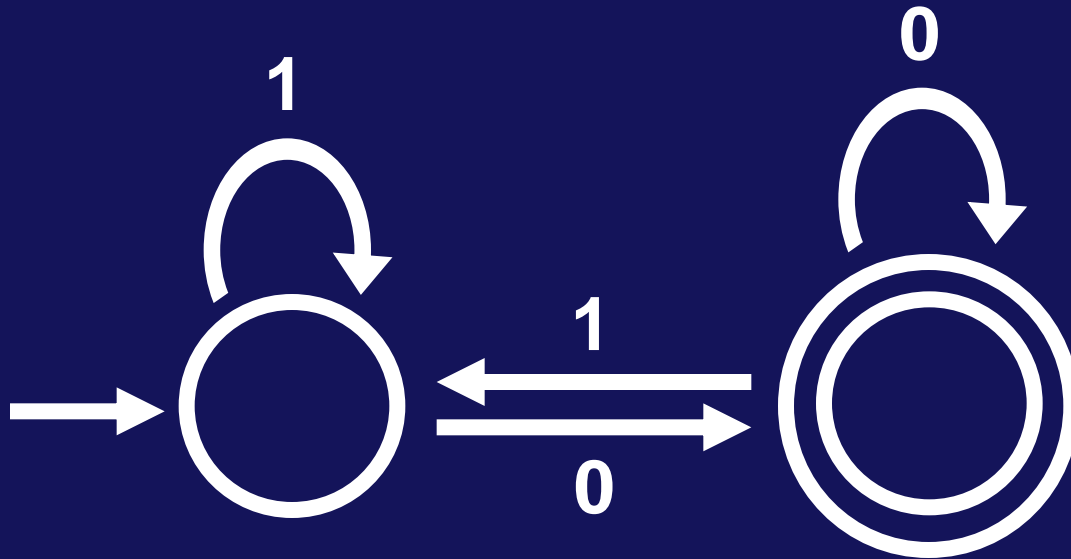
# MINIMIZING DFA<sub>s</sub>

# IS THIS MINIMAL?

## NO



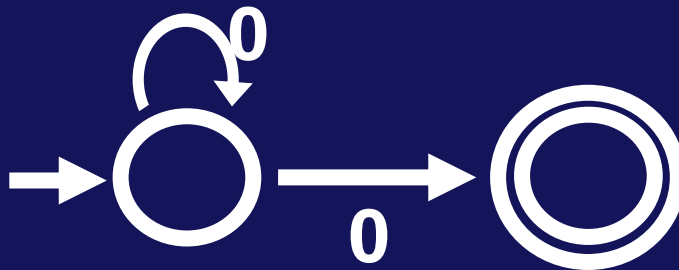
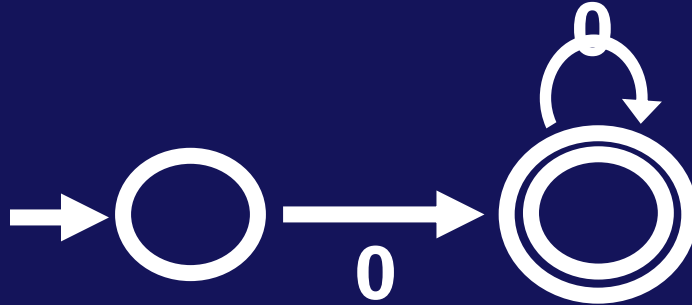
# IS THIS MINIMAL?



# THEOREM

For every regular language  $L$ , there exists  
a **unique** (up to re-labeling of the states)  
minimal DFA  $M$  such that  $L = L(M)$

# NOT TRUE FOR NFAs



Because of this, minimization of NFA is complicated and is out of scope of current ToC course.

# EXTENDING $\delta$

Given DFA  $M = (Q, \Sigma, \delta, q_0, F)$  extend  $\delta$  to  $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$  as follows:

$$\hat{\delta}(q, \epsilon) = q$$

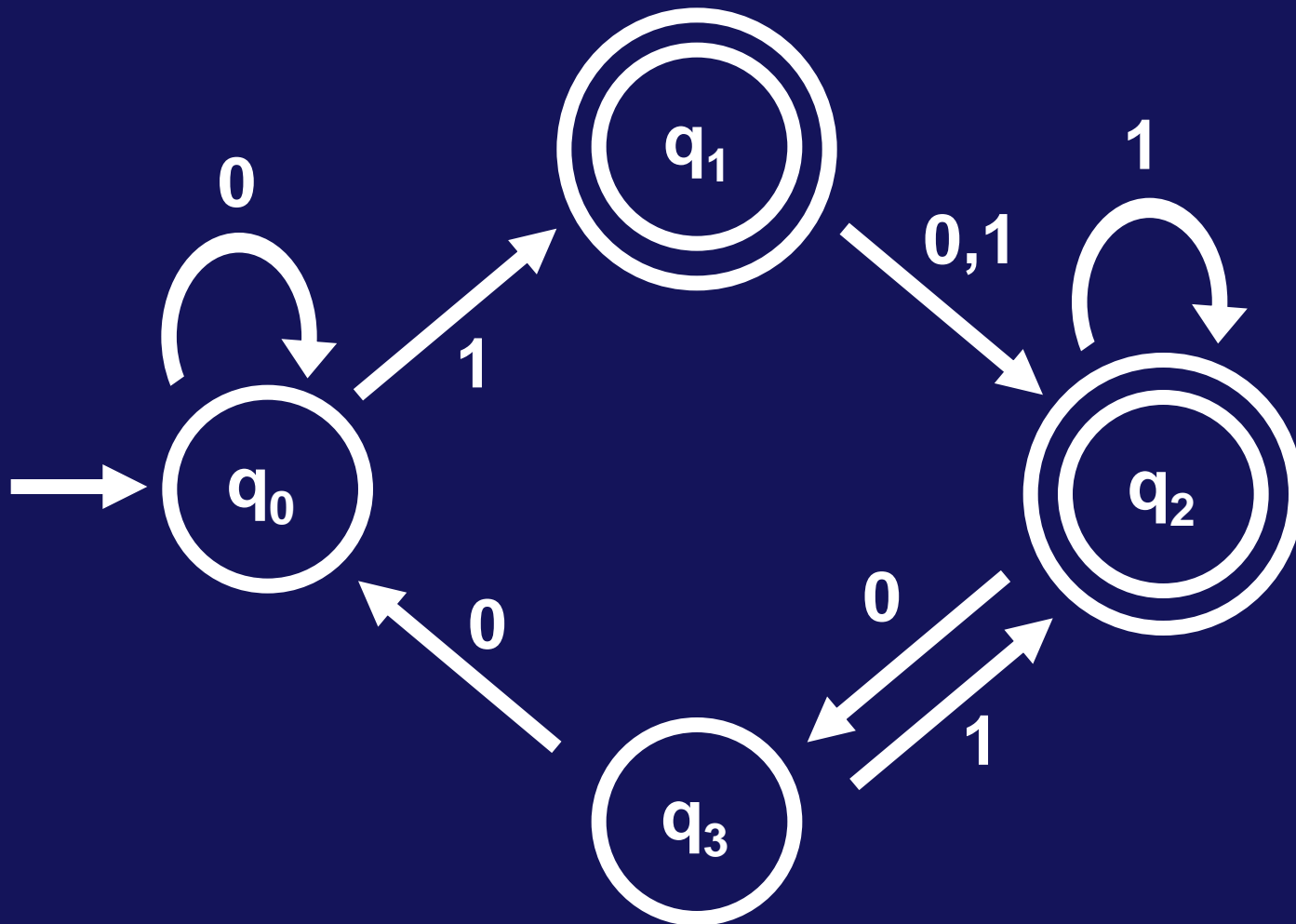
$$\hat{\delta}(q, a) = \delta(q, a) \text{ where } a \in \Sigma$$

$$\hat{\delta}(q, w_1 \dots w_{k+1}) = \delta(\hat{\delta}(q, w_1 \dots w_k), w_{k+1})$$

Note: in  $\hat{\delta}(q, a)$ ,  $a$  is a string. Context should clear this.

A string  $w \in \Sigma^*$  **distinguishes states**  $q_1$  from  $q_2$  if

$$\delta(q_1, w) \in F \Leftrightarrow \delta(q_2, w) \notin F$$



$\varepsilon$  distinguishes accept from non-accept states



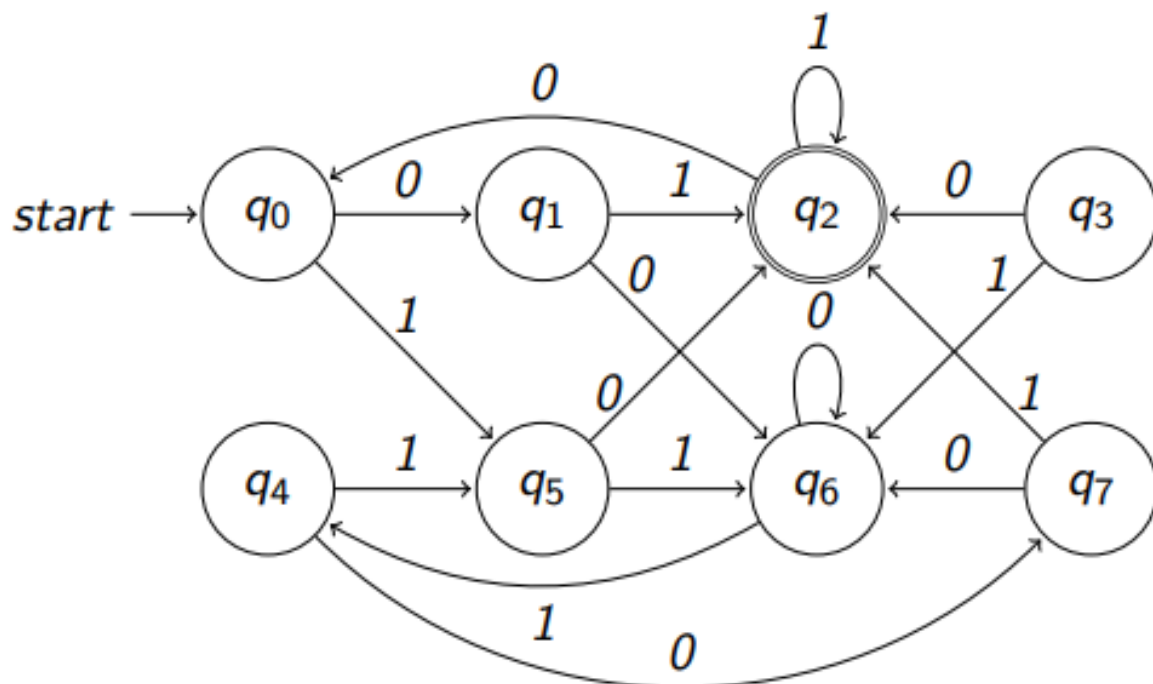
Let  $M = (Q, \Sigma, \delta, q_0, F)$  and let  $p, q, r \in Q$

**Definition** :

(1)  $p$  is **equivalent** to  $q$  iff there is *no*  $w \in \Sigma^*$  that distinguishes  $p$  and  $q$ ,

(2) Otherwise  $p$  is **not equivalent** to  $q$ . In this case, we say  $p$  and  $q$  are **distinguishable**.

## Example: distinguishable states



- ▶  $\epsilon$  distinguishes  $q_2$  and  $q_6$ .
- ▶ 01 distinguishes  $q_0$  and  $q_6$ .

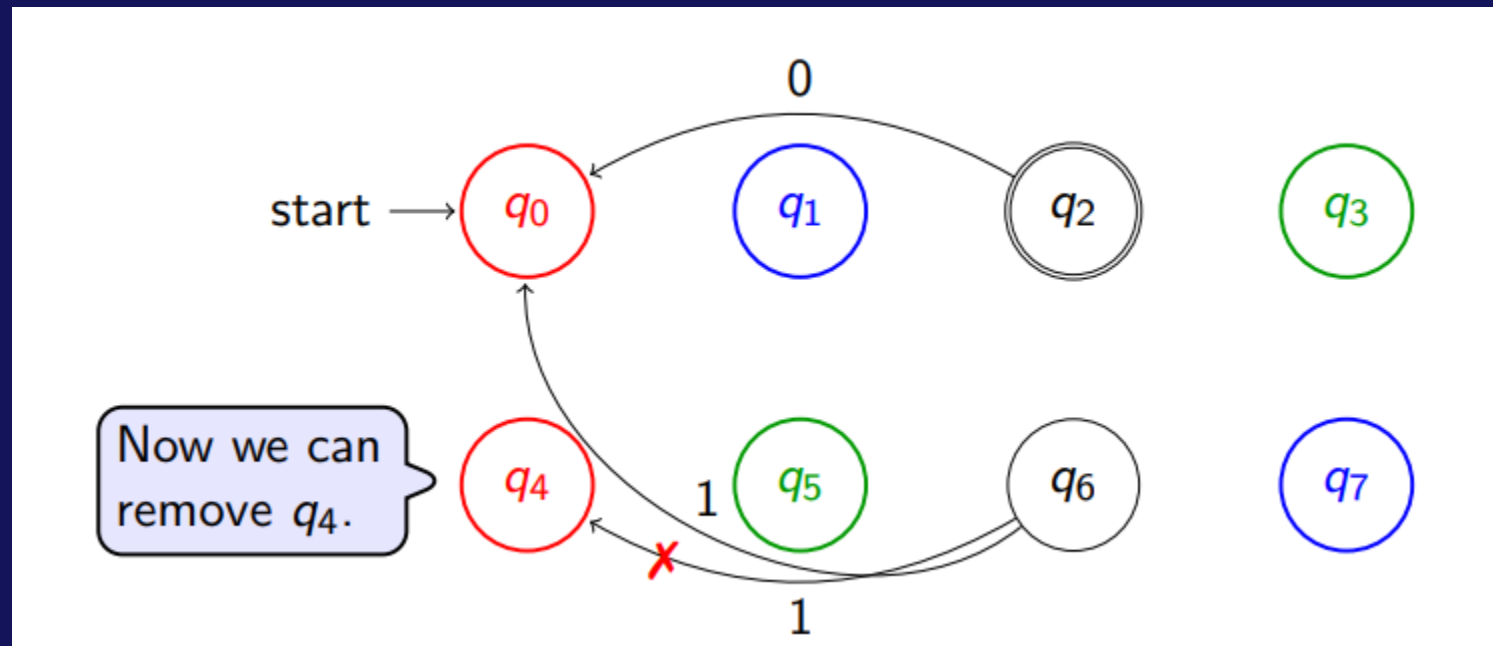
### Exercise

Give strings that distinguish the following pair of states.

- ▶  $q_1$  and  $q_5$
- ▶  $q_4$  and  $q_3$
- ▶  $q_2$  and  $q_7$
- ▶  $q_1$  and  $q_7$

# DFA Minimization, intuition

- We can remove unreachable states (**why and how to find unreachable states?**)
- If states  $q_0$  and  $q_4$  are equivalent, then, we can move all incoming transitions (arrows) from  $q_4$  to  $q_0$
- Because of this  $q_4$  becomes unreachable, hence can be removed.



- But, how is that you know these two are equivalent states?

Let  $M = (Q, \Sigma, \delta, q_0, F)$  and let  $p, q, r \in Q$

Define the relation “ $\sim$ ”:

$p \sim q$  iff  $p$  is **equivalent** to  $q$

$p \not\sim q$  iff  $p$  is distinguishable from  $q$

**Proposition:** “ $\sim$ ” is an **equivalence relation**

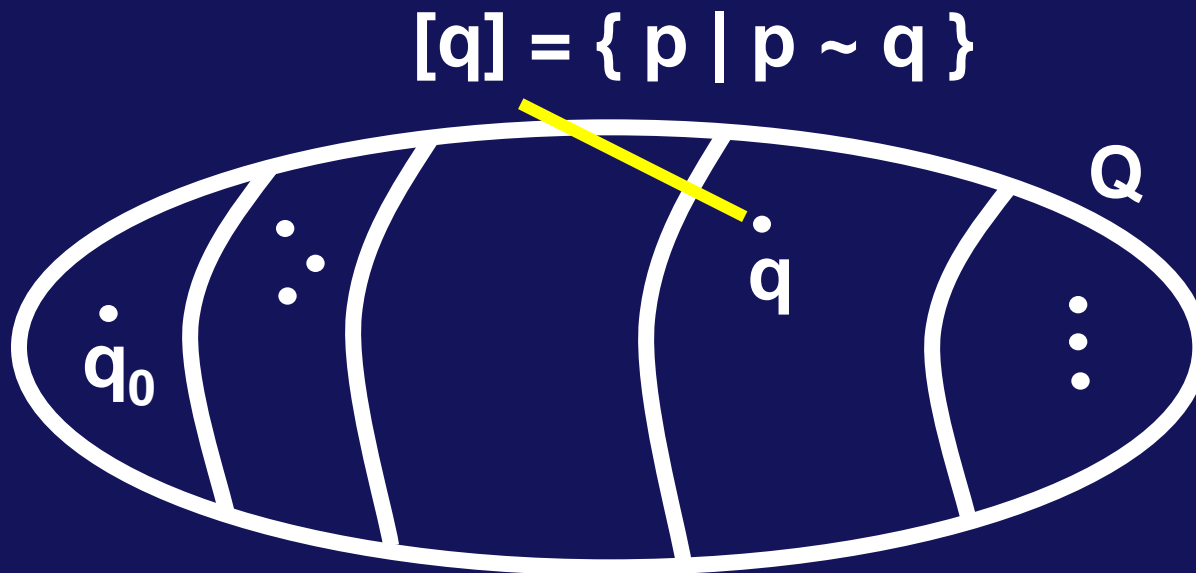
$p \sim p$  (reflexive)

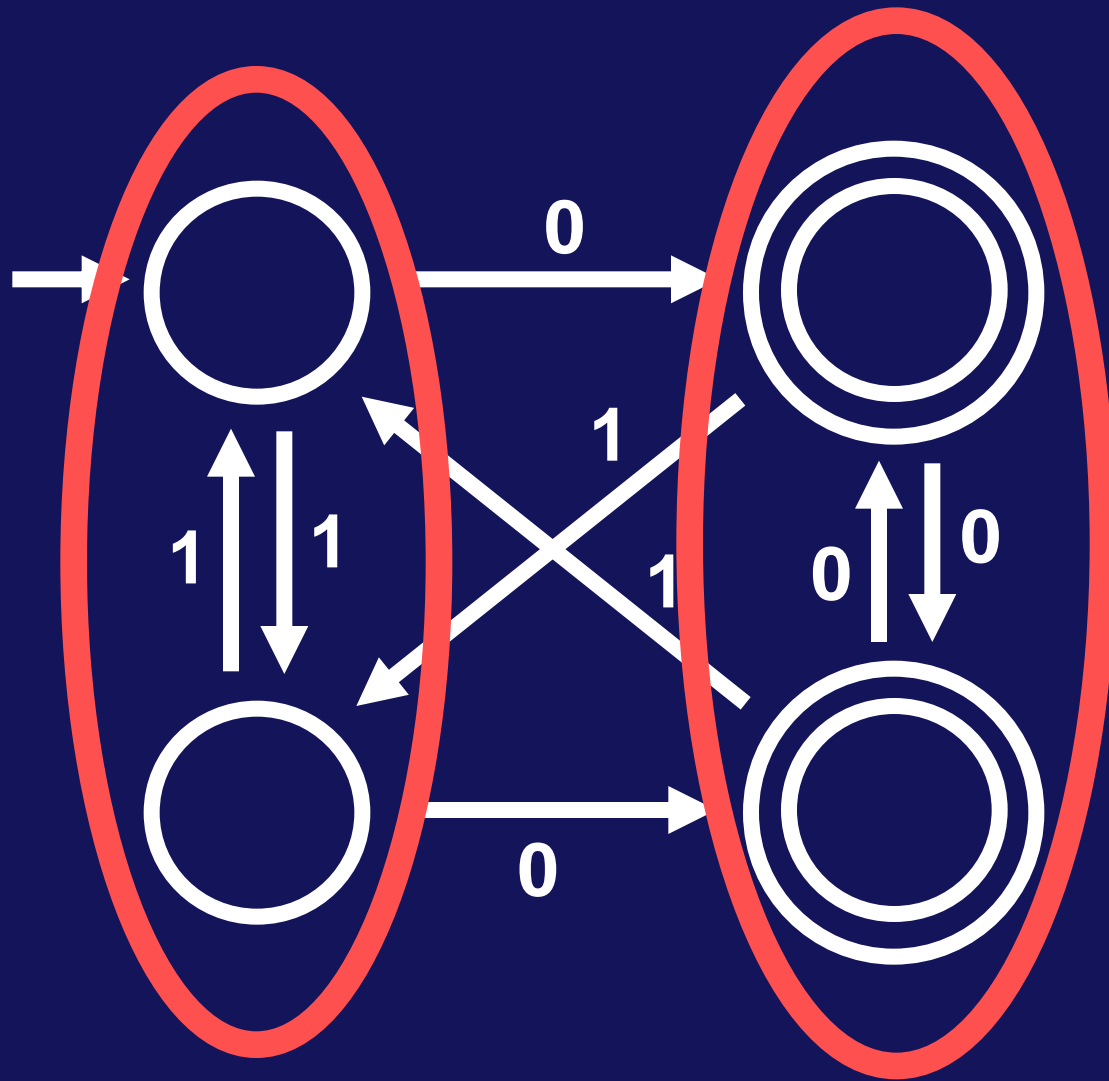
$p \sim q \Rightarrow q \sim p$  (symmetric)

$p \sim q$  and  $q \sim r \Rightarrow p \sim r$  (transitive)

Let  $M = (Q, \Sigma, \delta, q_0, F)$  and let  $p, q, r \in Q$

**Proposition:** “ $\sim$ ” is an **equivalence relation**  
so “ $\sim$ ” partitions the set of states of  $M$  into  
disjoint equivalence classes





## Algorithm MINIMIZE(DFA $M$ )

---

Input: DFA  $M$

Output: DFA  $M_{MIN}$  such that:

$$M \equiv M_{MIN}$$

$M_{MIN}$  has no inaccessible states

$M_{MIN}$  is **irreducible**

||

states of  $M_{MIN}$  are pairwise distinguishable

**Theorem:  $M_{MIN}$  is the unique minimum**



## Algorithm MINIMIZE(DFA $M$ )

---

(1) Remove all inaccessible states from  $M$

(2) Apply Table-Filling algorithm to get  
 $E_M = \{ [q] \mid q \text{ is an accessible state of } M \}$

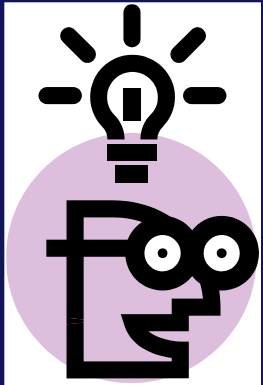
$$M_{MIN} = (Q_{MIN}, \Sigma, \delta_{MIN}, q_{0\ MIN}, F_{MIN})$$

$$Q_{MIN} = E_M, \quad q_{0\ MIN} = [q_0], \quad F_{MIN} = \{ [q] \mid q \in F \}$$

$$\delta_{MIN}([q], a) = [ \delta(q, a) ]$$

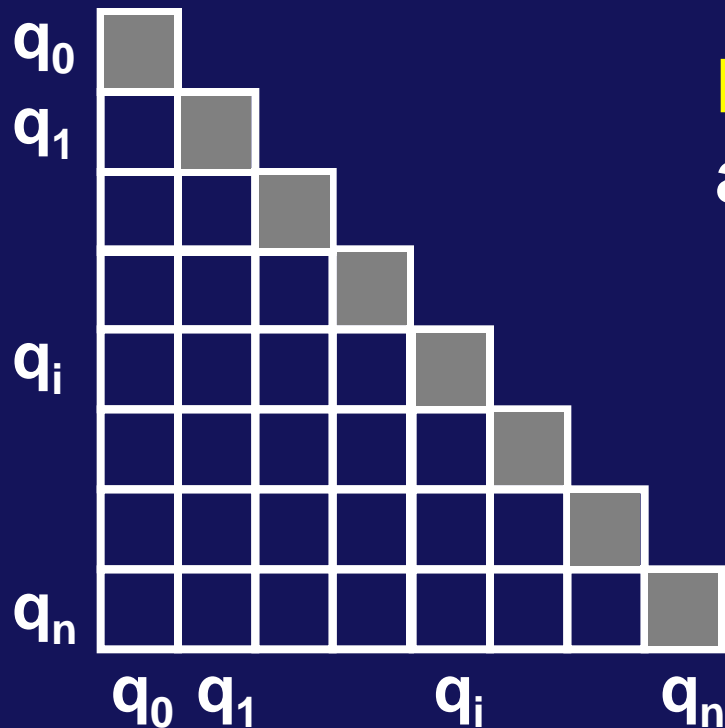
# TABLE-FILLING ALGORITHM

## IDEA!



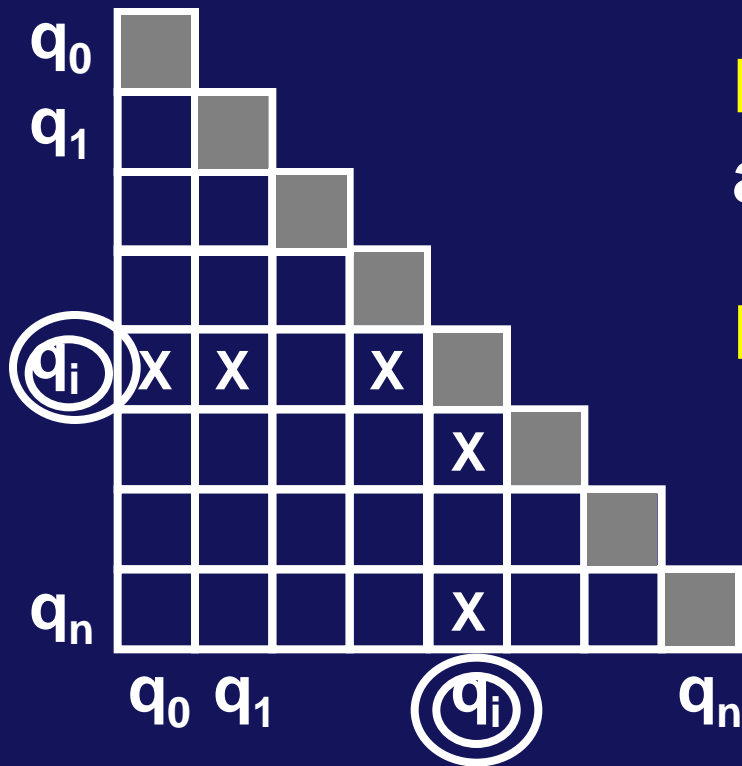
- Make best effort to find pairs of states that are distinguishable.
- Pairs leftover will help us.

# TABLE-FILLING ALGORITHM



**Base Case:**  $p$  accepts  
and  $q$  rejects  $\Rightarrow p \neq q$

# TABLE-FILLING ALGORITHM

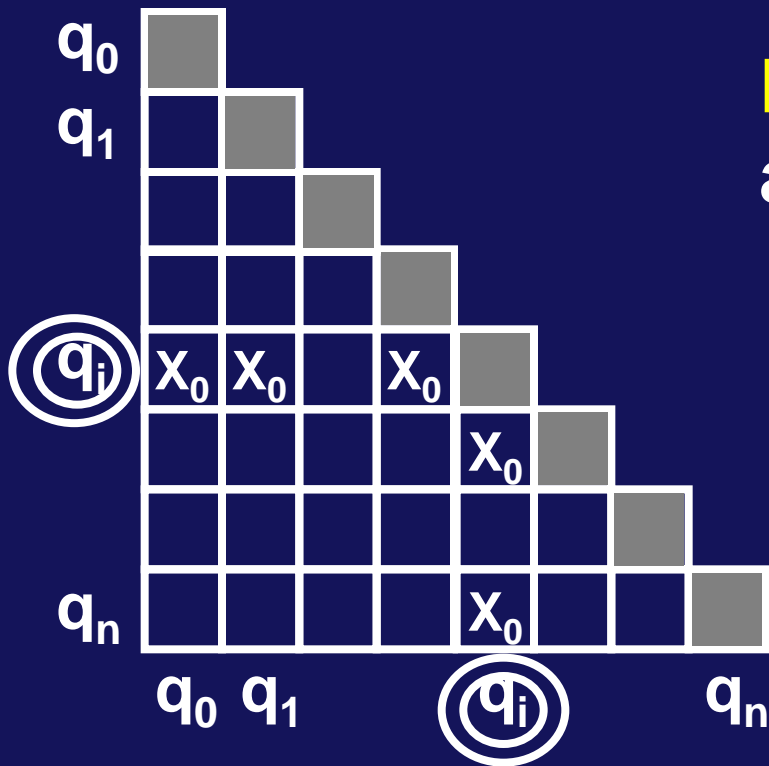


**Base Case:**  $p$  accepts  
and  $q$  rejects  $\Rightarrow p \neq q$

**Recursion:**

# TABLE-FILLING ALGORITHM

For base case, we put  $X_0$  in the corresponding cell.  
This states, that the two states are not equivalent.



**Base Case:**  $p$  accepts  
and  $q$  rejects  $\Rightarrow p \neq q$



$\delta \sim \delta$

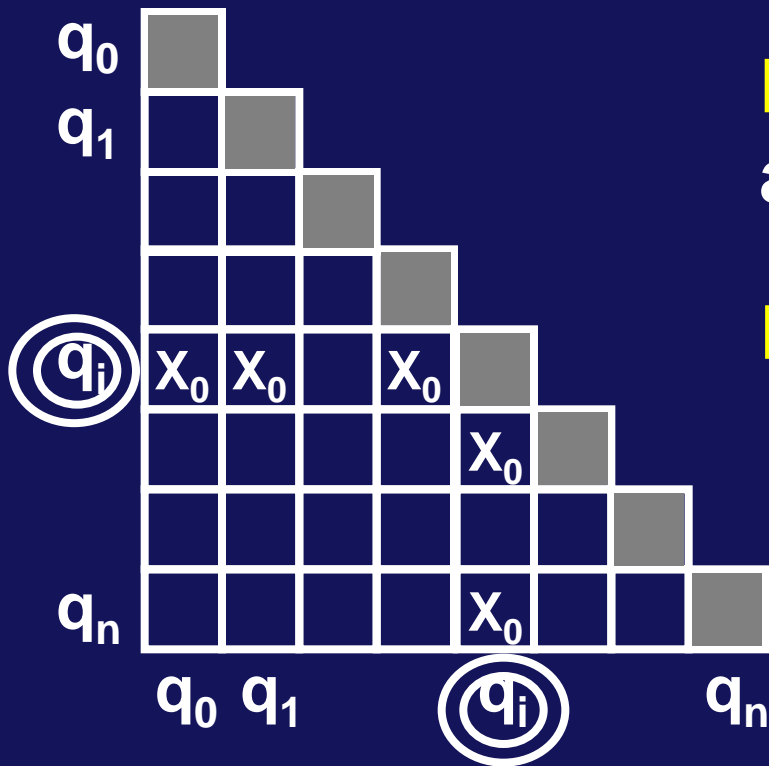


$\Rightarrow$

$p \sim q$

# TABLE-FILLING ALGORITHM

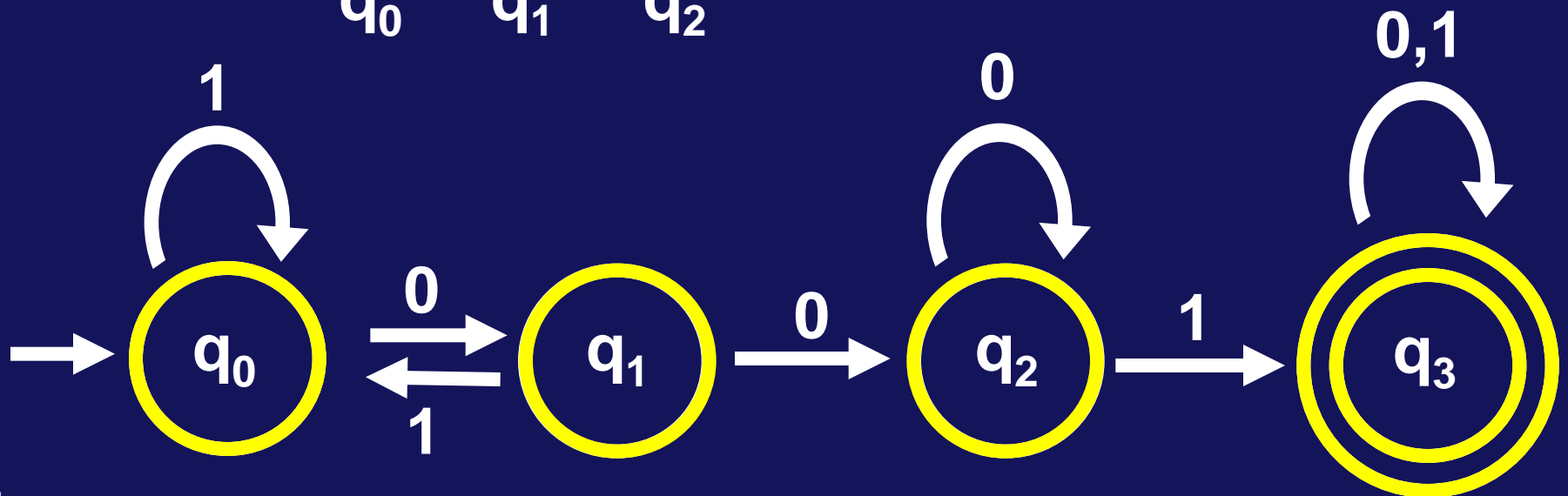
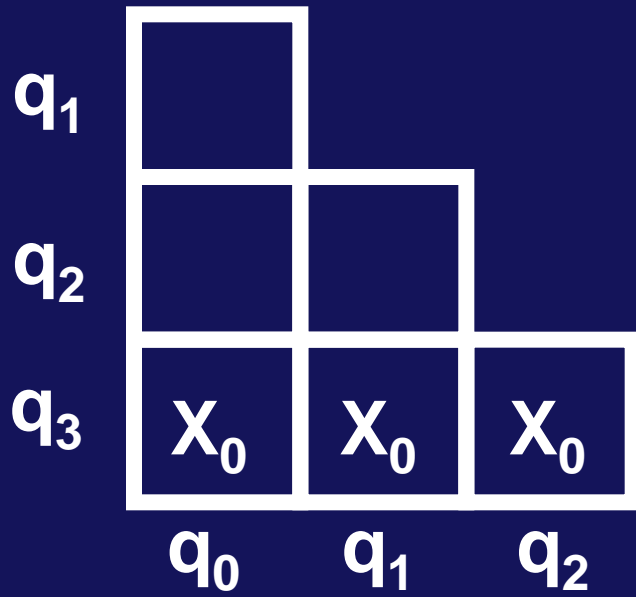
For base case, we put  $X_0$  in the corresponding cell.  
This states, that the two states are not equivalent.



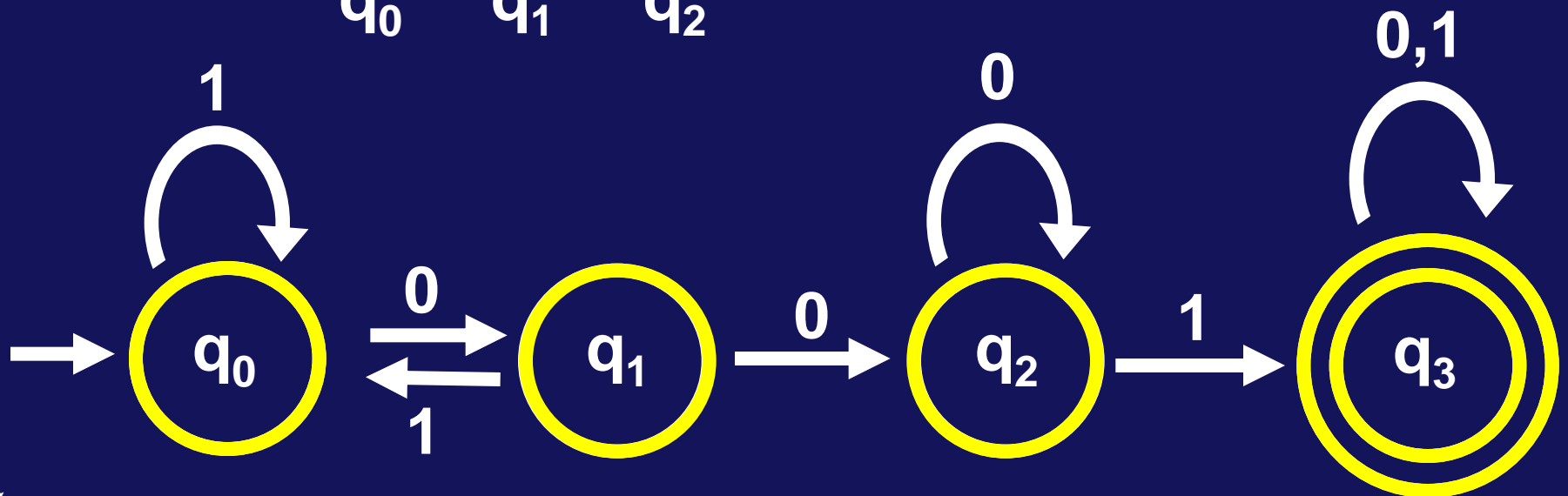
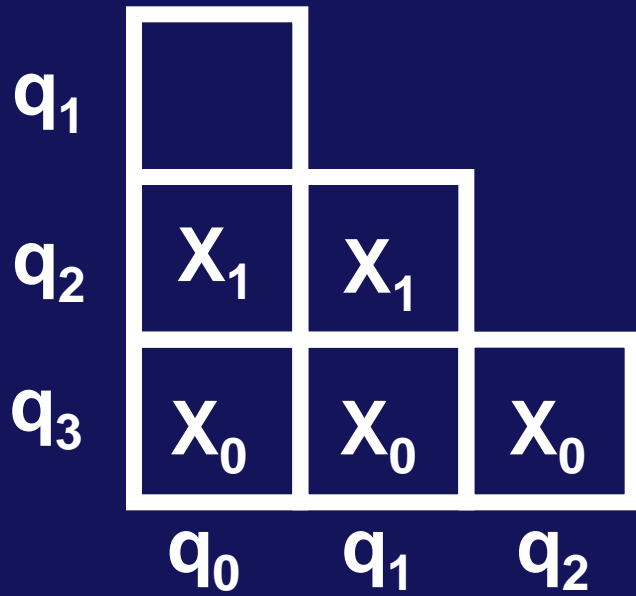
**Base Case:**  $p$  accepts  
and  $q$  rejects  $\Rightarrow p \neq q$

**Recursion:**

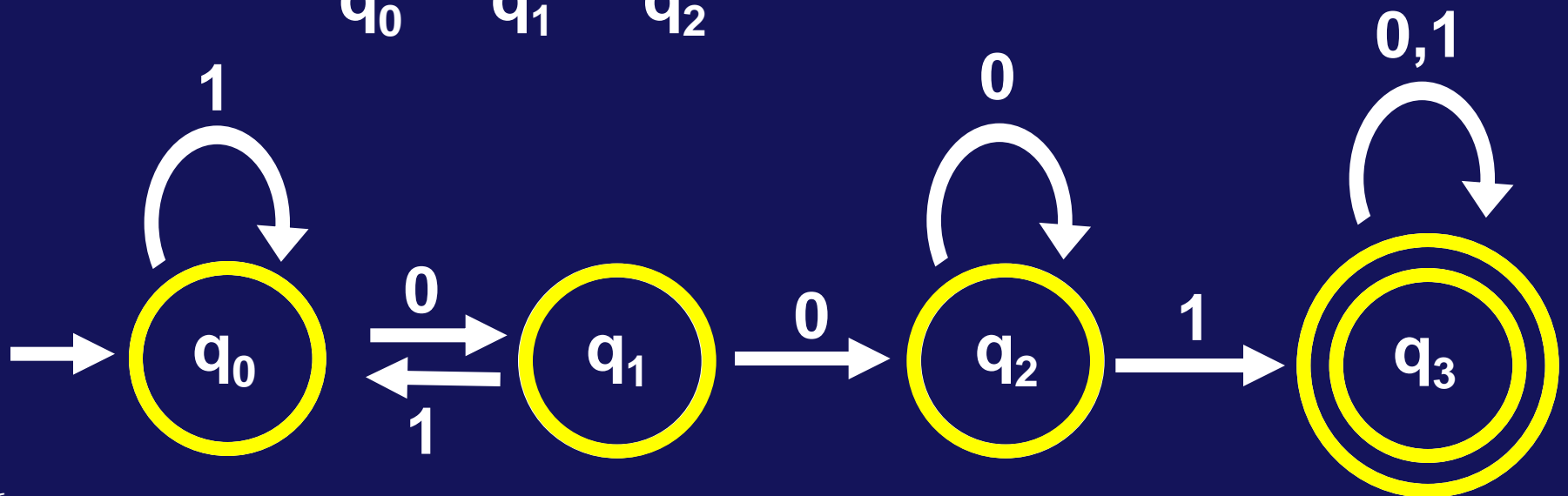
$$\begin{array}{lcl} p & \xrightarrow{a} & r \\ q & \xrightarrow{a} & s \end{array} \neq \Rightarrow p \neq q$$



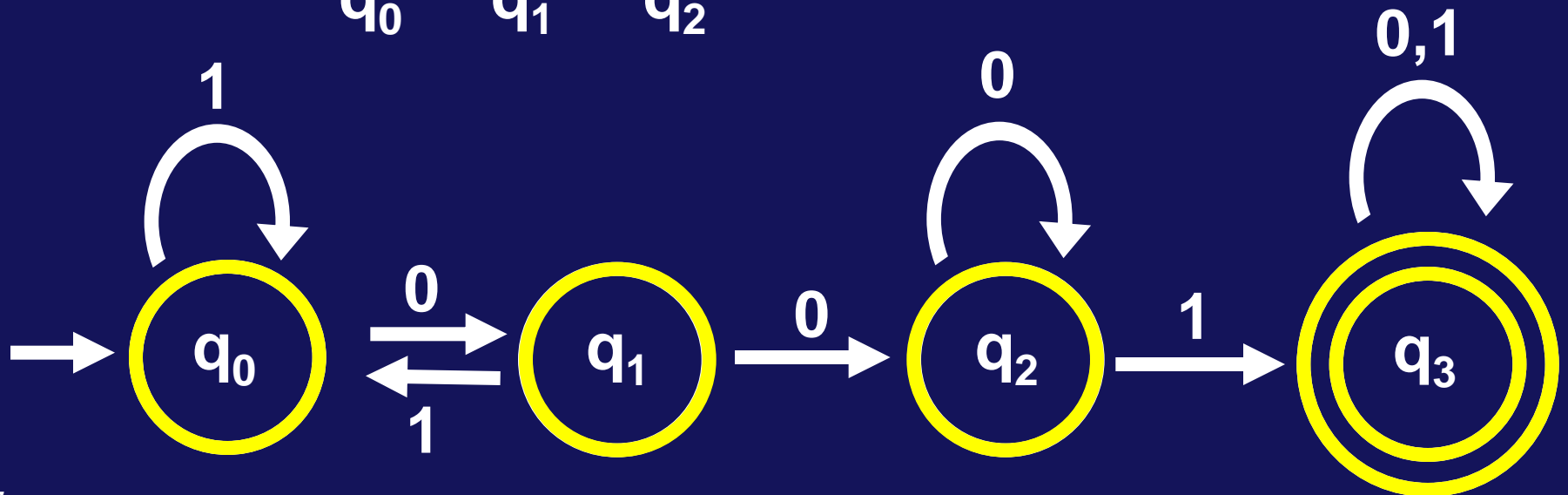
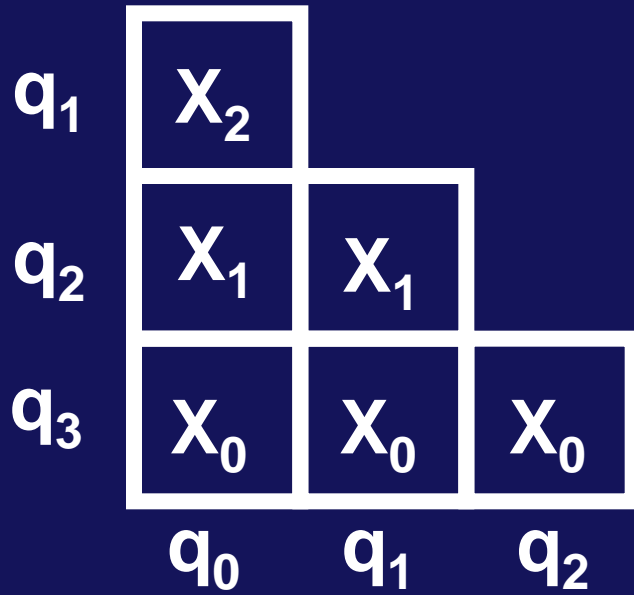




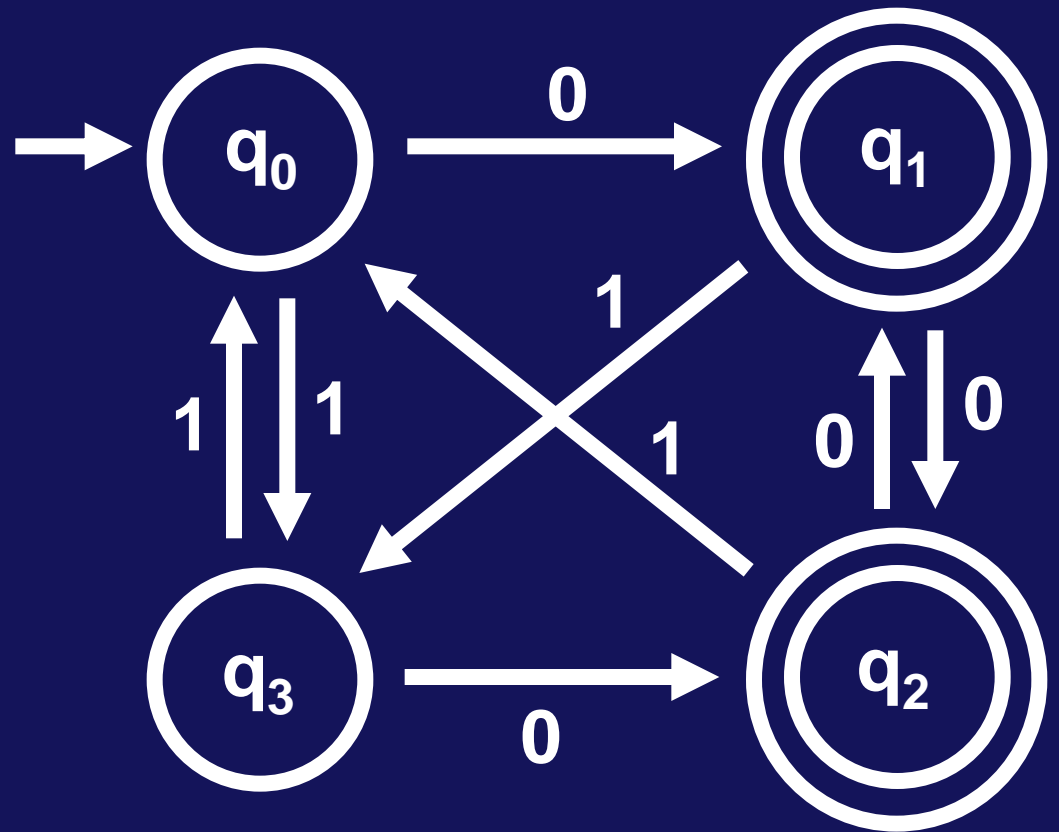
$q_1$	$x_2$		
$q_2$	$x_1$	$x_1$	
$q_3$	$x_0$	$x_0$	$x_0$
	$q_0$	$q_1$	$q_2$

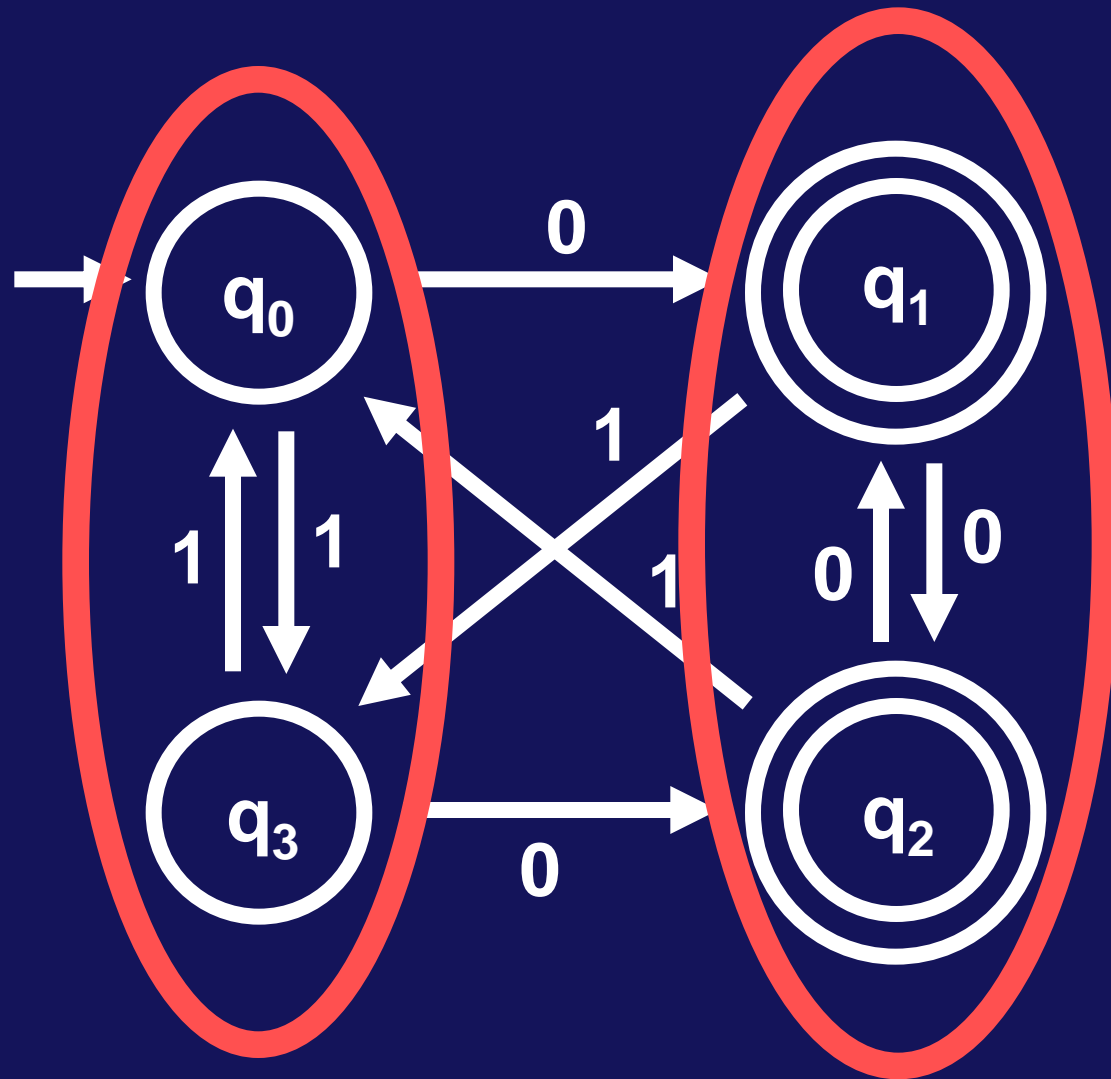


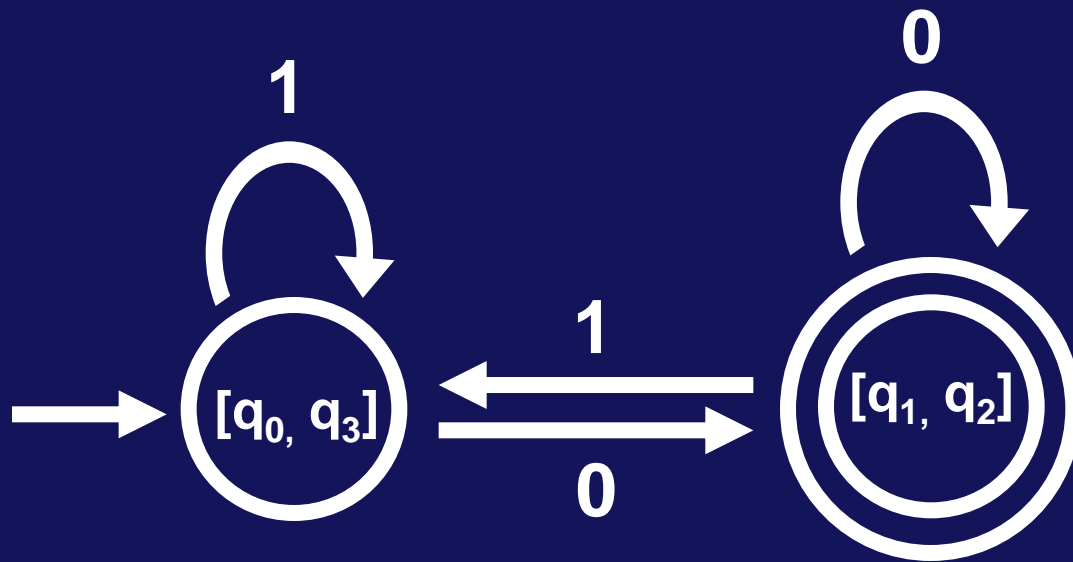
Have you noticed the reason for putting symbols  $X_1, X_2$



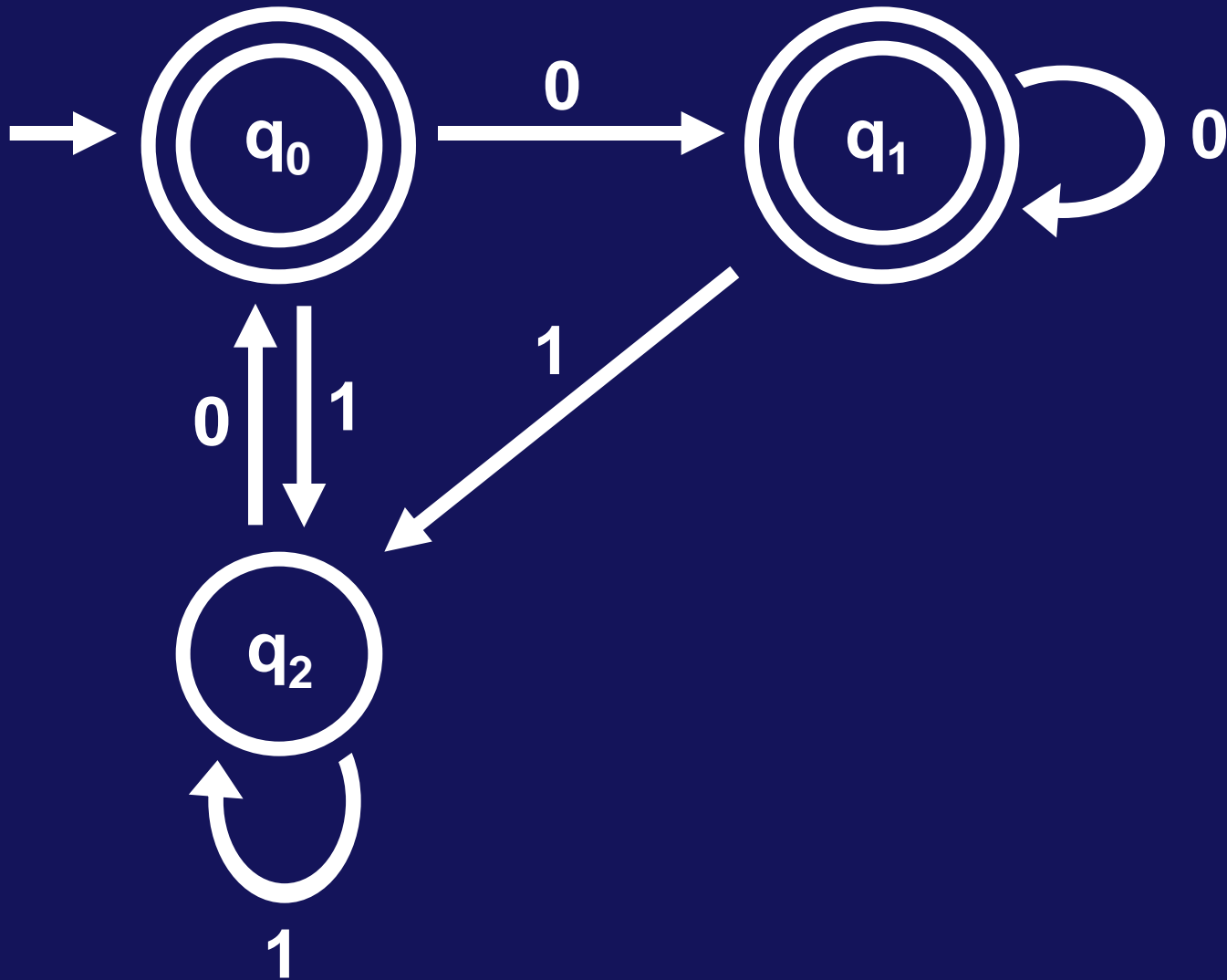
$q_1$	$x_0$		
$q_2$	$x_0$		
$q_3$		$x_0$	$x_0$
	$q_0$	$q_1$	$q_2$



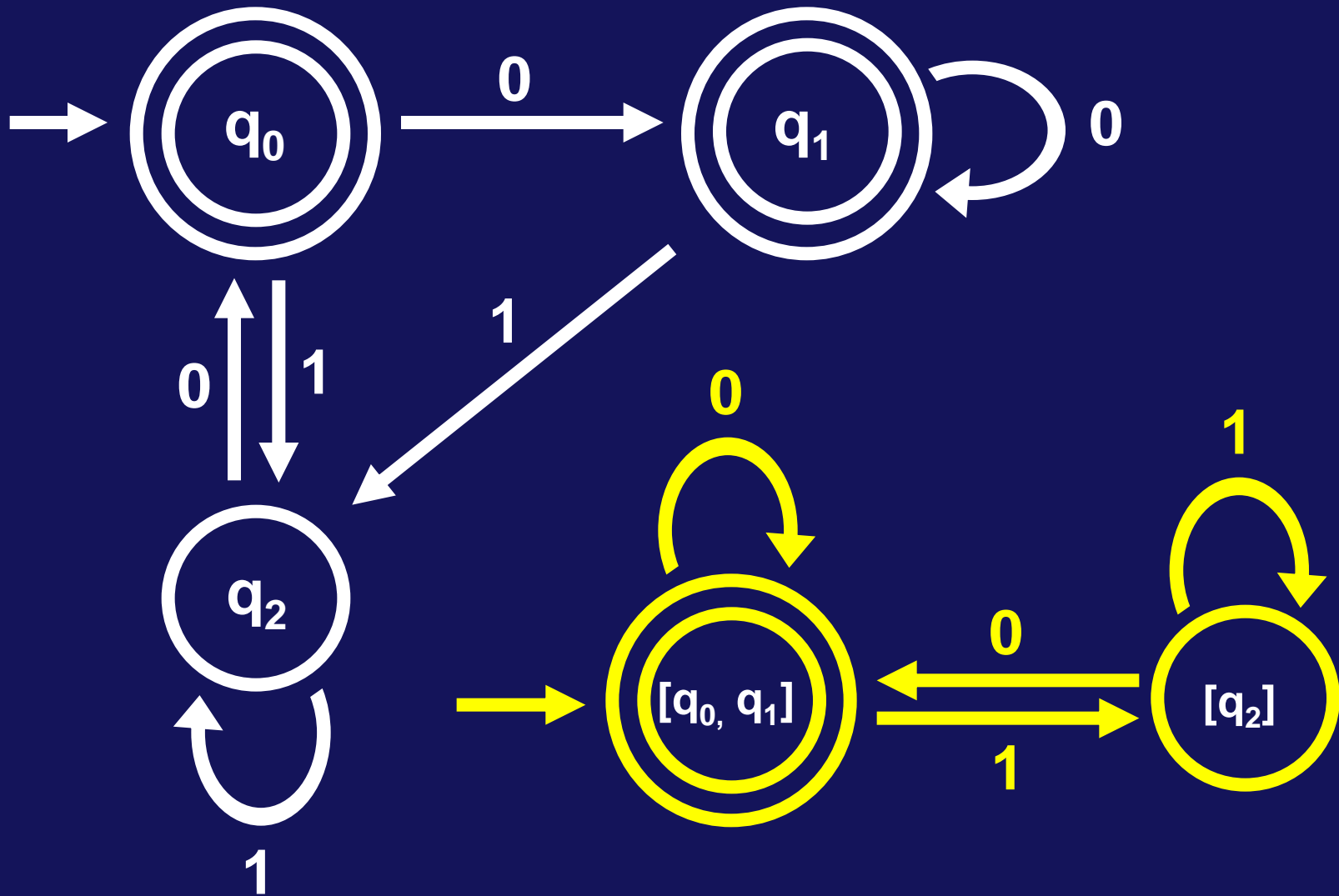




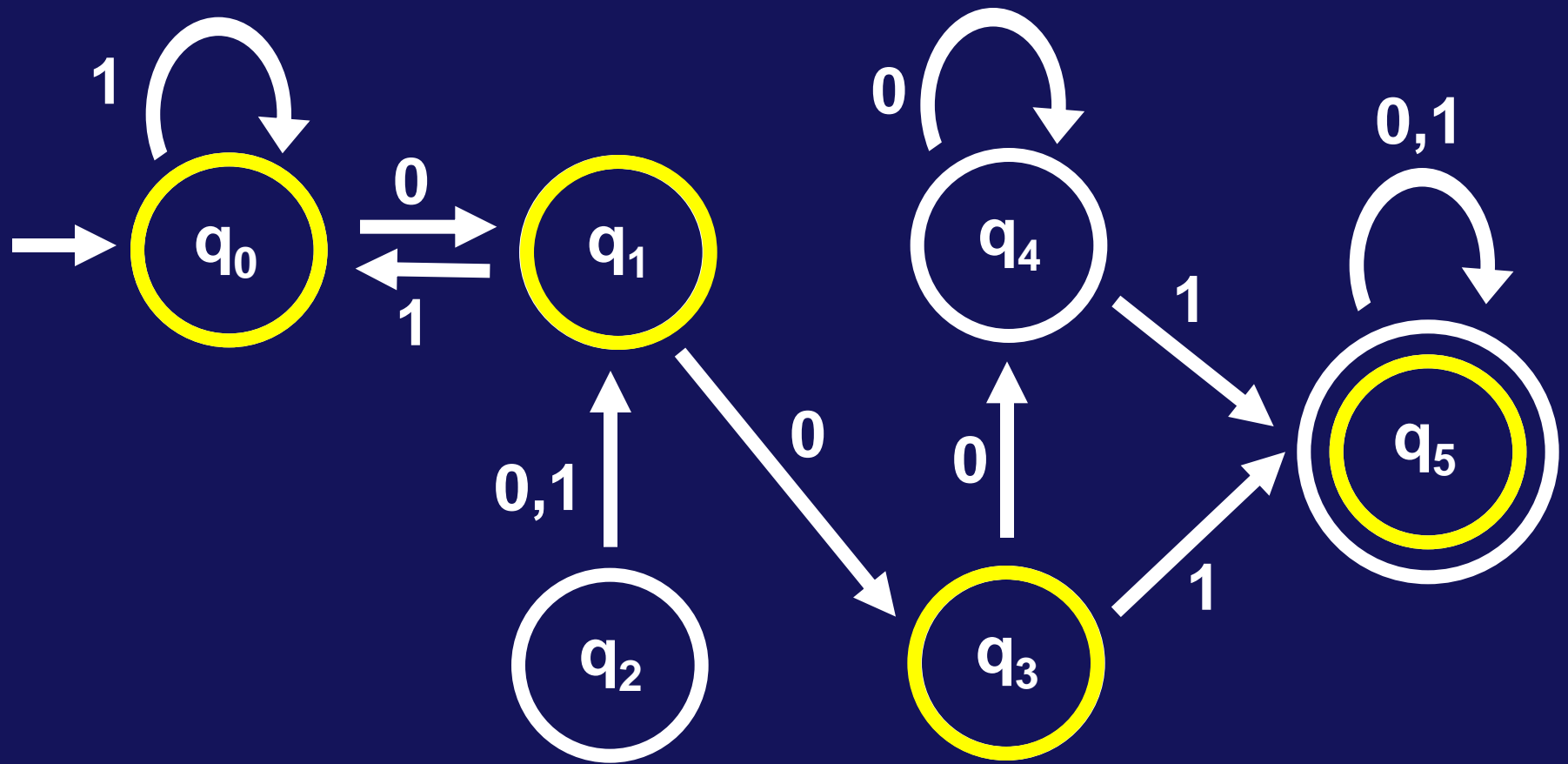
# MINIMIZE

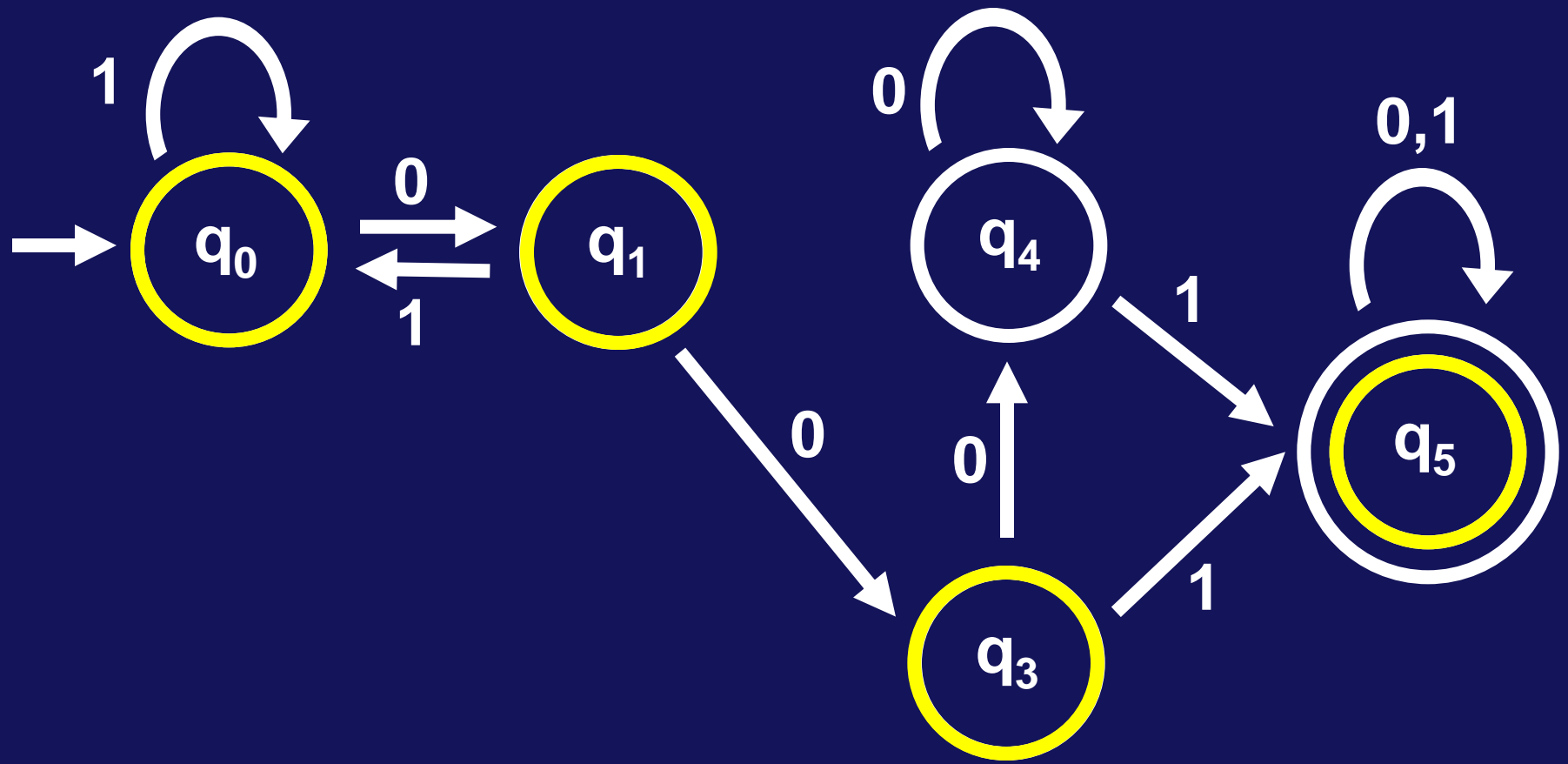


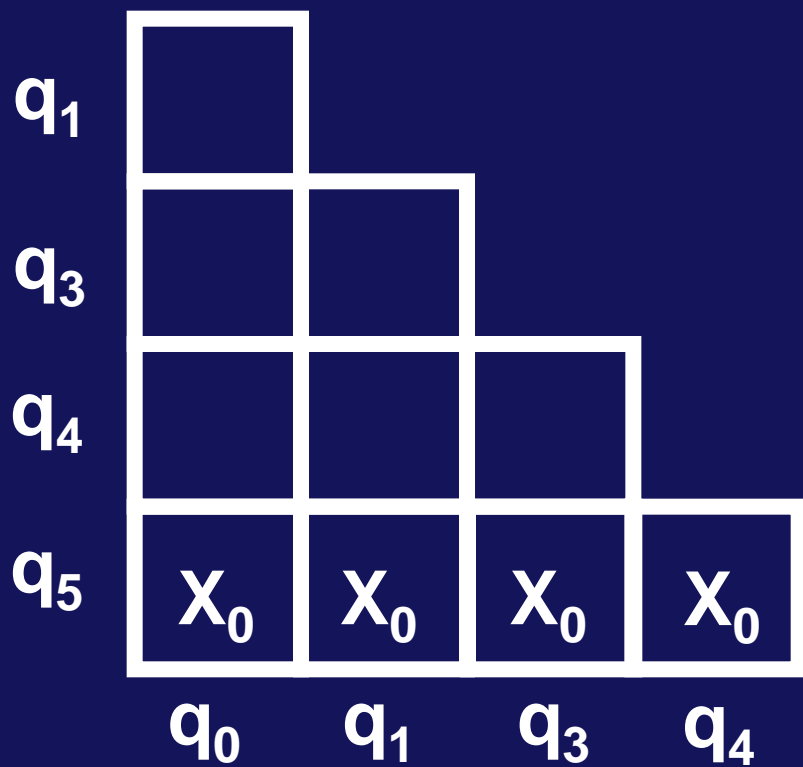
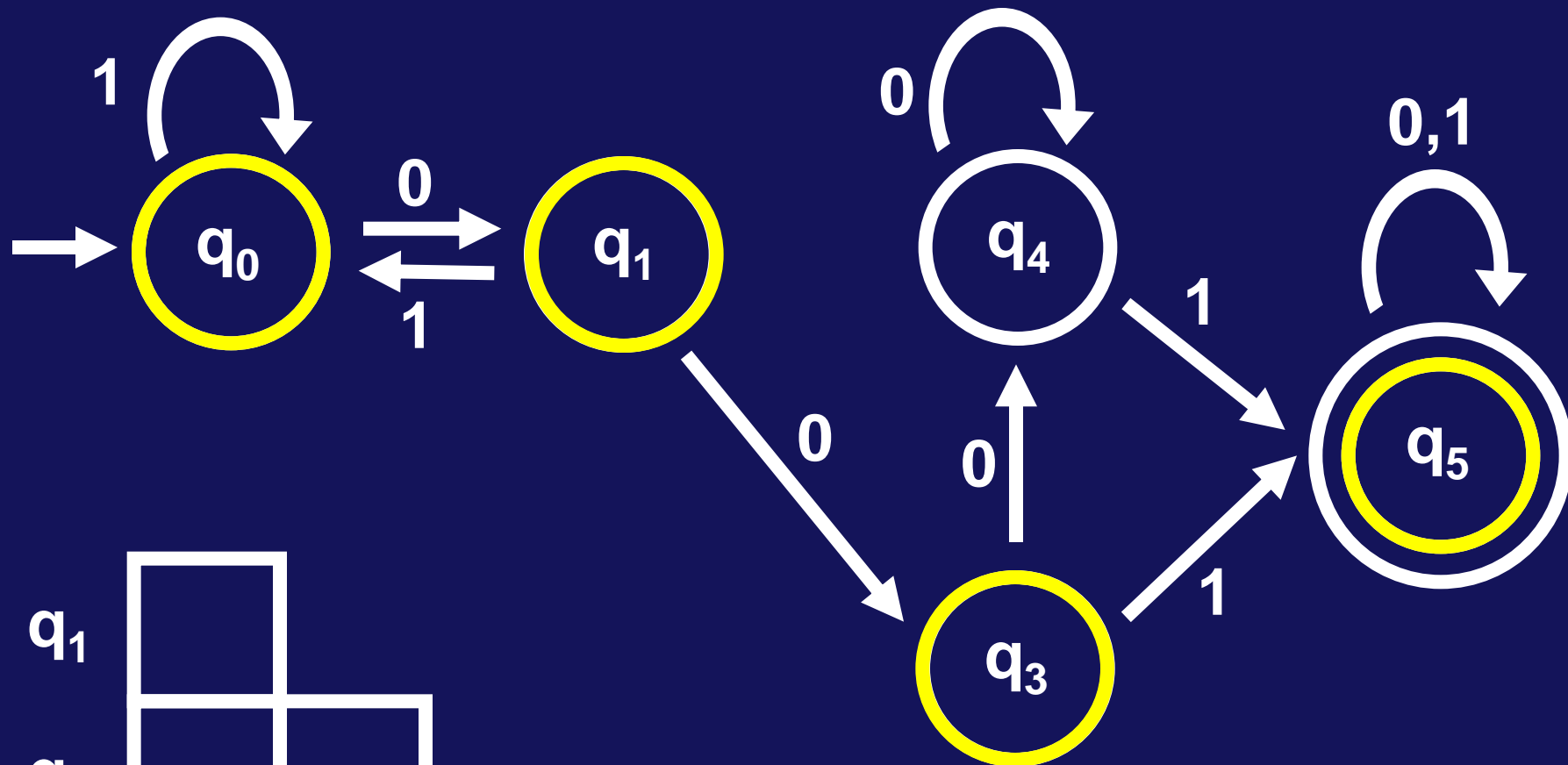
# MINIMIZE

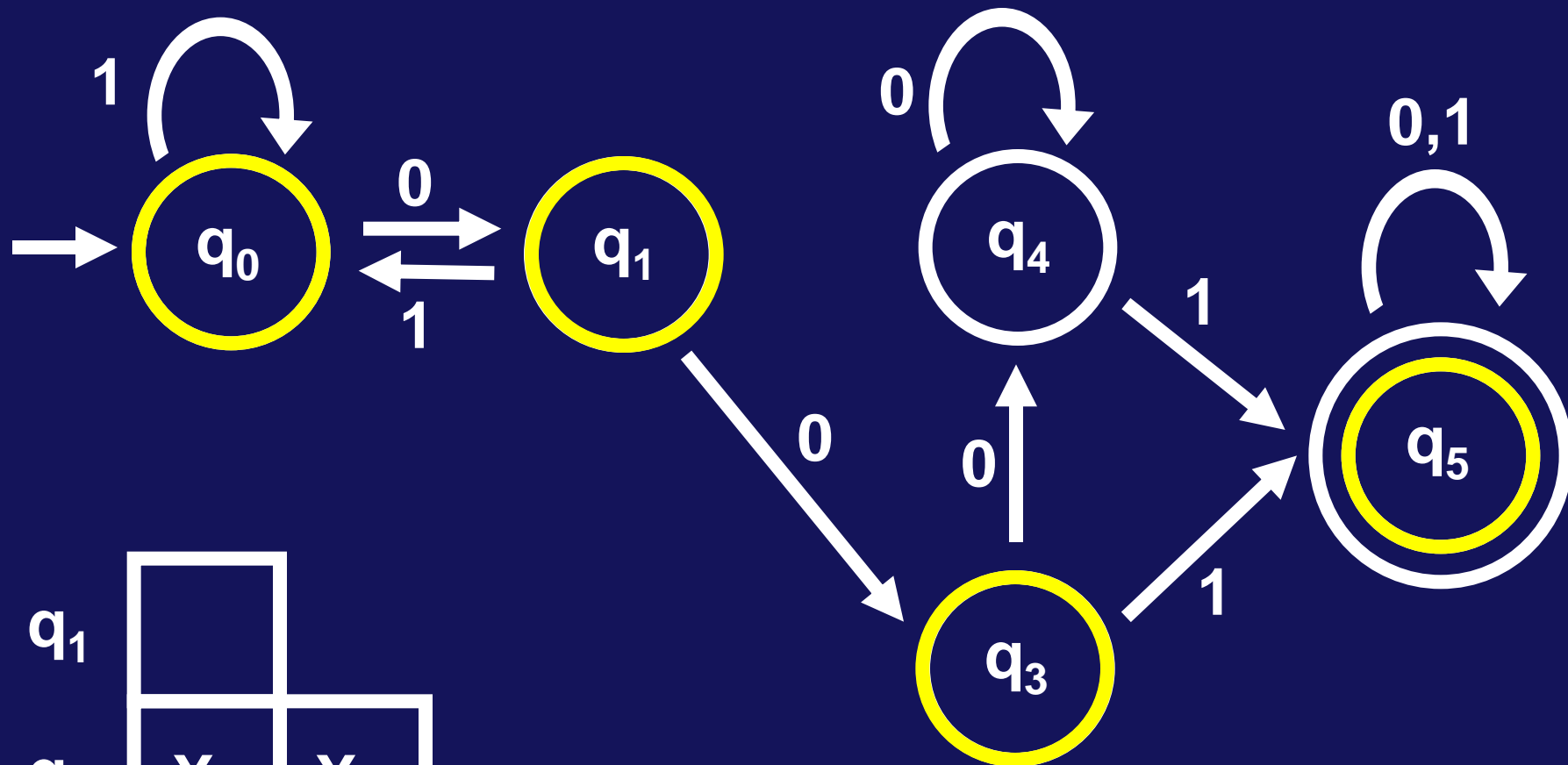




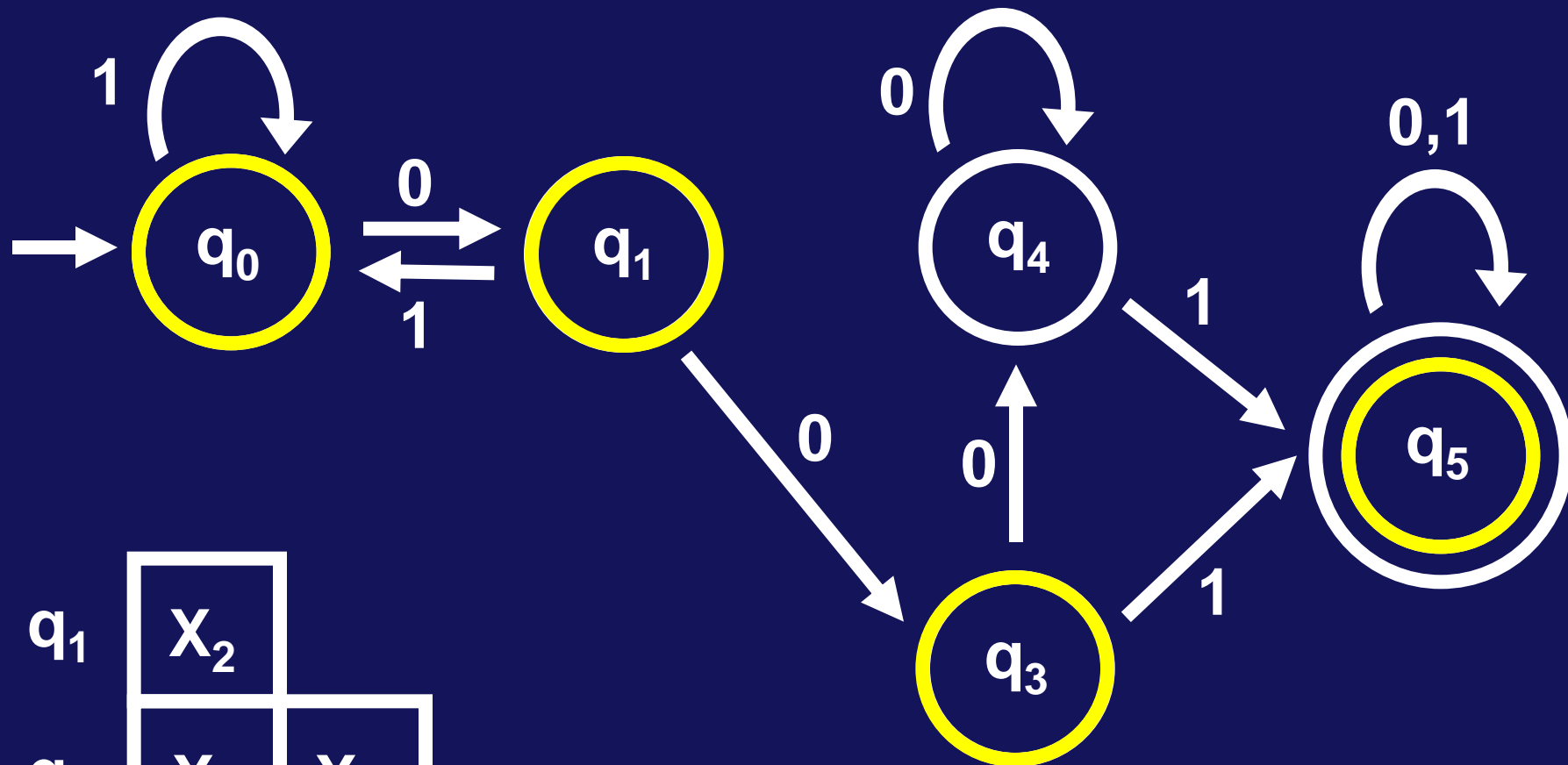




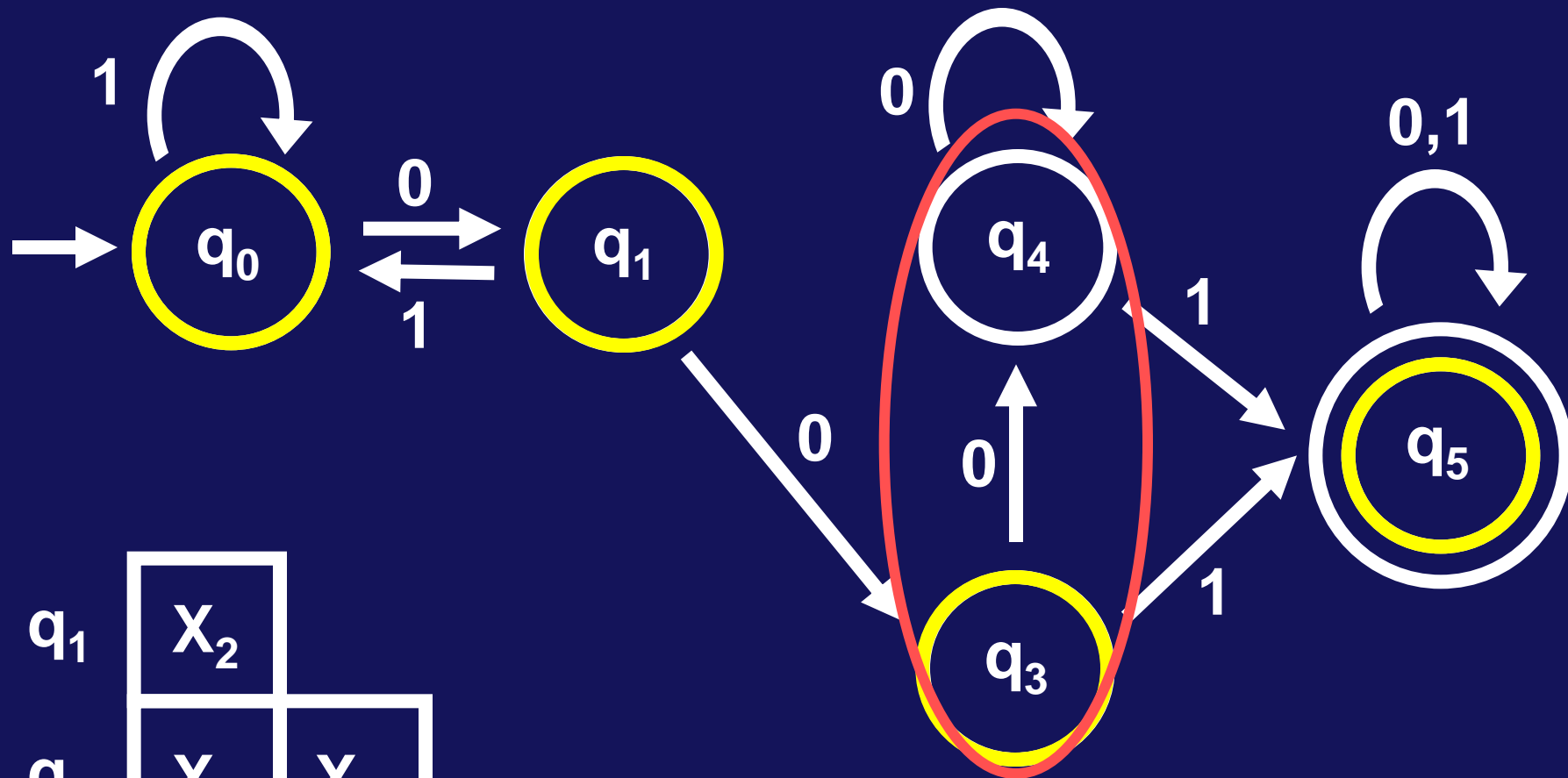




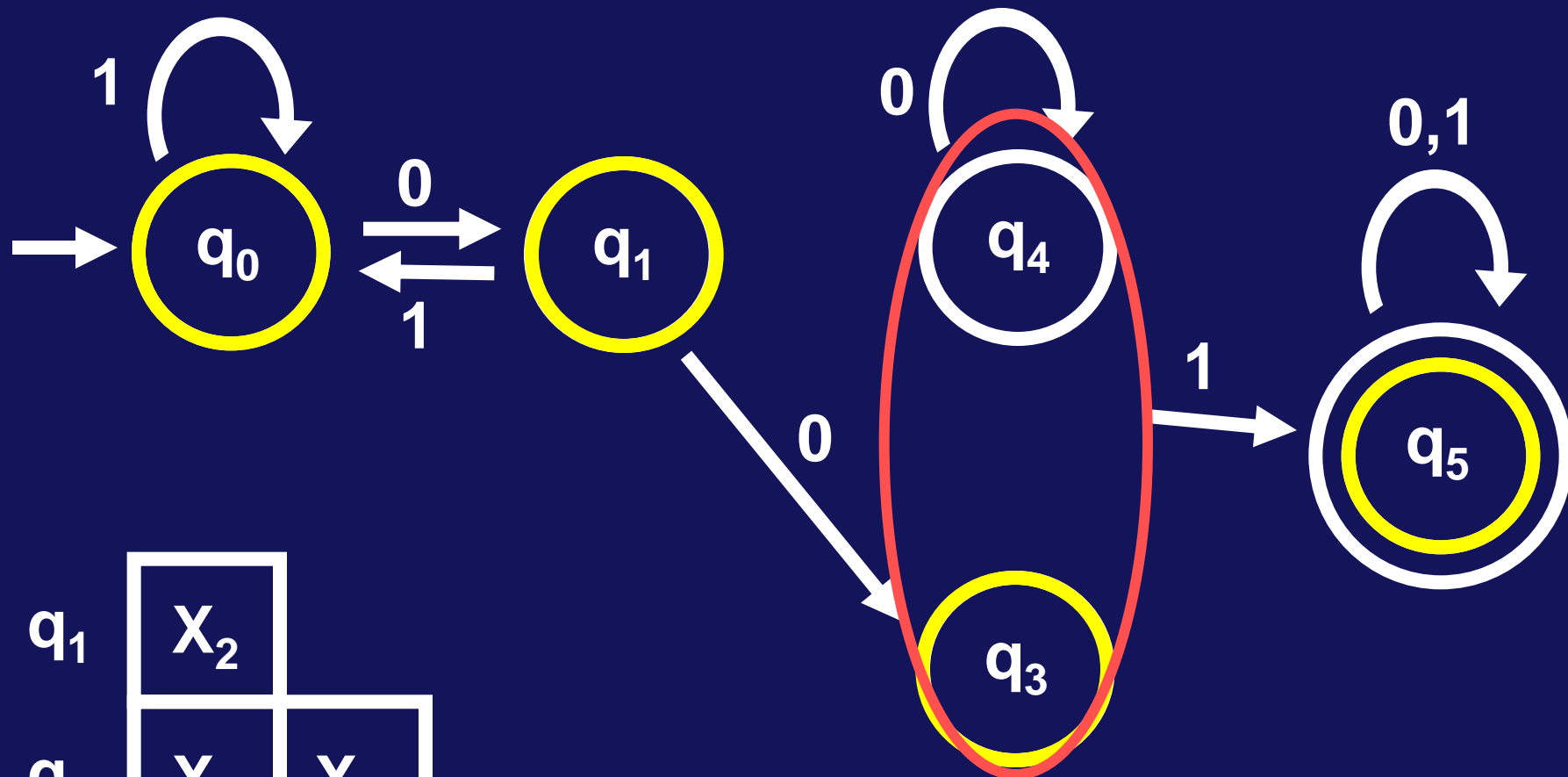
$q_1$				
$q_3$	$x_1$	$x_1$		
$q_4$	$x_1$	$x_1$		
$q_5$	$x_0$	$x_0$	$x_0$	$x_0$
	$q_0$	$q_1$	$q_3$	$q_4$



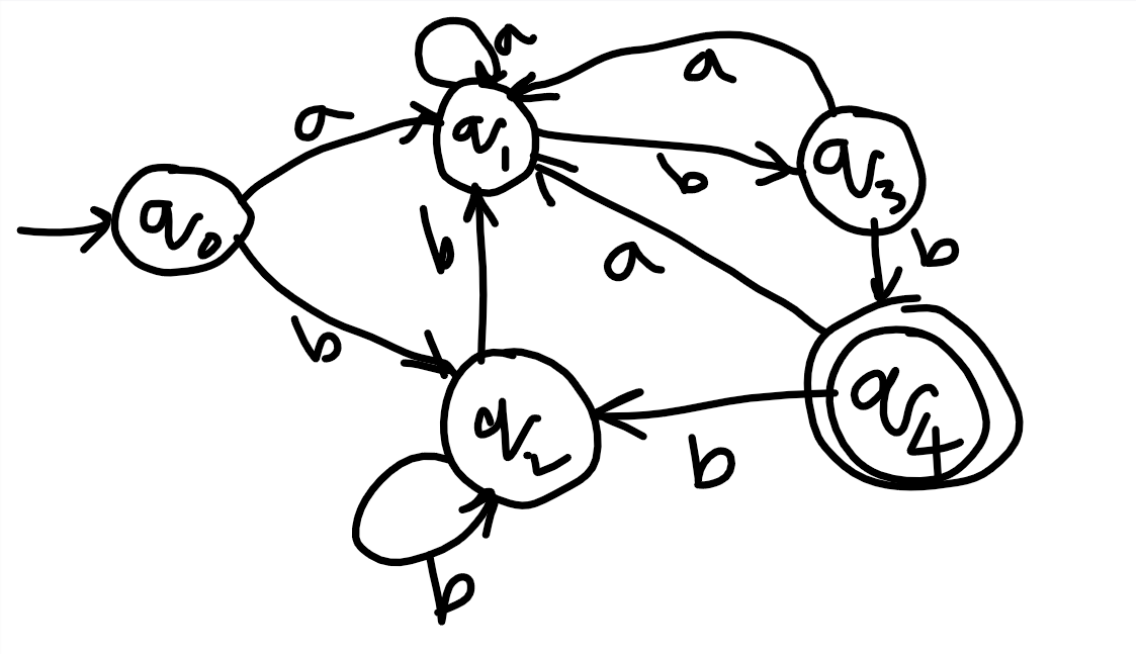
$q_1$	$x_2$			
$q_3$	$x_1$	$x_1$		
$q_4$	$x_1$	$x_1$		
$q_5$	$x_0$	$x_0$	$x_0$	$x_0$
	$q_0$	$q_1$	$q_3$	$q_4$



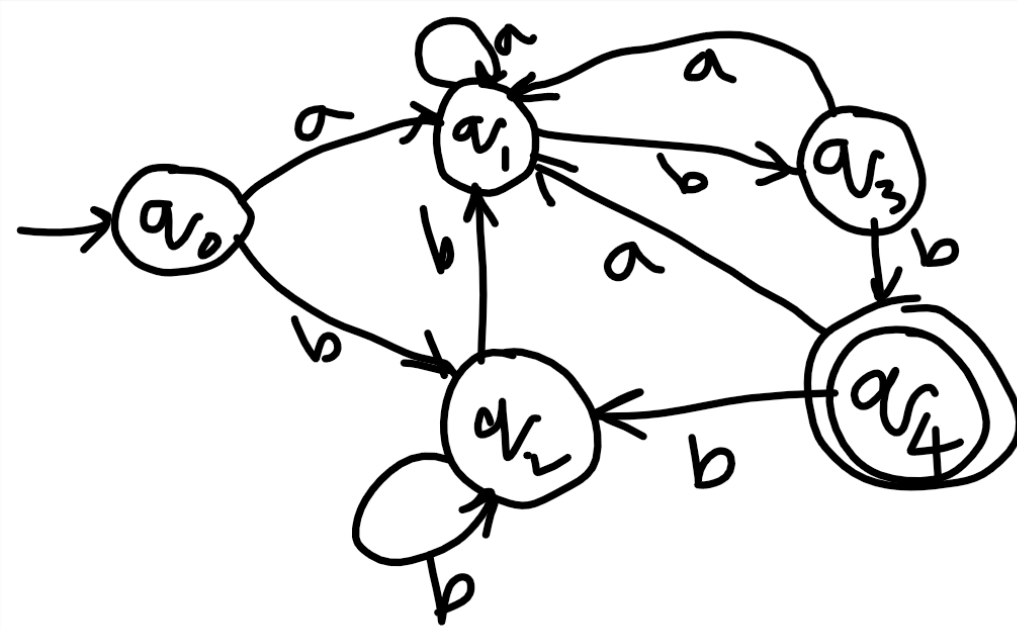
$q_1$	$x_2$			
$q_3$	$x_1$	$x_1$		
$q_4$	$x_1$	$x_1$		
$q_5$	$x_0$	$x_0$	$x_0$	$x_0$
	$q_0$	$q_1$	$q_3$	$q_4$



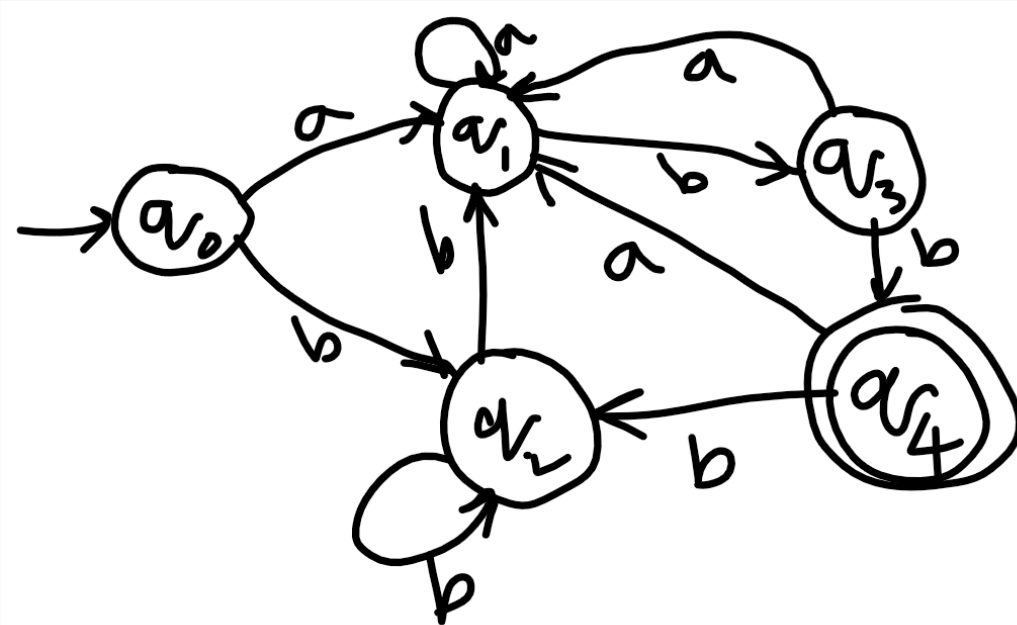
$q_1$	$x_2$			
$q_3$	$x_1$	$x_1$		
$q_4$	$x_1$	$x_1$		
$q_5$	$x_0$	$x_0$	$x_0$	$x_0$
	$q_0$	$q_1$	$q_3$	$q_4$





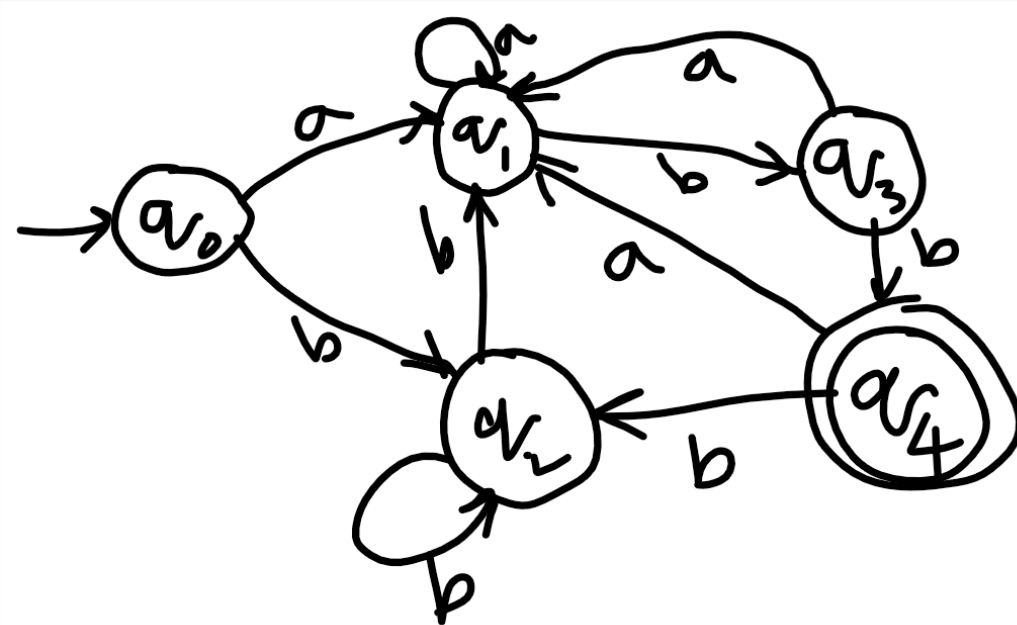


	a	b
→ q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>
q <sub>1</sub>	q <sub>1</sub>	q <sub>3</sub>
q <sub>2</sub>	q <sub>1</sub>	q <sub>2</sub>
q <sub>3</sub>	q <sub>1</sub>	q <sub>4</sub>
* q <sub>4</sub>	q <sub>1</sub>	q <sub>2</sub>



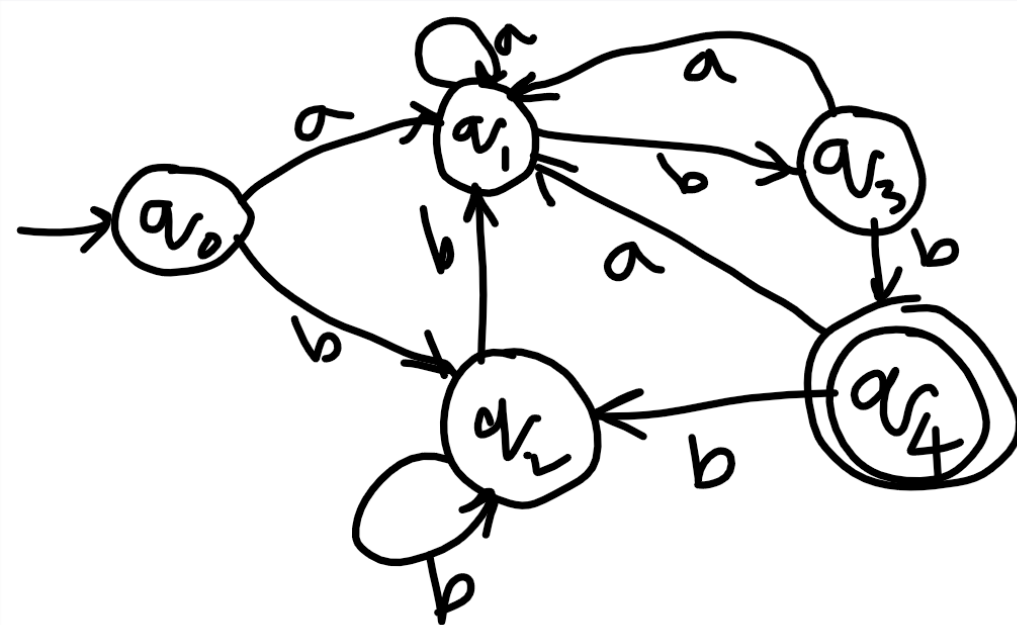
	$q_0$	$q_1$	$q_2$	$q_3$
$q_0$	X0			
$q_1$	X0			
$q_2$	X0			
$q_3$	X0			

	a	b
$q_0$	$q_1$	$q_2$
$q_1$	$q_1$	$q_3$
$q_2$	$q_1$	$q_2$
$q_3$	$q_1$	$q_4$
* $q_4$	$q_1$	$q_2$



	$q_0$	$q_1$	$q_2$	$q_3$
$q_0$	X0	X1		
$q_1$	X0	X1		
$q_2$	X0	X1		
$q_3$	X0			

	a	b
$\rightarrow q_0$	$q_1$	$q_2$
$q_1$	$q_1$	$q_3$
$q_2$	$q_1$	$q_2$
$q_3$	$q_1$	$q_4$
$*q_4$	$q_1$	$q_2$



	$q_0$	$q_1$	$q_2$	$q_3$
$q_0$	$x_0$	$x_1$		$x_2$
$q_1$	$x_0$	$x_1$	$x_2$	
$q_2$	$x_0$	$x_1$		
$q_3$	$x_0$			

	$a$	$b$
$\rightarrow q_0$	$q_1$	$q_2$
$q_1$	$q_1$	$q_3$
$q_2$	$q_1$	$q_2$
$q_3$	$q_1$	$q_4$
$*q_4$	$q_1$	$q_2$

Eg

	0	1
→ $q_0$	$q_1$	$q_5$
$q_1$	$q_6$	$q_2$
* $q_2$	$q_0$	$q_2$
<del>Removed</del> $q_3$	$q_2$	$q_6$
$q_4$	$q_7$	$q_5$
$q_5$	$q_2$	$q_6$
$q_6$	$q_6$	$q_7$
$q_7$	$q_6$	$q_2$

Step 1: Identify unreachable states

$$\{q_0\}^+ = \{q_0, q_1, q_5, q_6, q_2, q_4, q_7\}$$

$q_3$  is not reachable

Eg

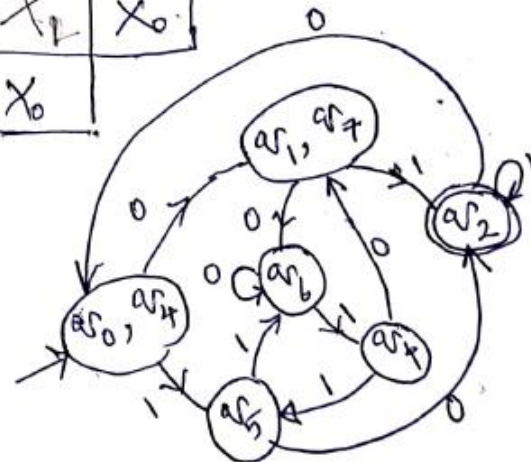
	0	1
→ $q_0$	$q_1$	$q_5$
$q_1$	$q_6$	$q_2$
* $q_2$	$q_0$	$q_2$
Removed $q_3$	$q_2$	$q_6$
$q_4$	$q_7$	$q_5$
$q_5$	$q_2$	$q_6$
$q_6$	$q_6$	$q_7$
$q_7$	$q_6$	$q_2$

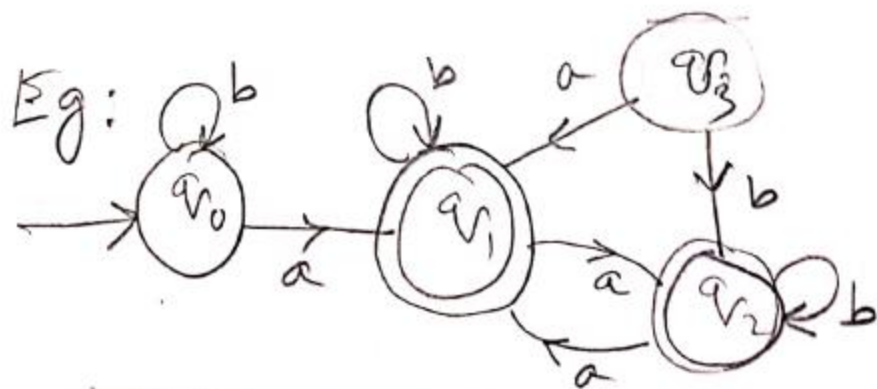
Step 1: Identify unreachable states

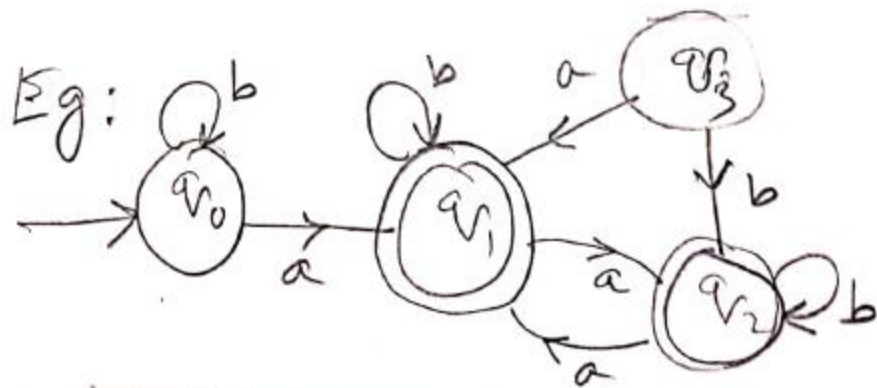
$$\{q_0\}^+ = \{q_0, q_1, q_5, q_6, q_2, q_4, q_7\}$$

$q_3$  is not reachable

	$q_7$	$q_6$	$q_5$	$q_4$	$q_2$	$q_1$
$q_0$	X <sub>1</sub>	X <sub>2</sub>	X <sub>1</sub>	✓	X <sub>0</sub>	X <sub>1</sub>
$q_1$	✓	X <sub>1</sub>	X <sub>1</sub>	X <sub>1</sub>	X <sub>0</sub>	
$q_2$	X <sub>0</sub>	X <sub>0</sub>	X <sub>0</sub>	X <sub>0</sub>		
$q_3$	X <sub>1</sub>	X <sub>2</sub>	X <sub>1</sub>			
$q_5$	X <sub>1</sub>	X <sub>1</sub>				
$q_6$	X <sub>1</sub>					

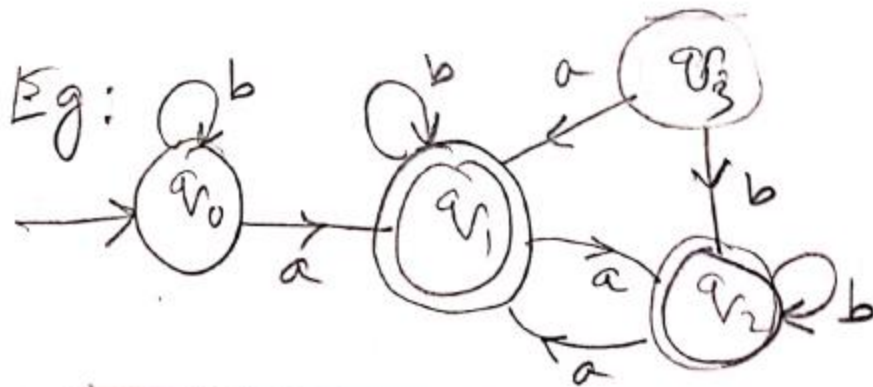






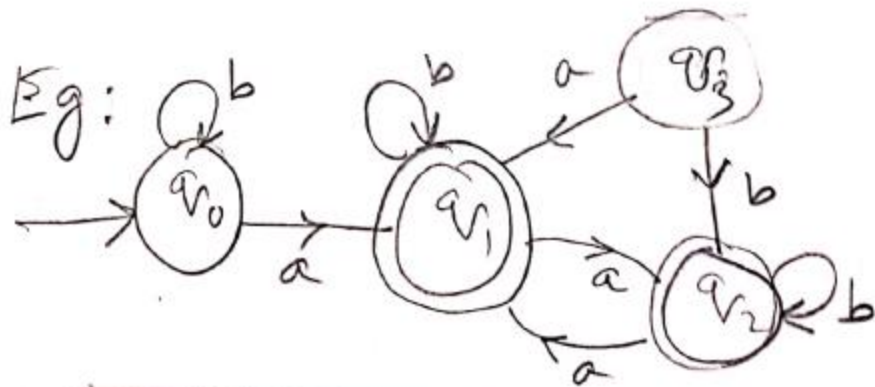
Unreachable =  $q_3$





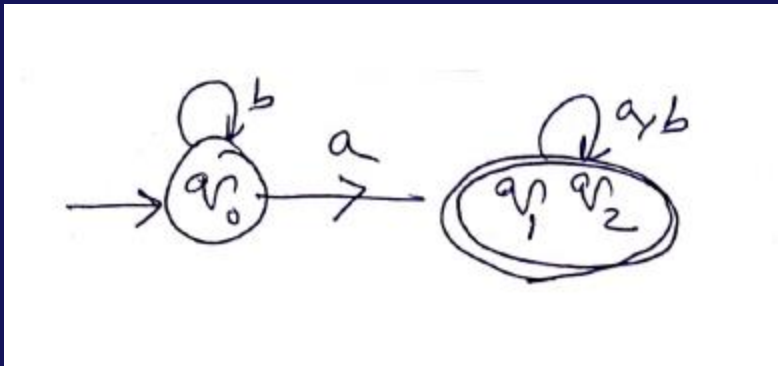
Unreachable =  $q_3$

We get  $q_1$  is equivalent to  $q_2$  (how?)



Unreachable =  $q_3$

We get  $q_1$  is equivalent to  $q_2$  (how?)



Minimal DFA

# HOW TO PROVE THAT TWO DFA<sub>s</sub> ARE EQUIVALENT

- The following is an extract from the Ullman's book.
- Read that book for more information (Reading assignment)

### 4.4.2 Testing Equivalence of Regular Languages

The table-filling algorithm gives us an easy way to test if two regular languages are the same. Suppose languages  $L$  and  $M$  are each represented in some way, e.g., one by a regular expression and one by an NFA. Convert each representation to a DFA. Now, imagine one DFA whose states are the union of the states of the DFA's for  $L$  and  $M$ . Technically, this DFA has two start states, but actually the start state is irrelevant as far as testing state equivalence is concerned, so make any state the lone start state.

Now, test if the start states of the two original DFA's are equivalent, using the table-filling algorithm. If they are equivalent, then  $L = M$ , and if not, then  $L \neq M$ .

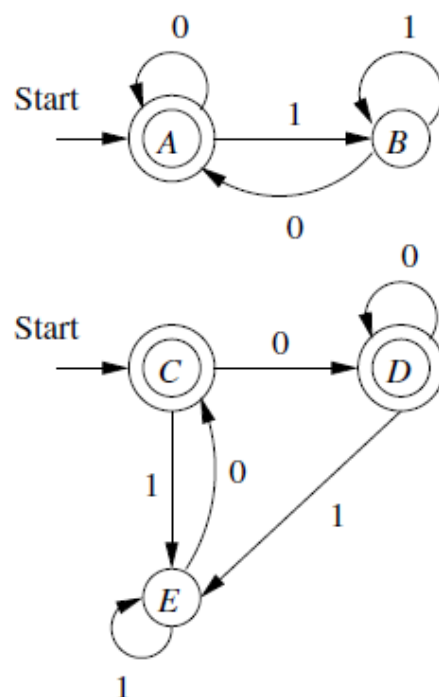


Figure 4.10: Two equivalent DFA's