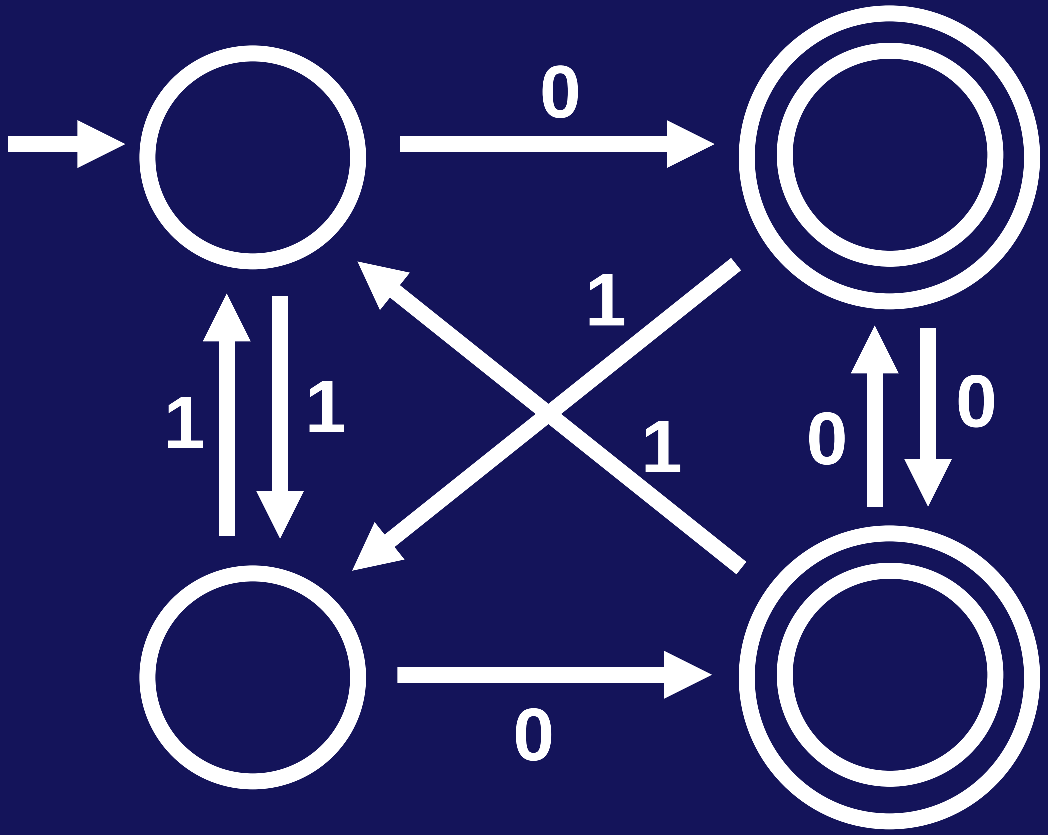
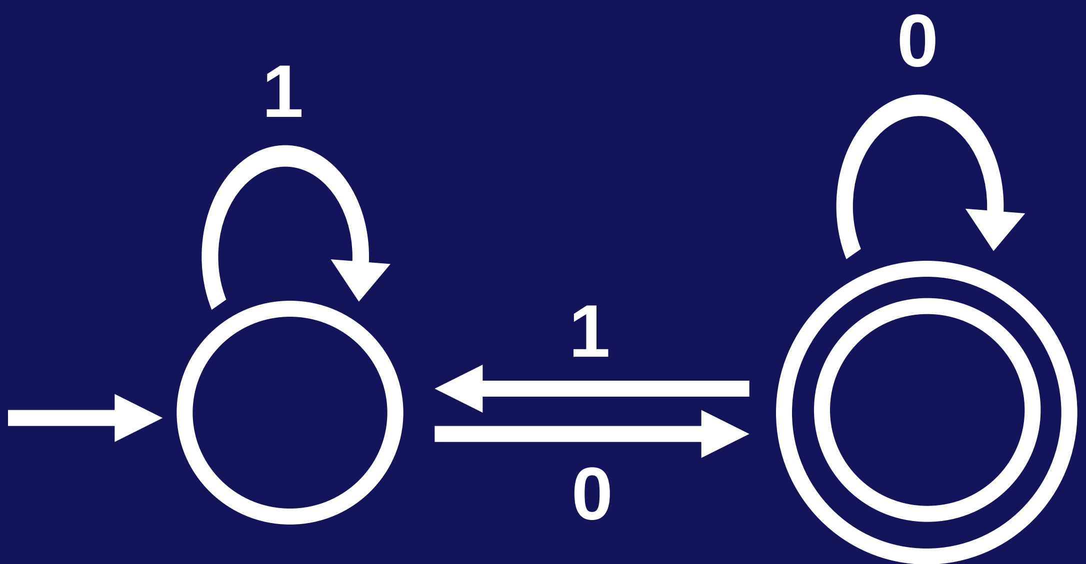
MINIMIZING DFAs

IS THIS **MINIMAL?**

NO



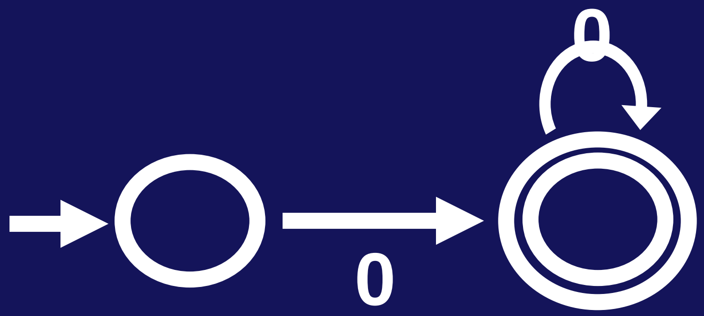
IS THIS **MINIMAL?**

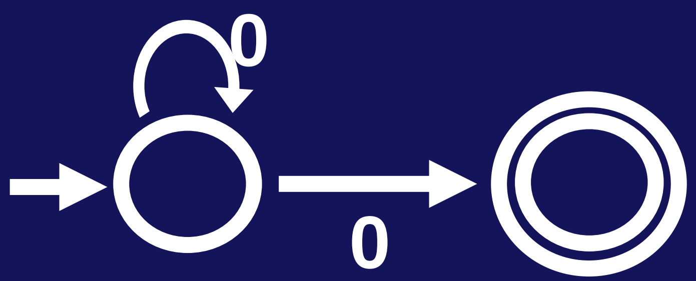


THEOREM

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

NOT TRUE FOR NFAs





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

EXTENDING δ

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$ extend δ

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

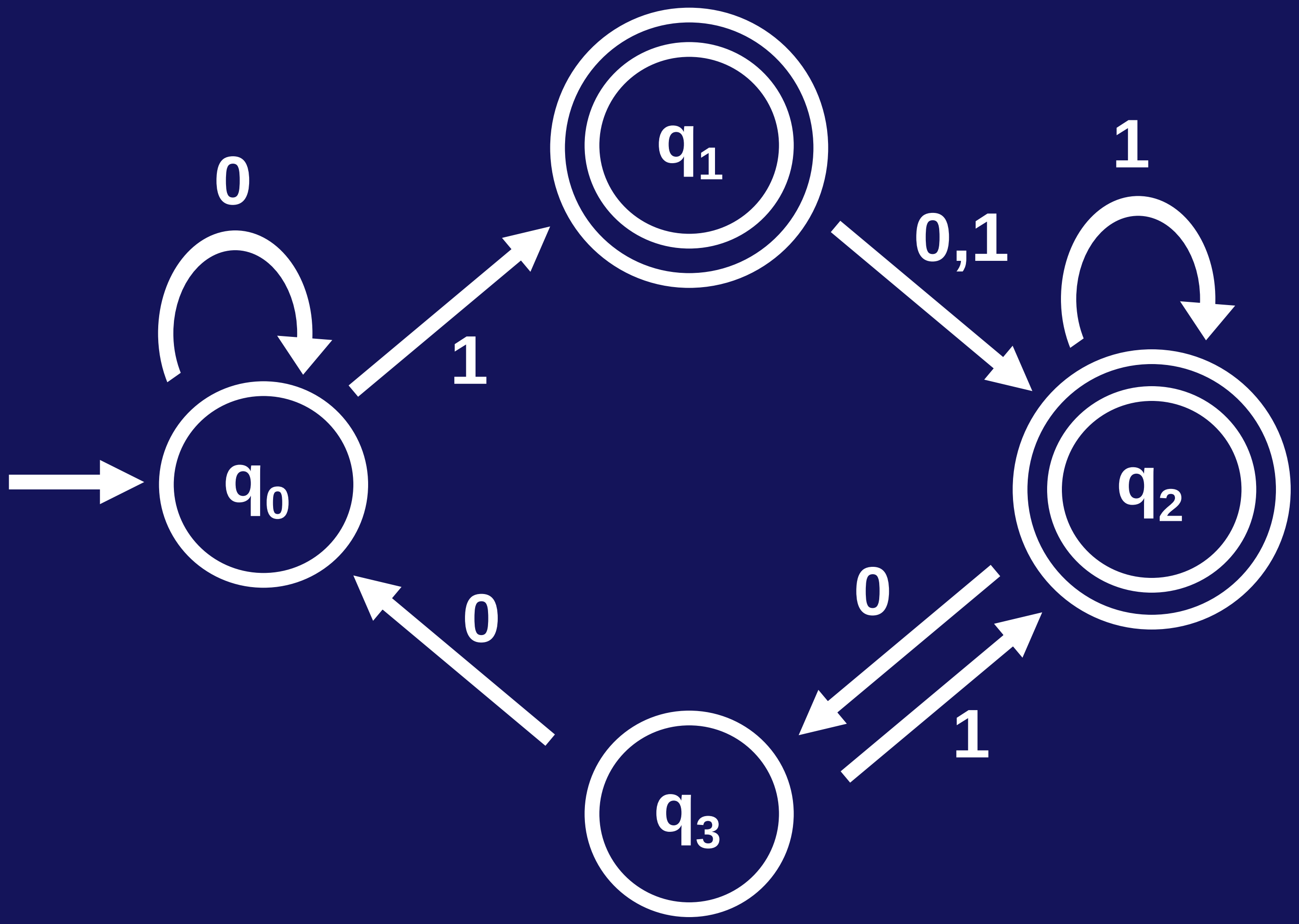
$$\hat{\delta}(q, \varepsilon) = q$$

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

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

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

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



ε distinguishes accept from non-accept states

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

Definition:

p is **distinguishable** from q iff there is a $w \in \Sigma^*$ that distinguishes p from q

p is **indistinguishable** from q iff p is **not** distinguishable from q

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

Define relation "~":

$p \sim q$ iff p is indistinguishable from q

$p \nsim q$ iff p is distinguishable from q

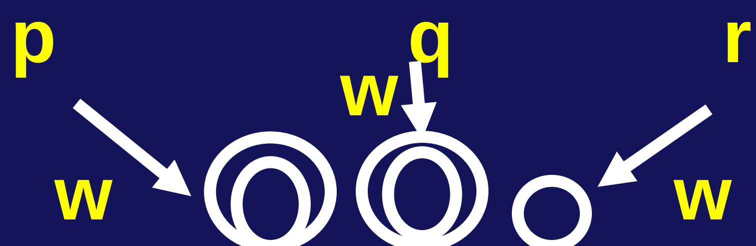
Proposition: "~" is an equivalence relation

$p \sim p$ (reflexive)

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

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

Suppose not:

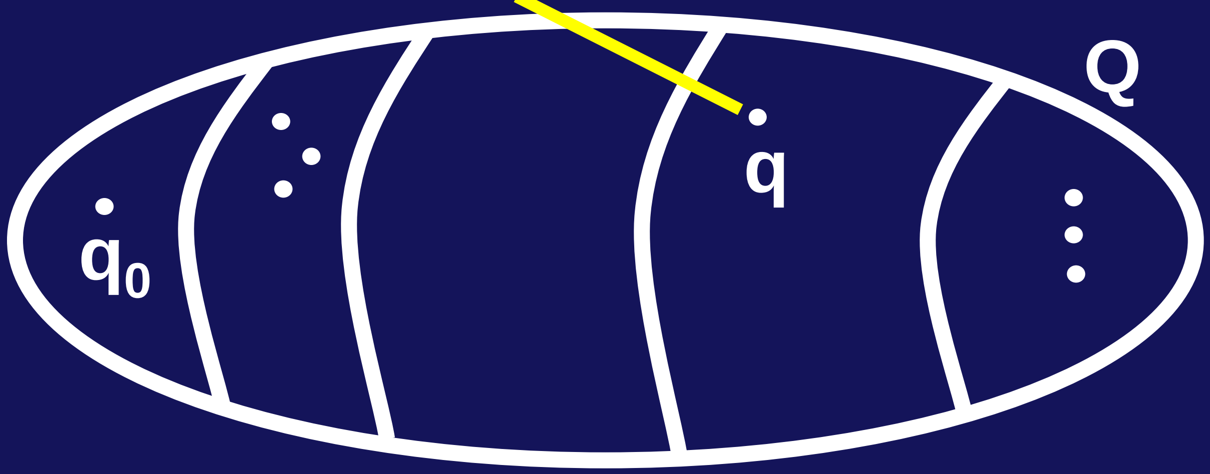


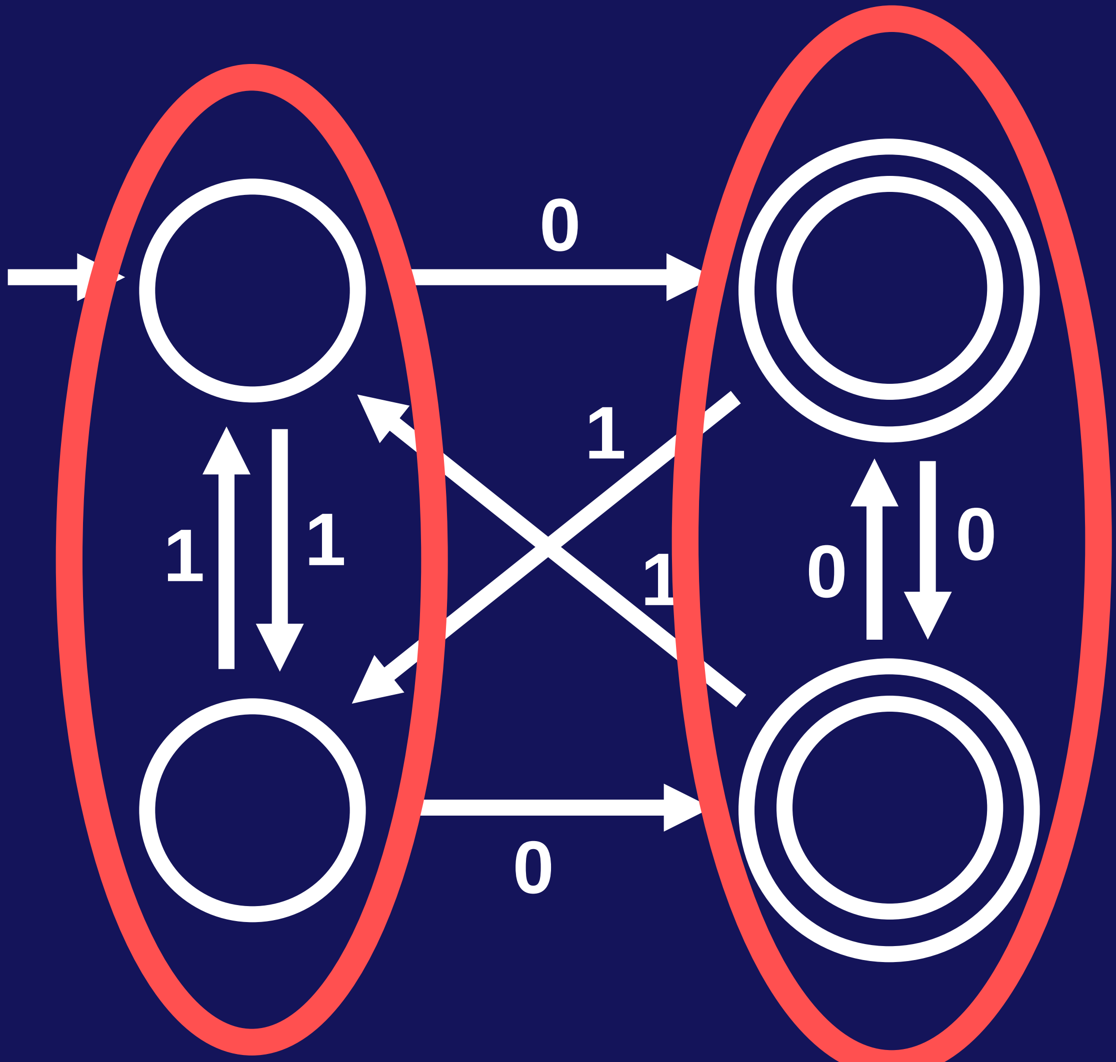
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

Proposition: "~" is an **equivalence relation**

so "~" partitions the set of states of M into disjoint equivalence classes

$[q] = \{ p \mid p \sim q \}$




0
1
1
1
1
0
0
0

Algorithm MINIMIZE

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

Input: DFA M

Output: DFA M_{MIN}

(1) Remove all inaccessible states from M

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

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

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

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

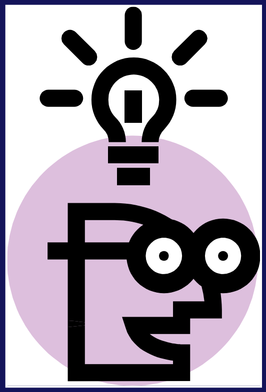
TABLE-FILLING ALGORITHM

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

Output: (1) $D_M = \{ (p,q) \mid p,q \in Q \text{ and } p \not\sim q \}$

(2) $E_M = \{ [q] \mid q \in Q \}$

IDEA!



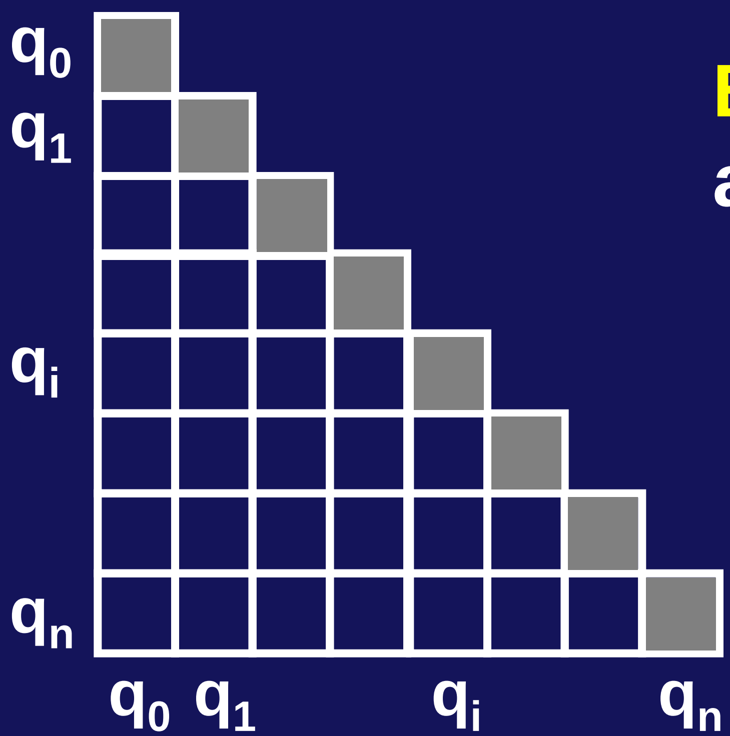
- **Make best effort to find pairs of states that are distinguishable.**
- **Pairs left over will be indistinguishable.**

TABLE-FILLING ALGORITHM

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

Output: (1) $D_M = \{ (p,q) \mid p,q \in Q \text{ and } p \not\sim q \}$

(2) $E_M = \{ [q] \mid q \in Q \}$



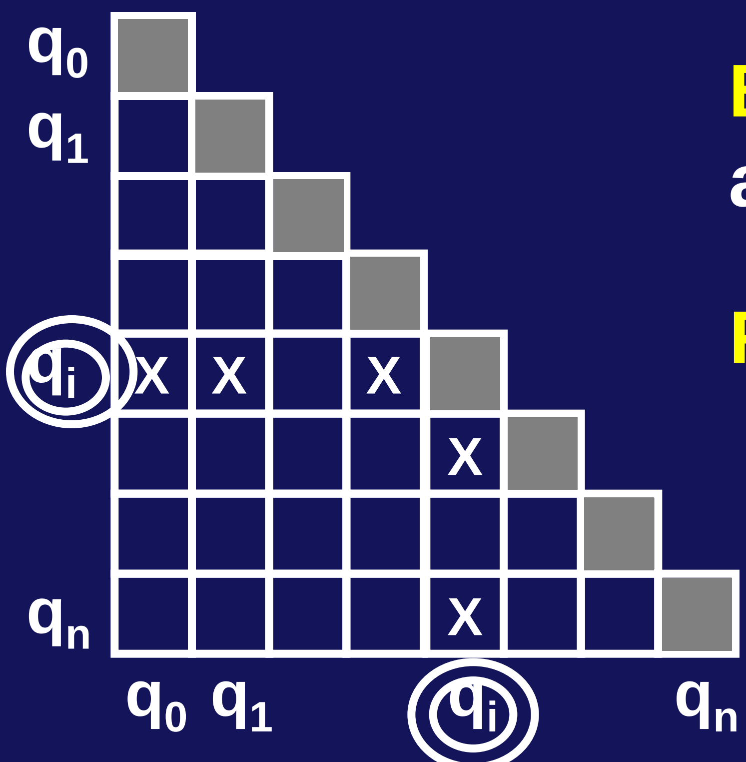
Base Case: p accepts and q rejects $\Rightarrow p \not\sim q$

TABLE-FILLING ALGORITHM

Input: DFA M = (Q, Σ, δ, q_0, F)

Output: (1) $D_M = \{ (p,q) \mid p,q \in Q \text{ and } p \not\equiv q \}$

(2) $E_M = \{ [q] \mid q \in Q \}$



Base Case: p accepts and q rejects $\Rightarrow p \not\equiv q$

Recursion:

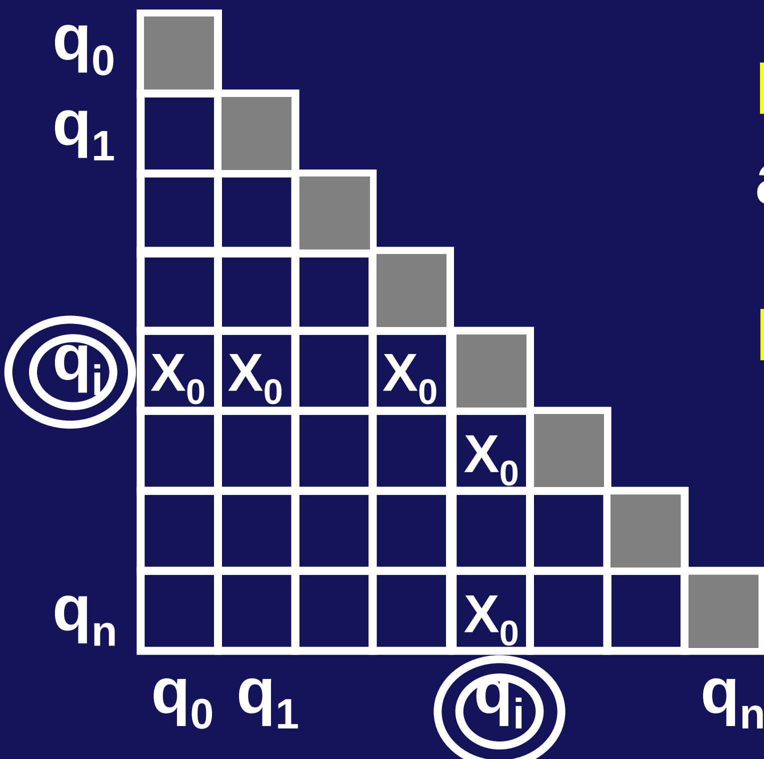
$$\begin{array}{l} p \xrightarrow{\sigma} p' \\ \qquad\quad \not\equiv \quad \Rightarrow p \not\equiv q \\ q \xrightarrow{\sigma} q' \end{array}$$

TABLE-FILLING ALGORITHM

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

Output: (1) $D_M = \{ (p,q) \mid p,q \in Q \text{ and } p \not\sim q \}$

(2) $E_M = \{ [q] \mid q \in Q \}$

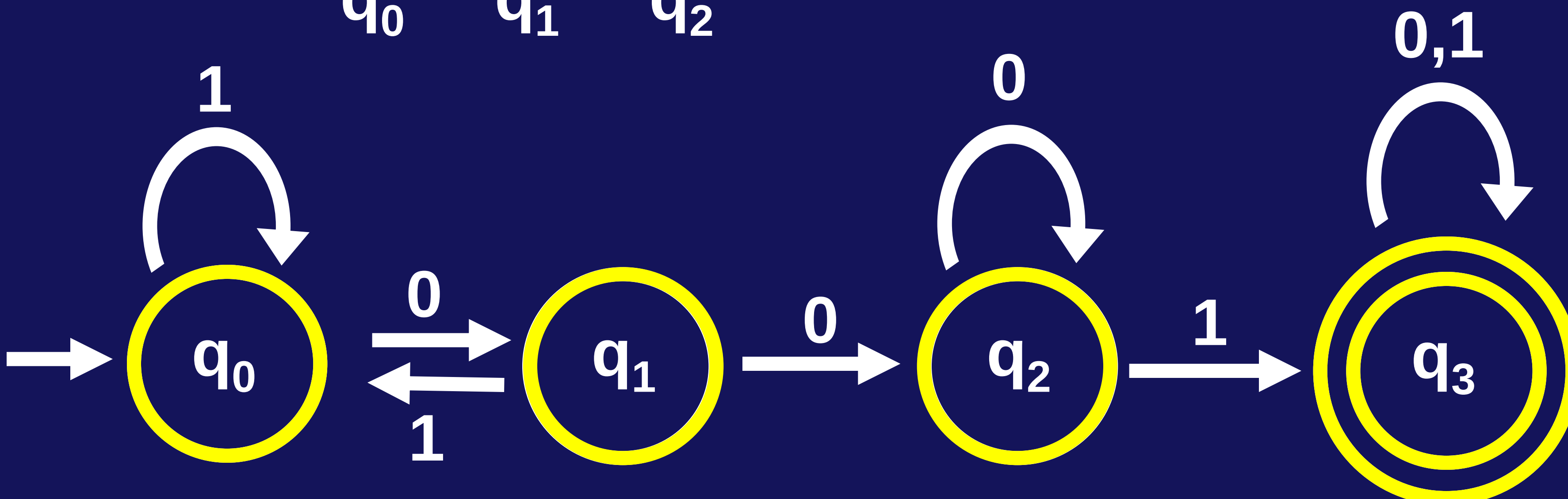


Base Case: p accepts and q rejects $\Rightarrow p \not\sim q$

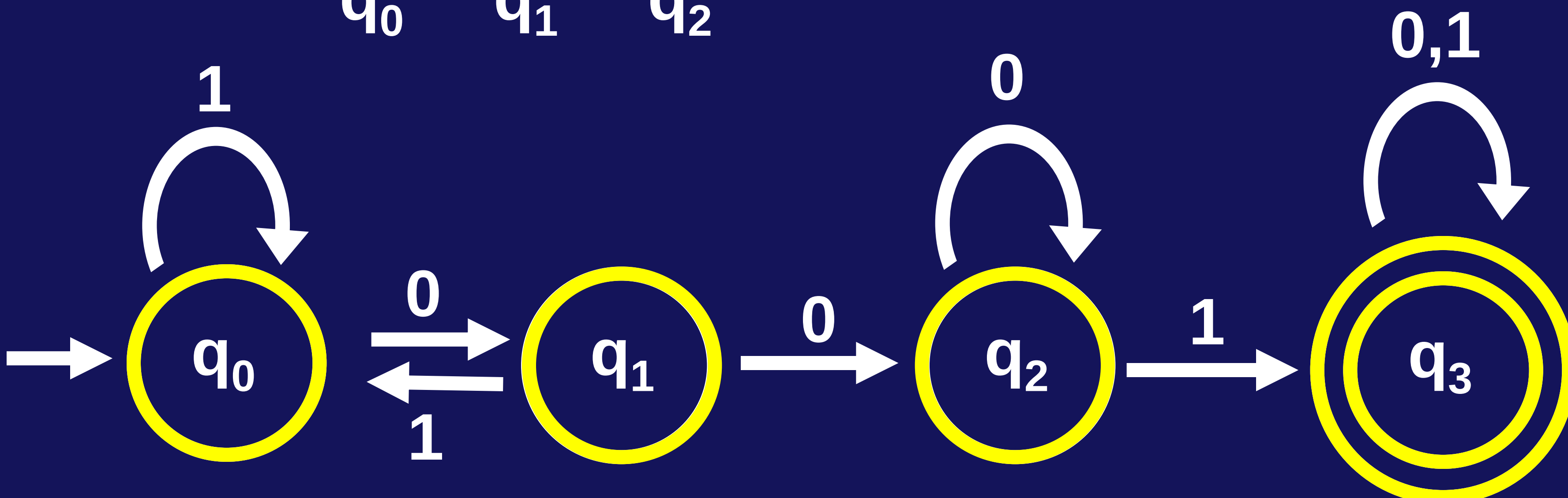
Recursion:

$$p \xrightarrow{\sigma} p'$$
$$\not\sim \quad \Rightarrow p \not\sim q$$
$$q \xrightarrow{\sigma} q'$$

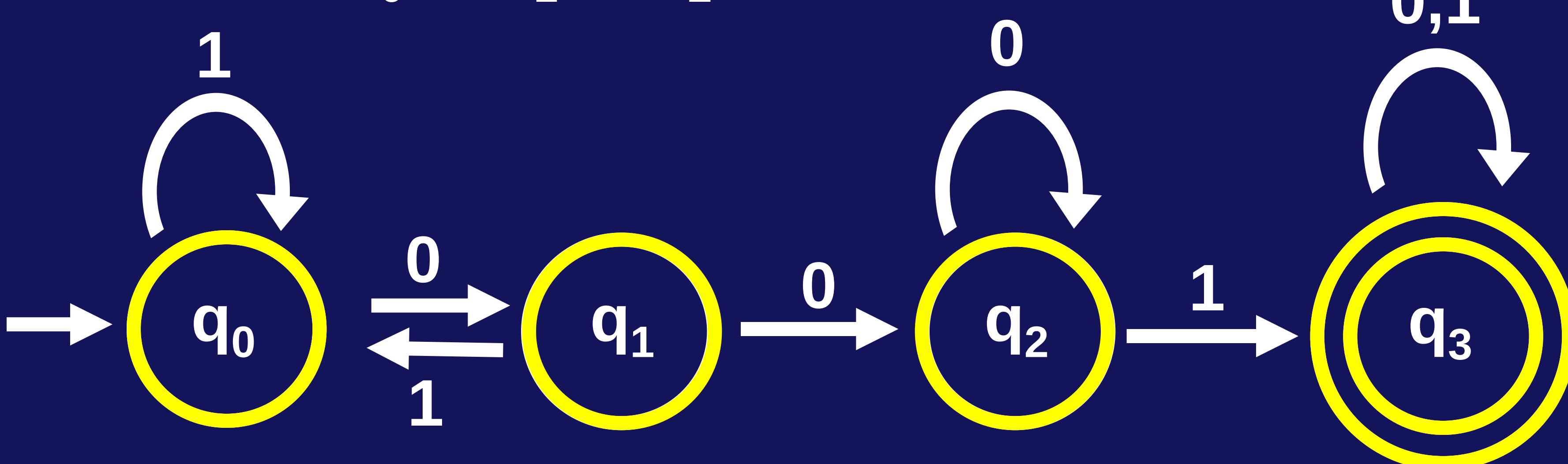
	q_0	q_1	q_2
q_1			
q_2			
q_3	X_0	X_0	X_0

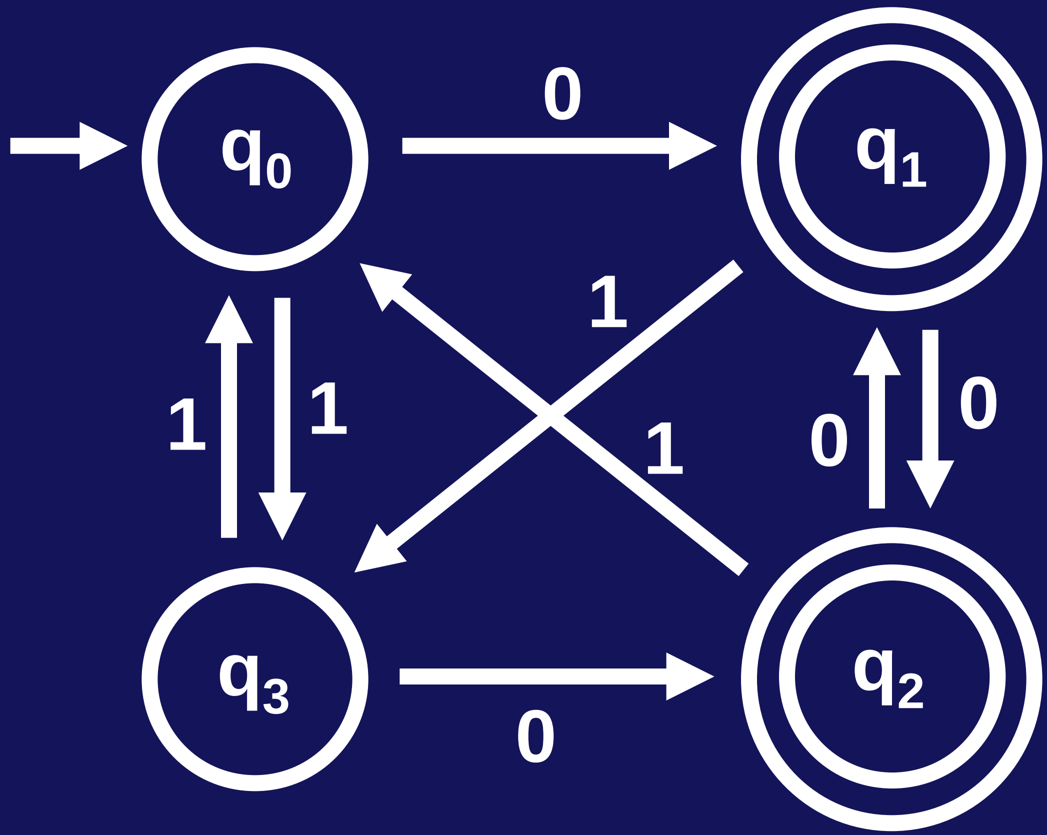


q_1			
q_2	X_1	X_1	
q_3	X_0	X_0	X_0
	q_0	q_1	q_2

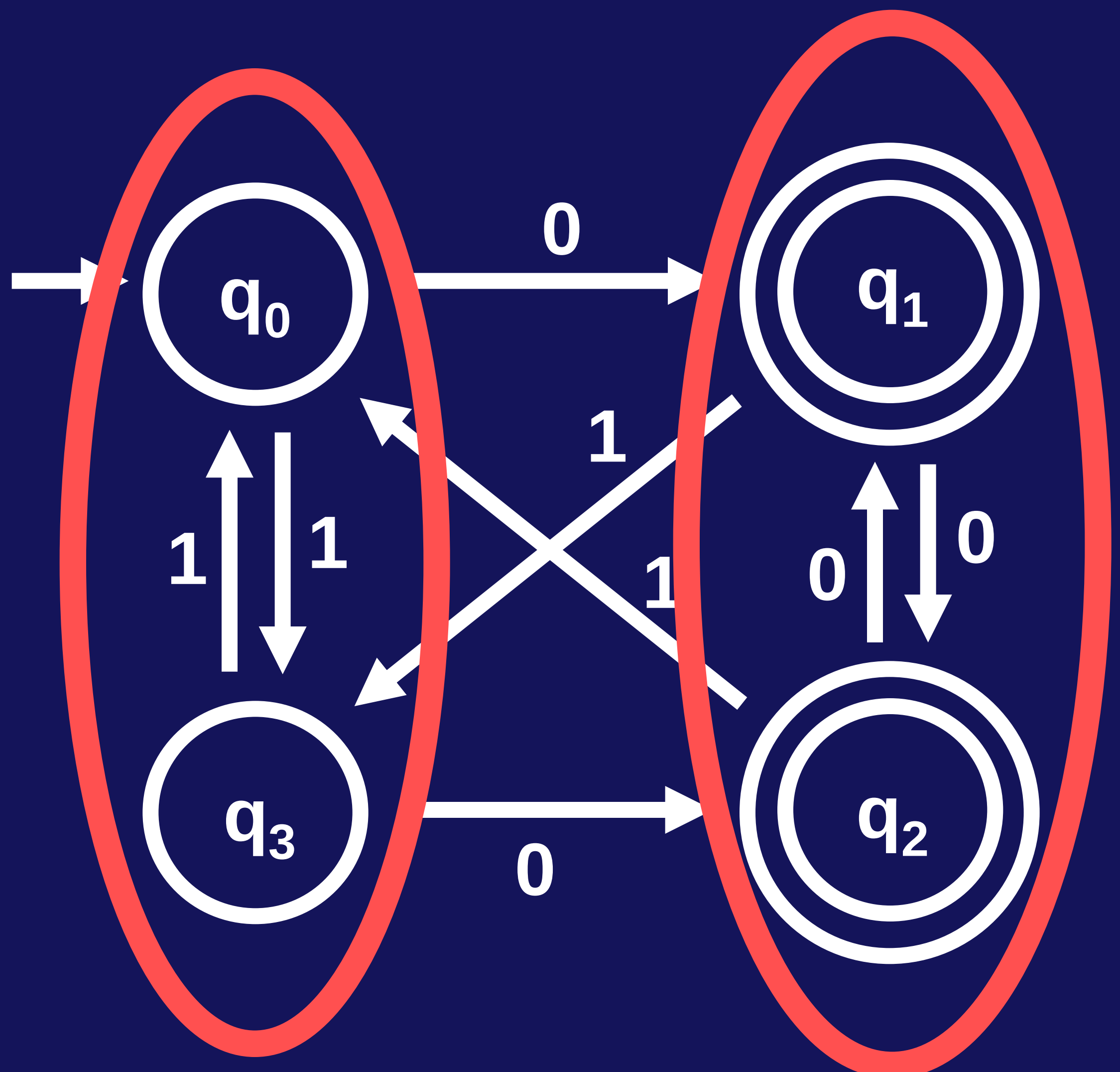


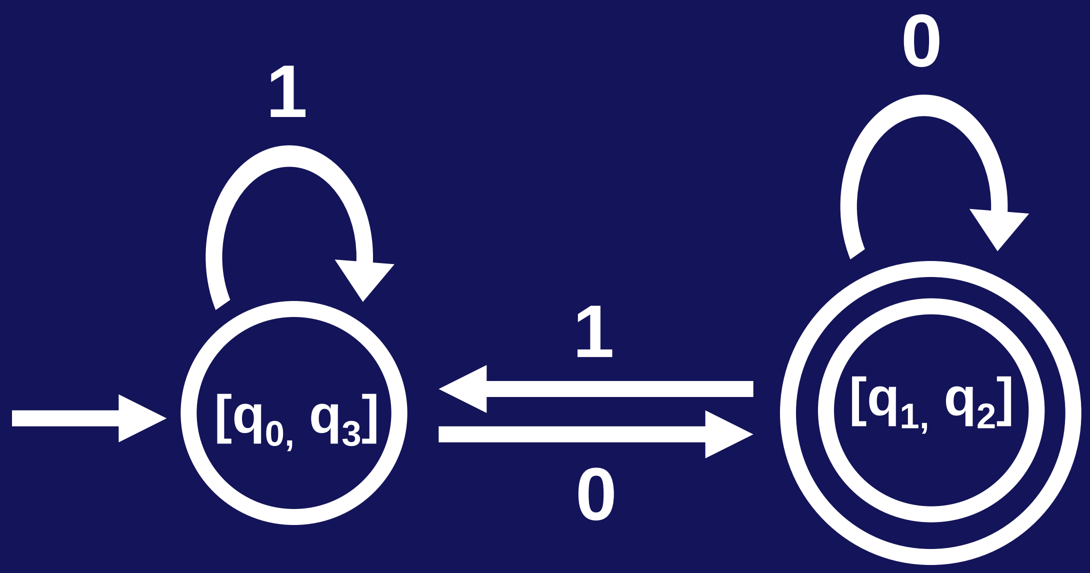
	q_0	q_1	q_2
q_1	X_2		
q_2	X_1	X_1	
q_3	X_0	X_0	X_0



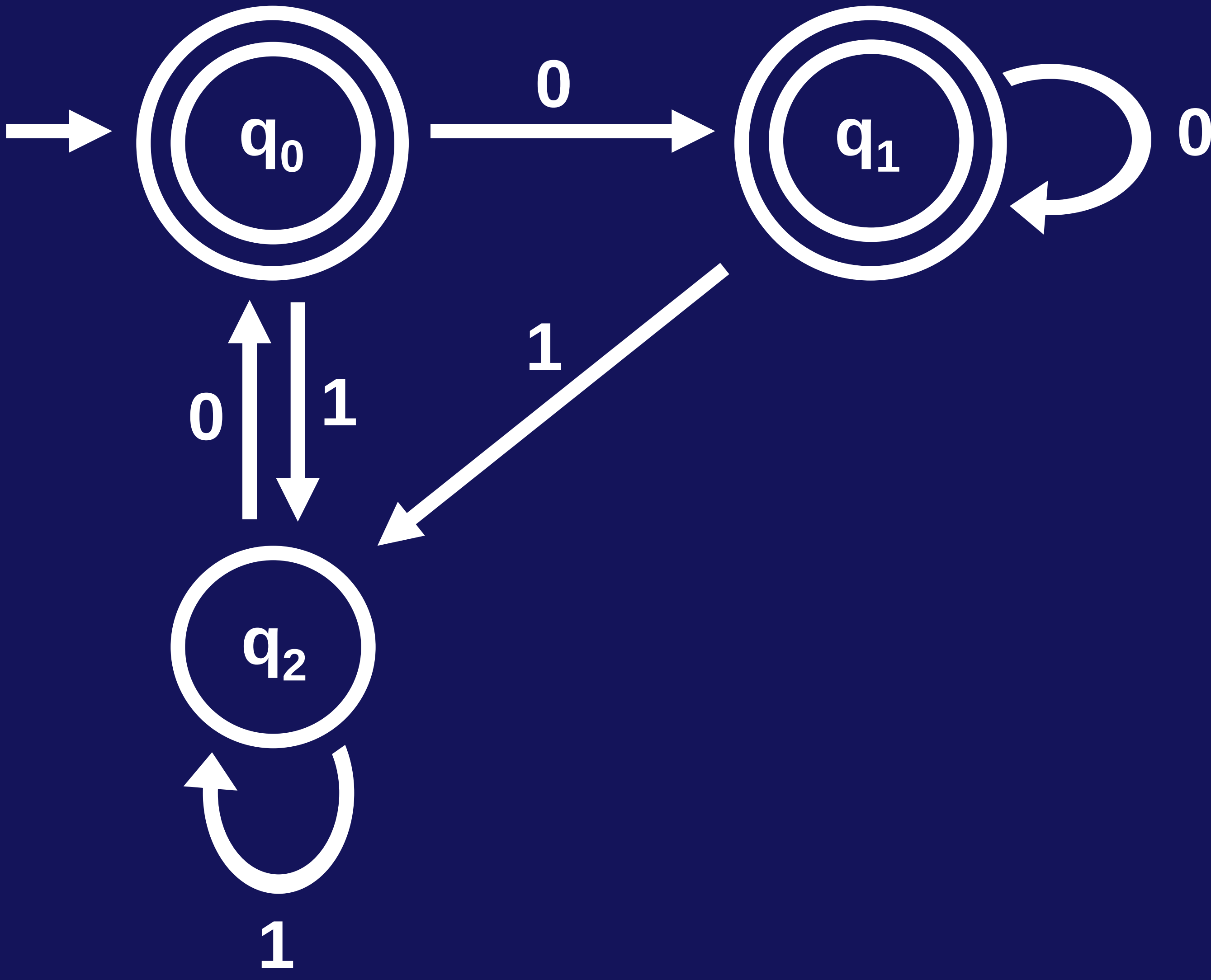


	q_0	q_1	q_2
q_1	X_0		
q_2	X_0		
q_3		X_0	X_0

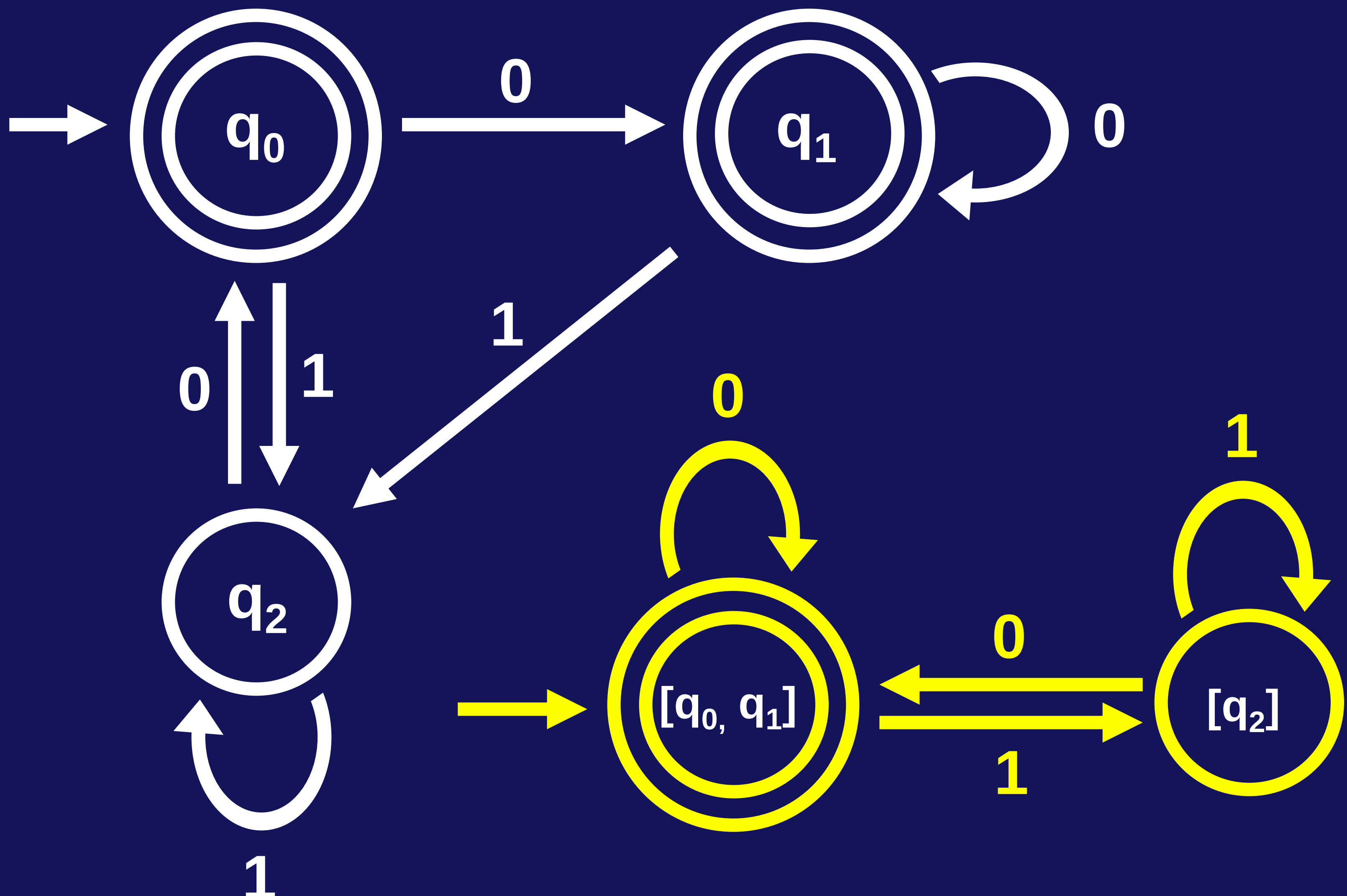

q0
q1
q2
q3
0
1
1
1
1
0
0
0

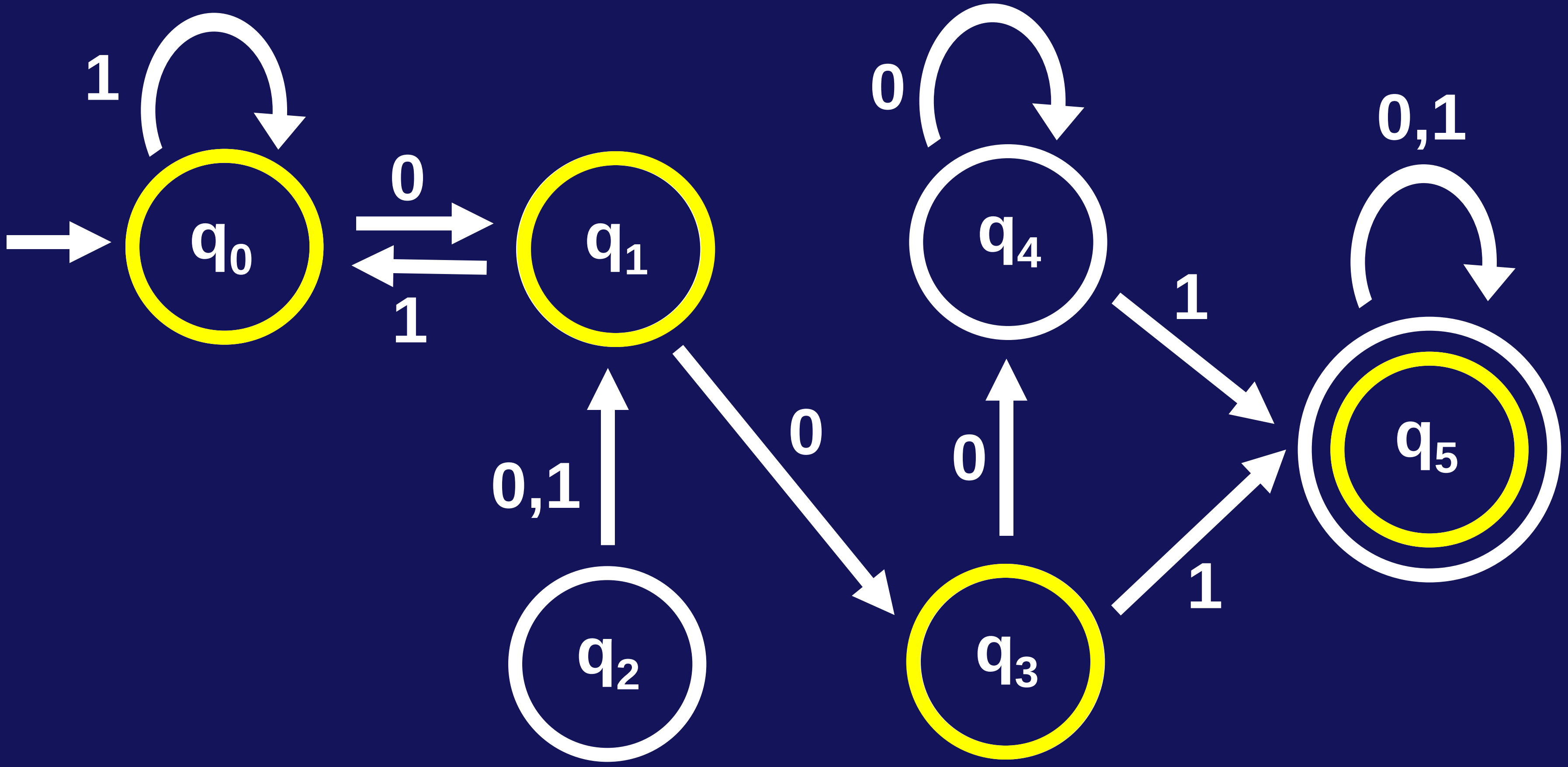

1
0
1
[q0, q3]
0
[q1, q2]

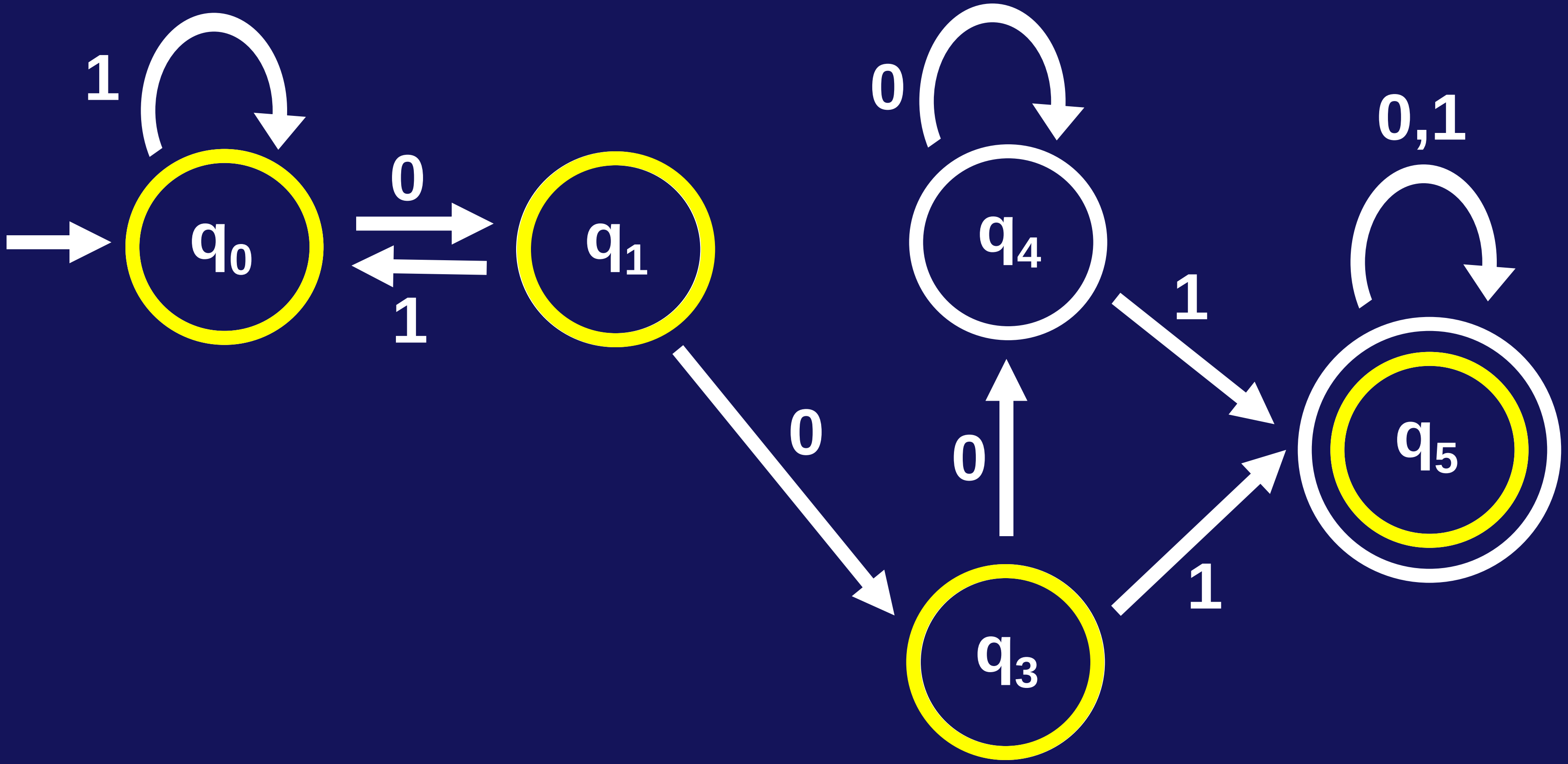
MINIMIZE

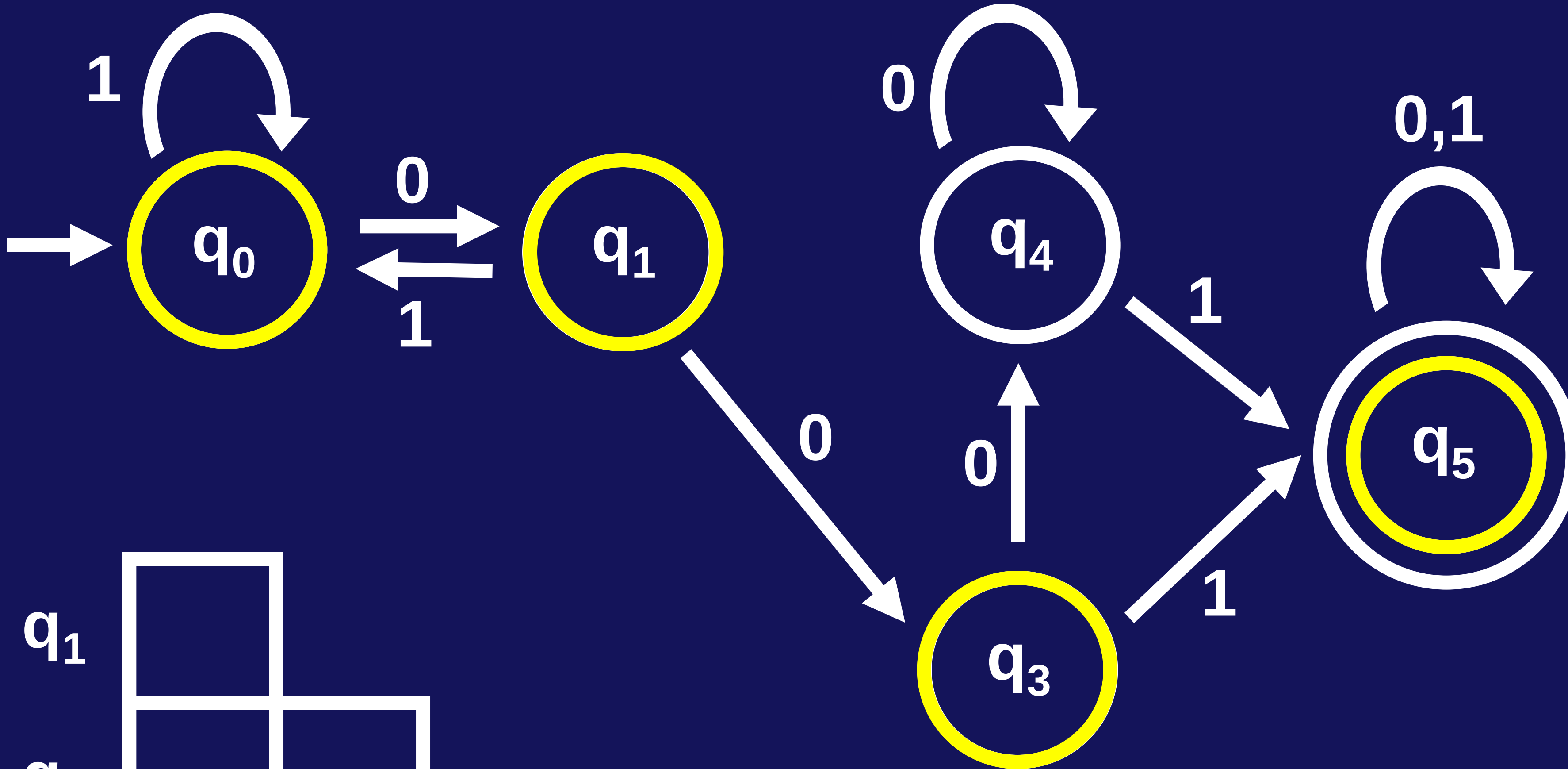


MINIMIZE

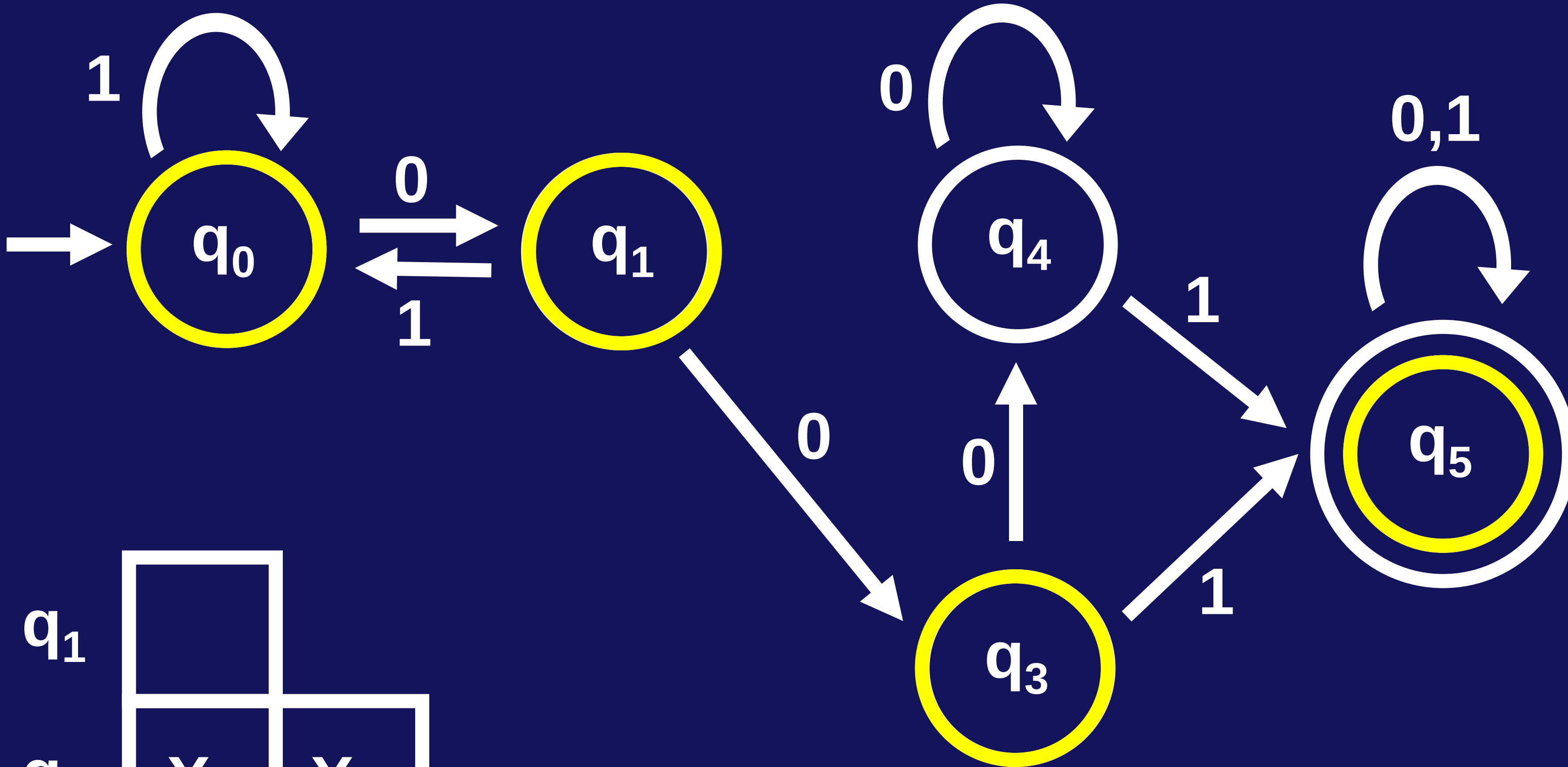



1
0
1
0
0,1
0
0
1
0,1
1
q0
q1
q2
q3
q4
q5

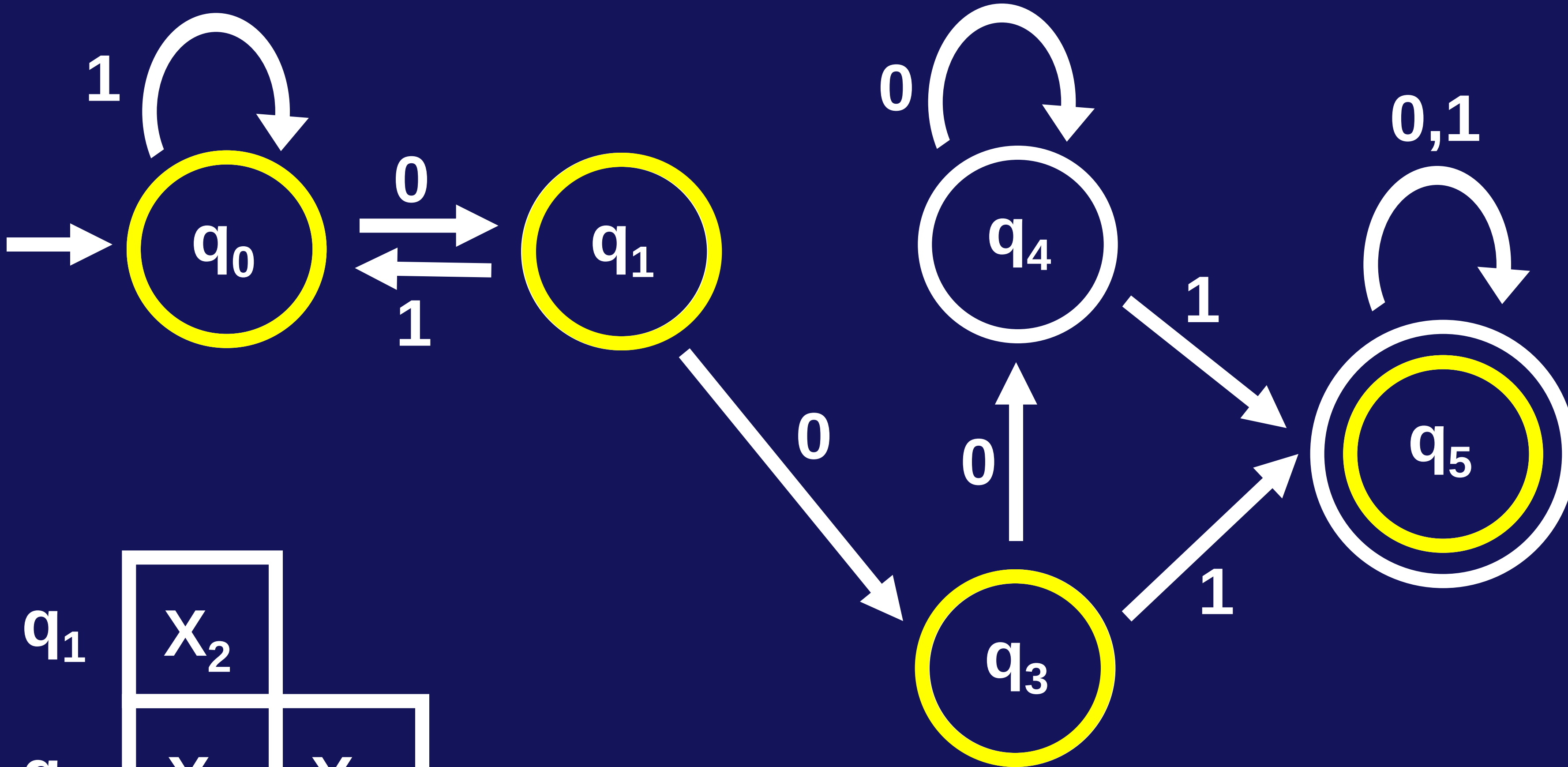

1
0
1
q0
q1
0
q4
0
1
0
0,1
q5
q3
1



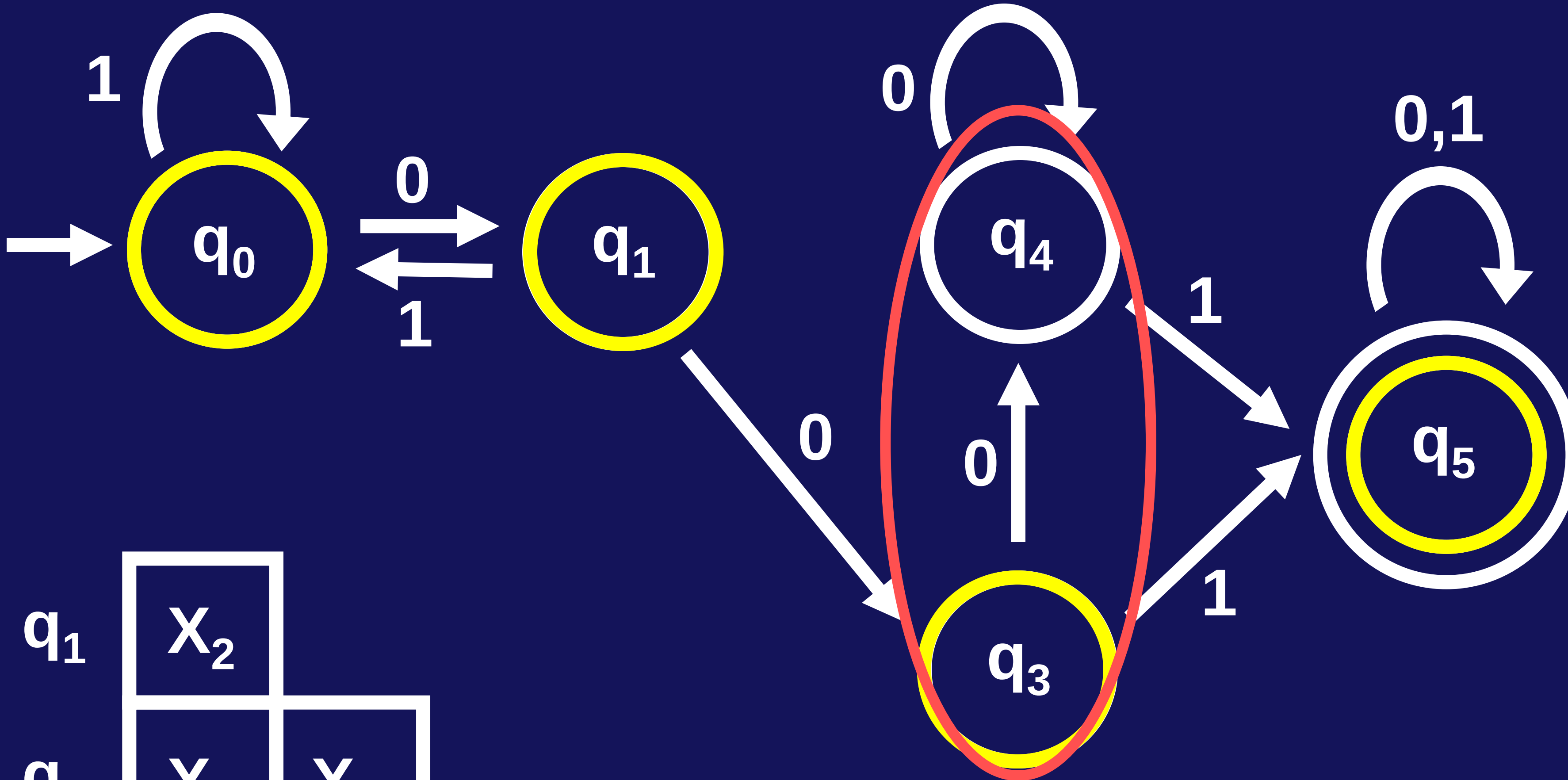
	q_0	q_1	q_3	q_4
q_1				
q_3				
q_4				
q_5	X_0	X_0	X_0	X_0



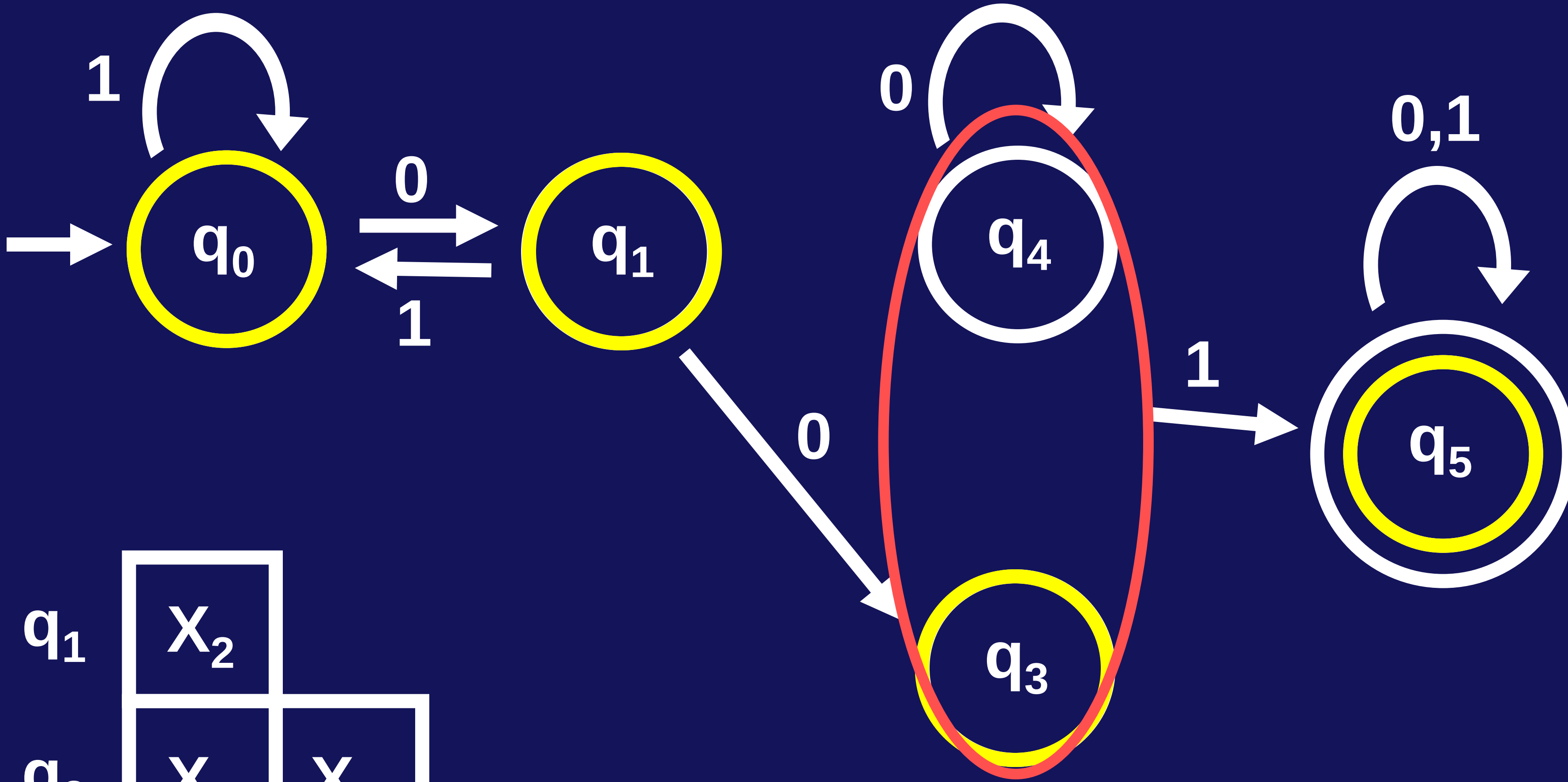
	q_0	q_1	q_3	q_4
q_1				
q_3	X_1	X_1		
q_4	X_1	X_1		
q_5	X_0	X_0	X_0	X_0



	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

HOW TO PROVE THAT TWO DFAs ARE EQUIVALENT