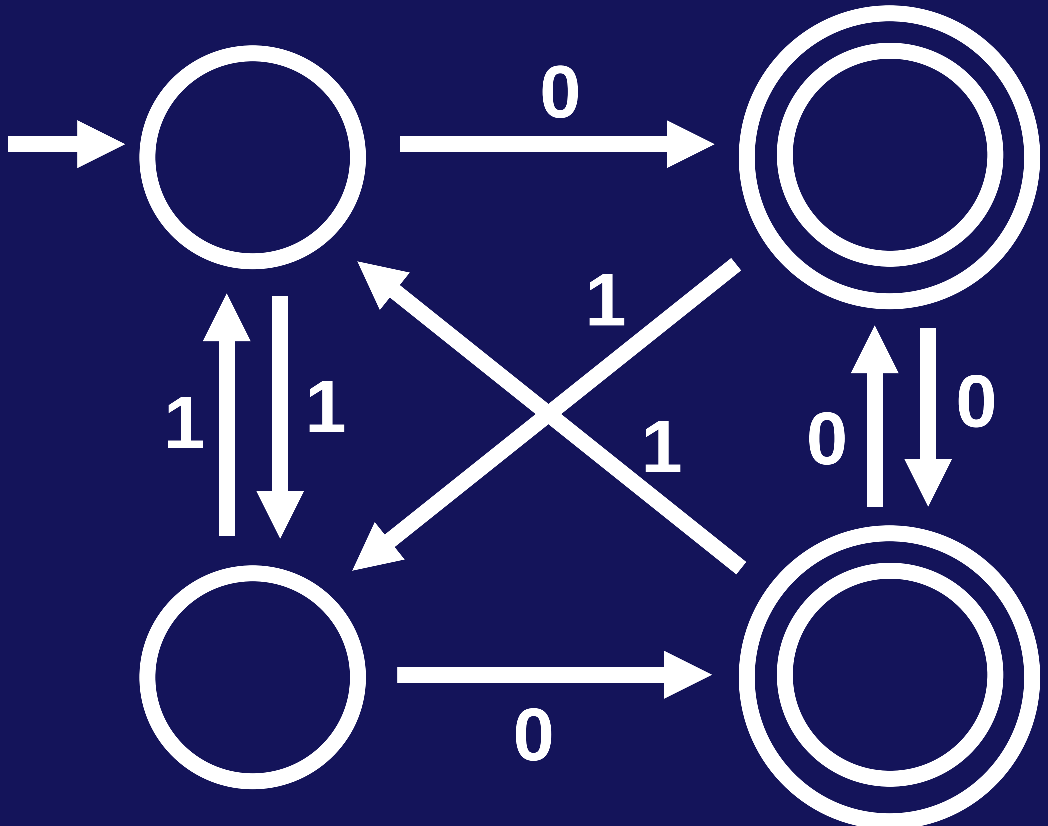
MINIMIZING DFAs

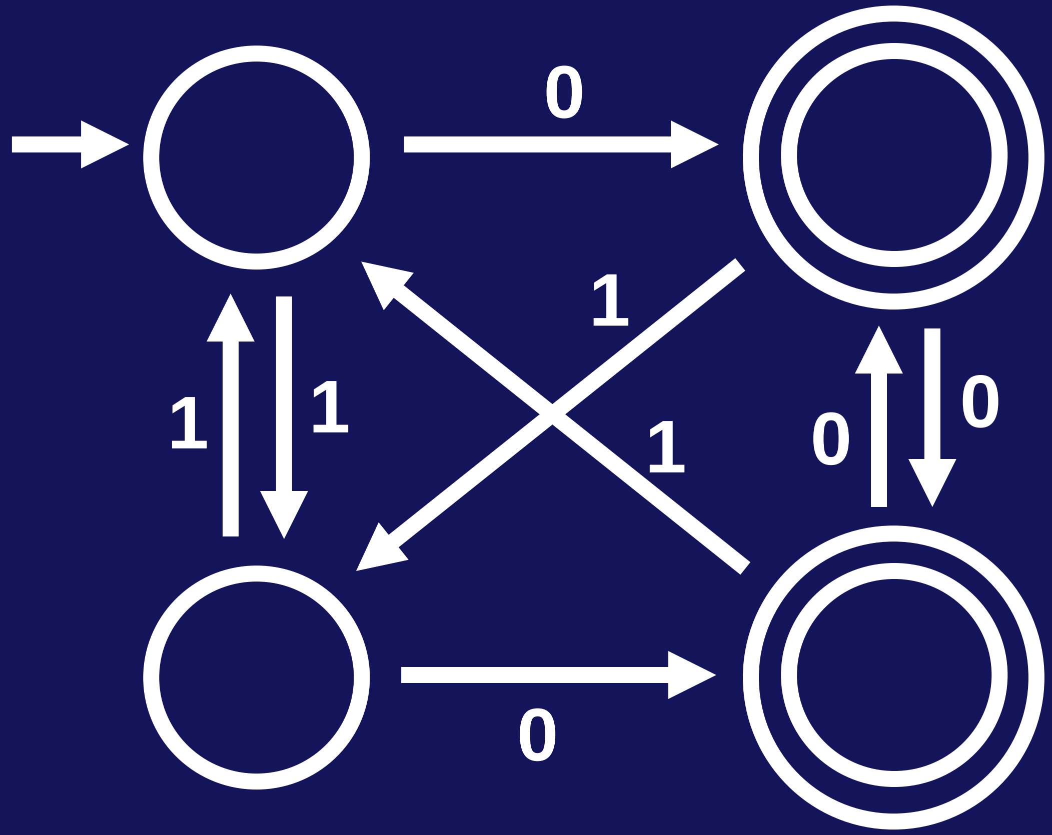
To have minimum number of states

IS THIS **MINIMAL?**

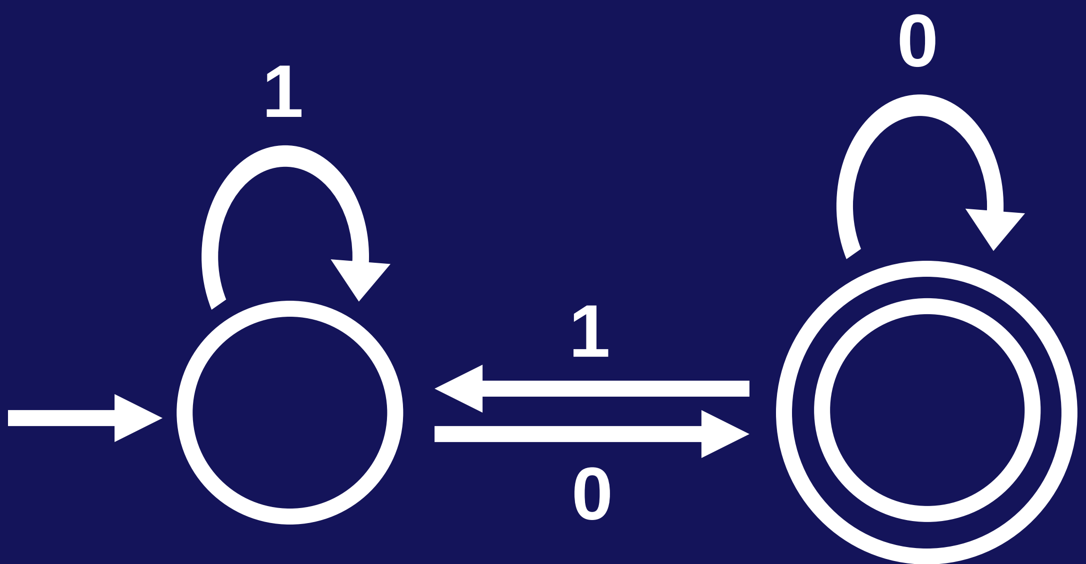


IS THIS **MINIMAL?**

NO



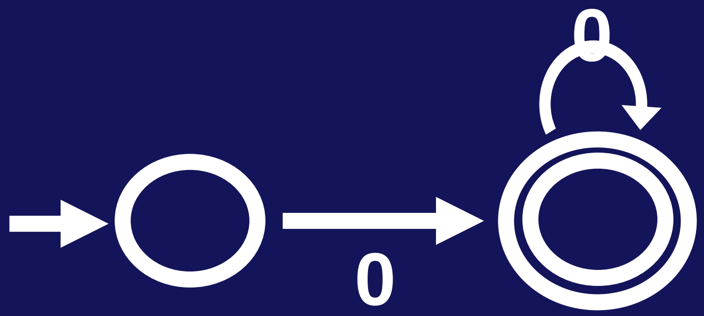
IS THIS **MINIMAL?**

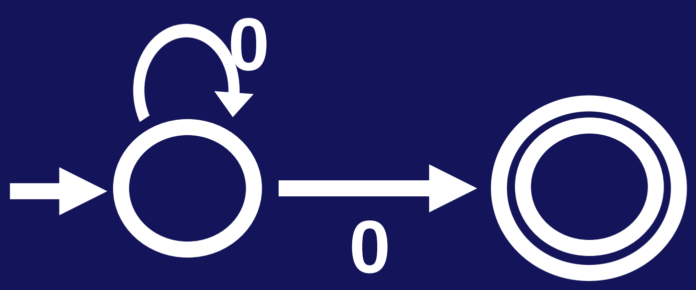


THEOREM

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 δ

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

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

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

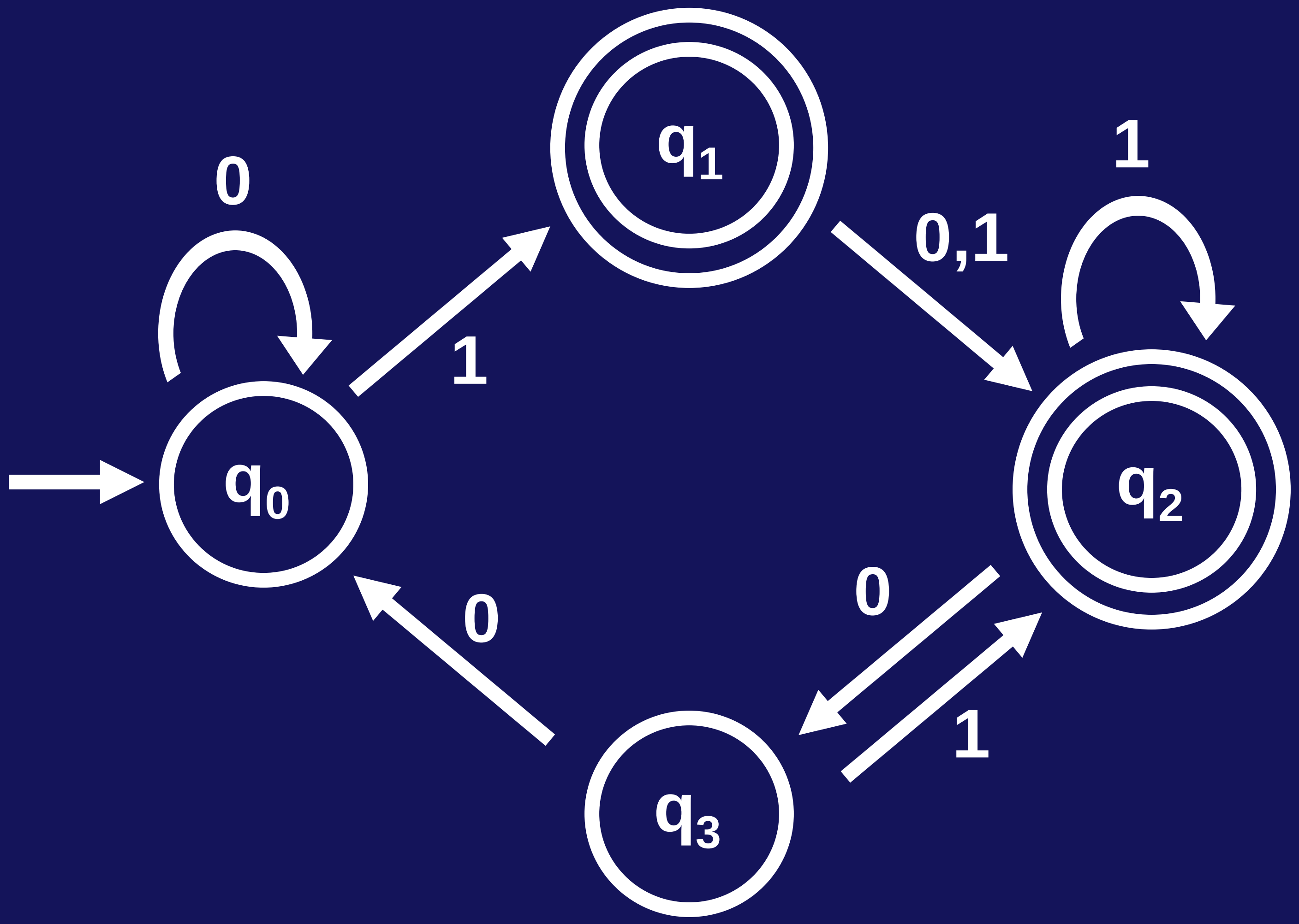
$$\hat{\delta}(q, a) = \delta(q, a) \text{ where } a \in \Sigma$$

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

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

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



ε distinguishes accept from non-accept states

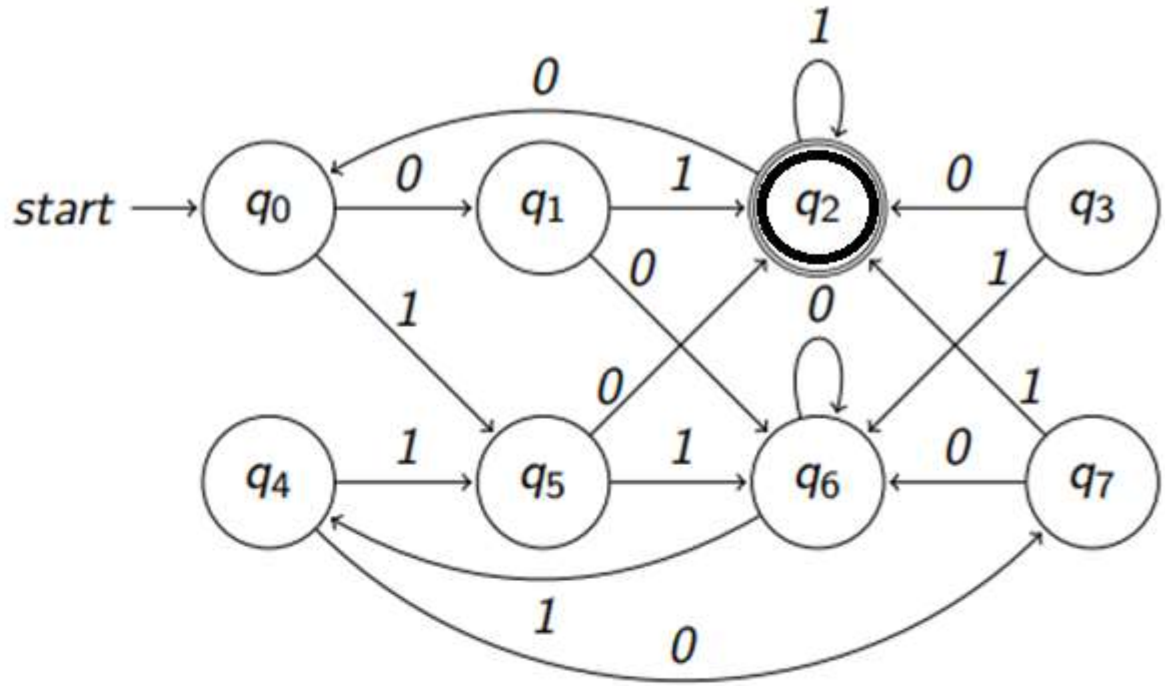
Let M = (Q, Σ, δ, q_0, F) and let p, q, r $\in$ Q

Definition **:**

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

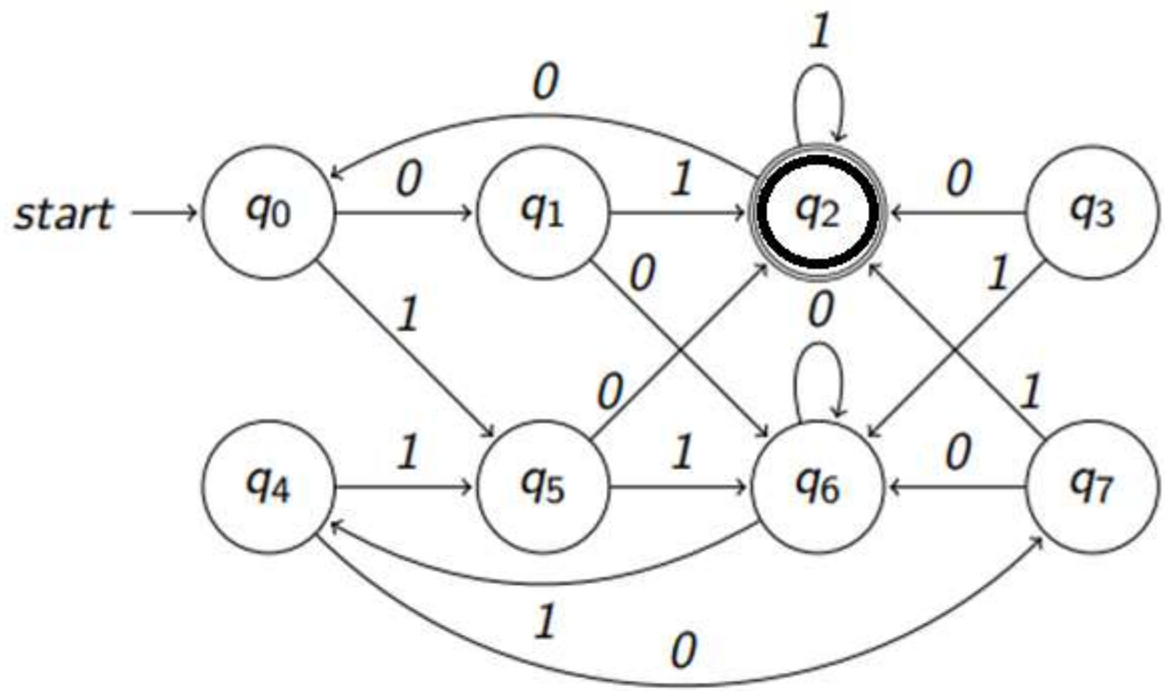
(2) Otherwise p is not equivalent to q. In this case, we say p and q are distinguishable.

Example: distinguishable states



- ϵ *distinguishes* q_2 *and* q_6.
- 01 *distinguishes* q_0 *and* q_6.

Example: distinguishable states



- ϵ *distinguishes* q_2 *and* q_6.
- 01 *distinguishes* q_0 *and* q_6.

Exercise

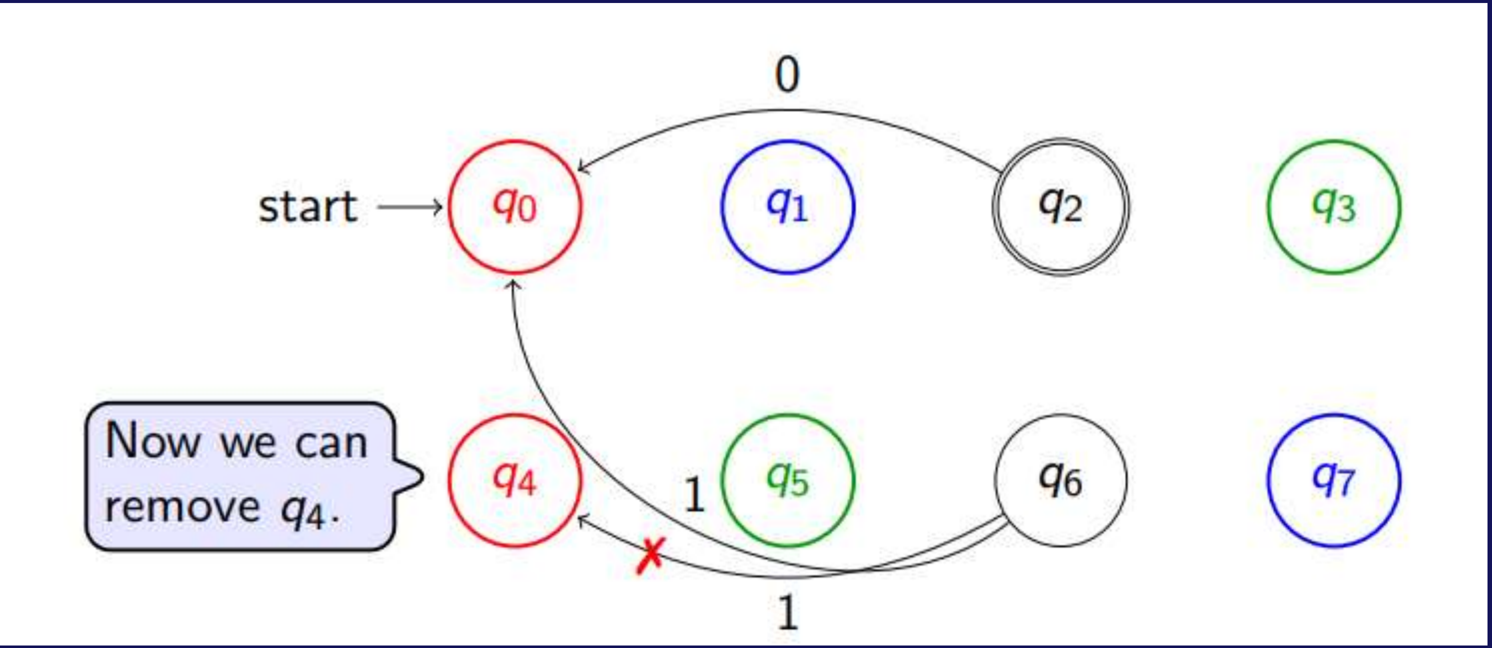
Give strings that distinguishes the following pair of states.

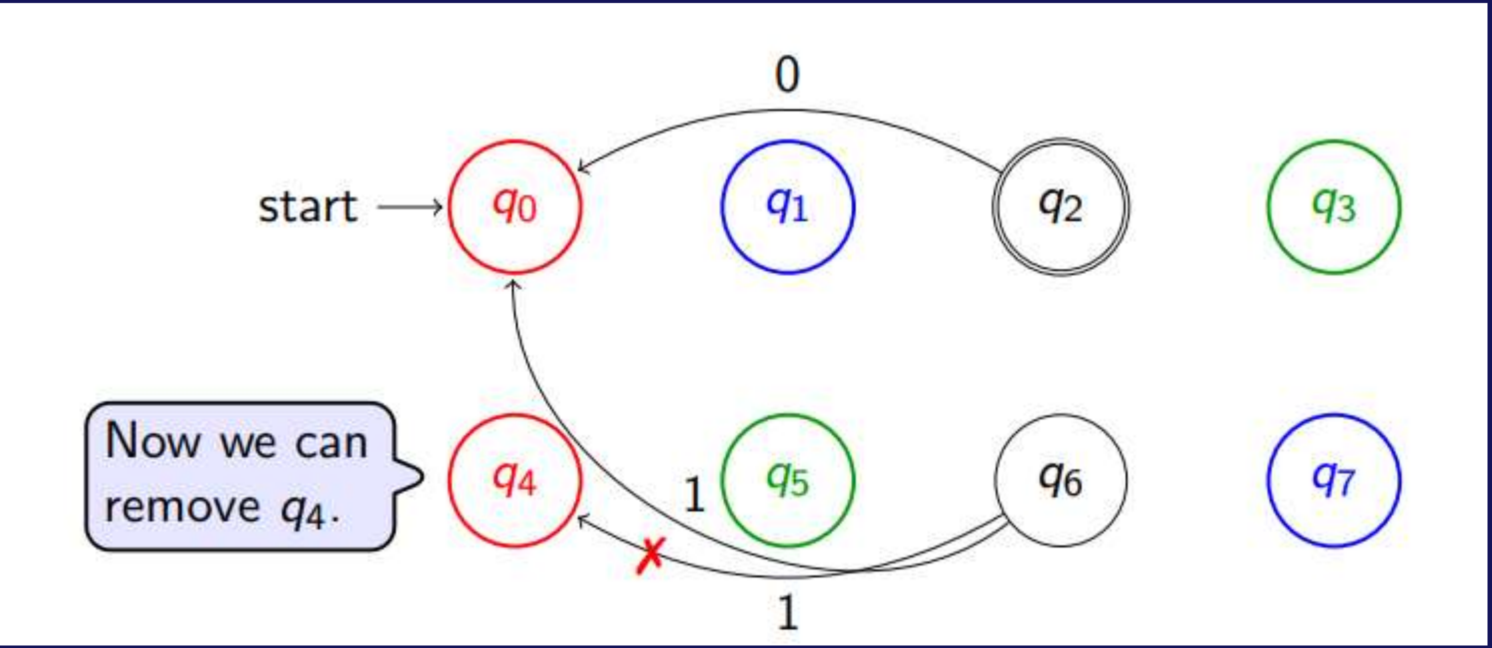
- q_1 *and* q_5
- q_4 *and* q_3
- q_2 *and* q_7
- q_1 *and* q_7

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.

Let q_0 *and* q_4 *are equivalent*





- But, how is that you know q_0 *and* q_4 *are equivalent*?

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

Define the relation "~":

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

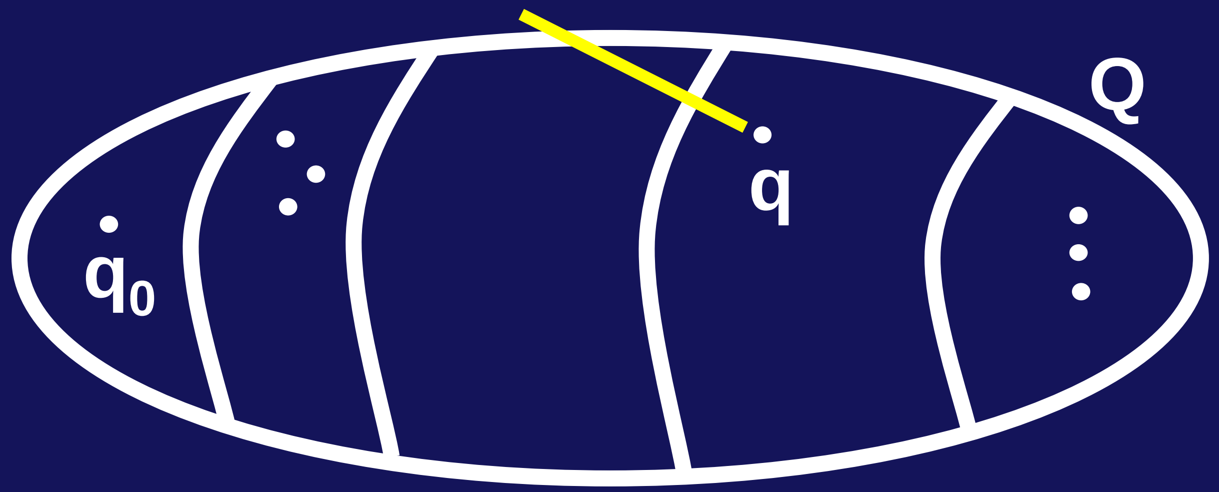
$p \sim q$ and $q \sim r \Rightarrow p \sim r$ (transitive)

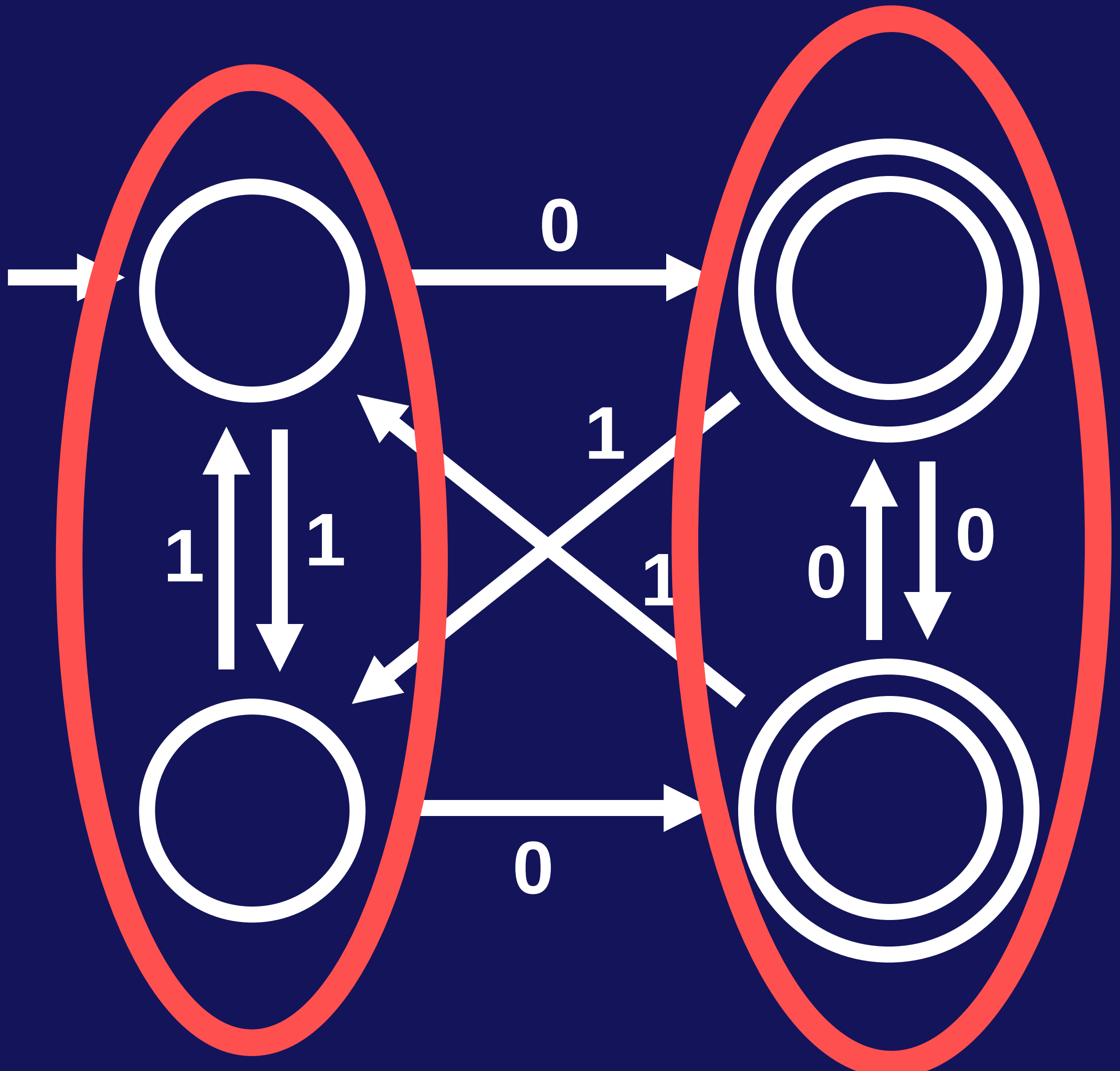
Let $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

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(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**

$\Updownarrow$

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
$E_M = \{ [q] \mid q \text{ is an accessible state of } M \}$

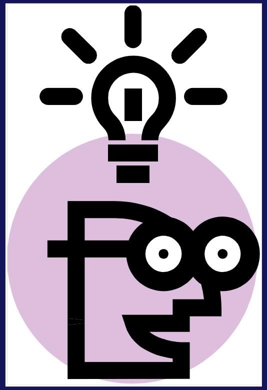
$$M_{MIN} = (Q_{MIN}, \Sigma, \delta_{MIN}, q_{0\ MIN}, F_{MIN})$$

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

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

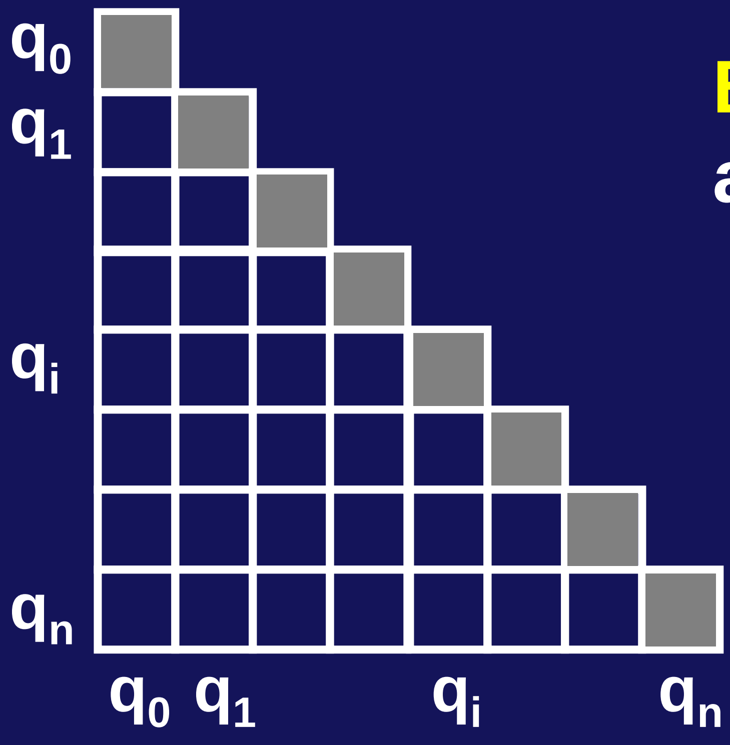
TABLE-FILLING ALGORITHM

IDEA!



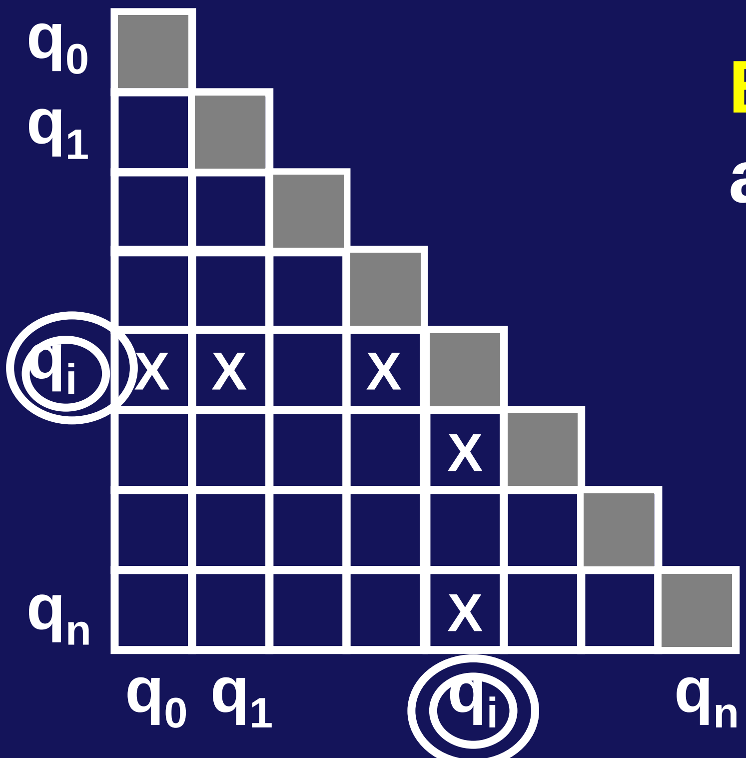
- **Make best effort to find pairs of states that are distinguishable.**
- **Pairs leftover will help us.**

TABLE-FILLING ALGORITHM



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

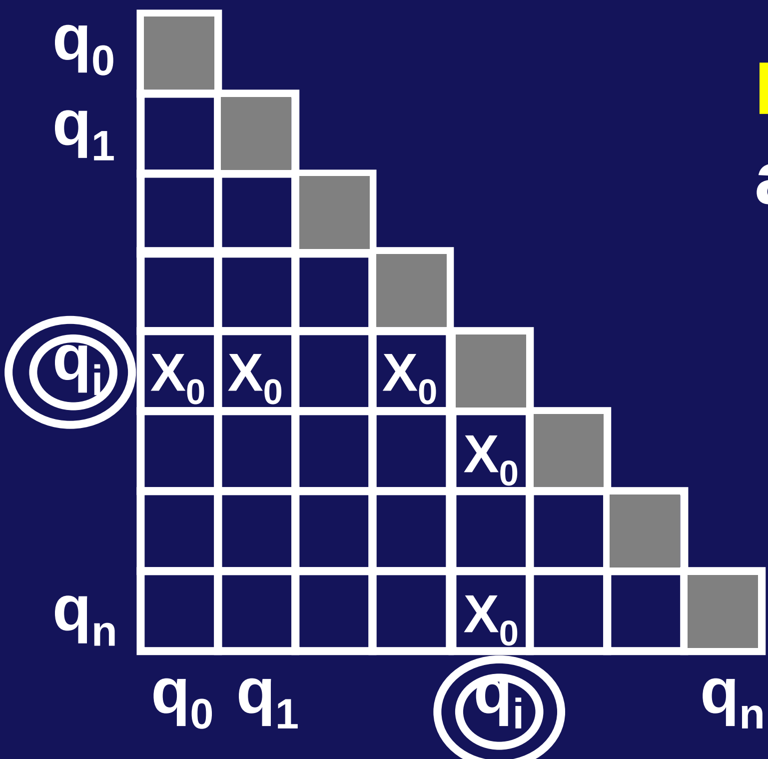
TABLE-FILLING ALGORITHM



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

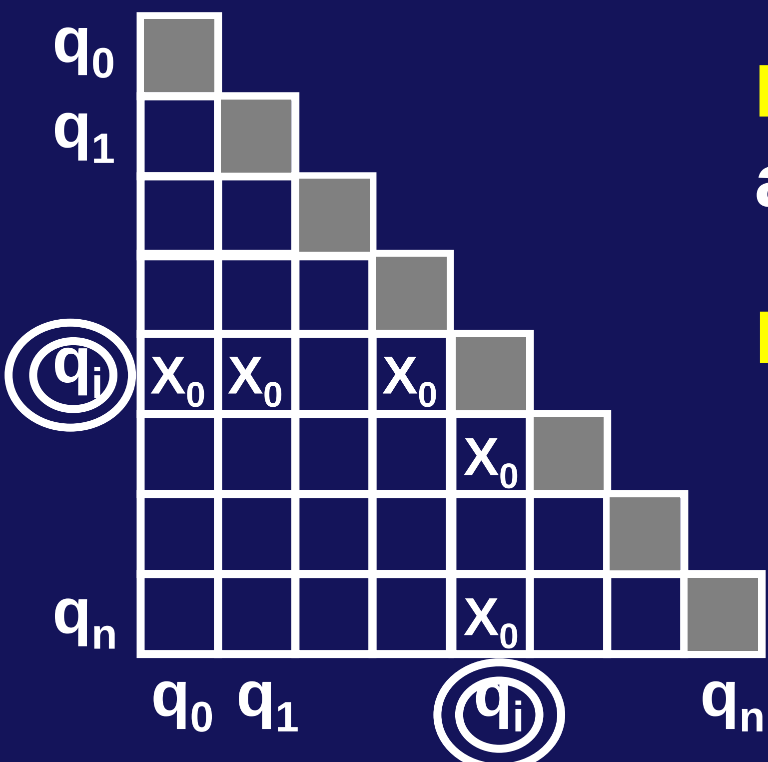
$p \xrightarrow{a} r$

$q \xrightarrow{a} s$

$r \not\sim s \Rightarrow p \not\sim q$

TABLE-FILLING ALGORITHM

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

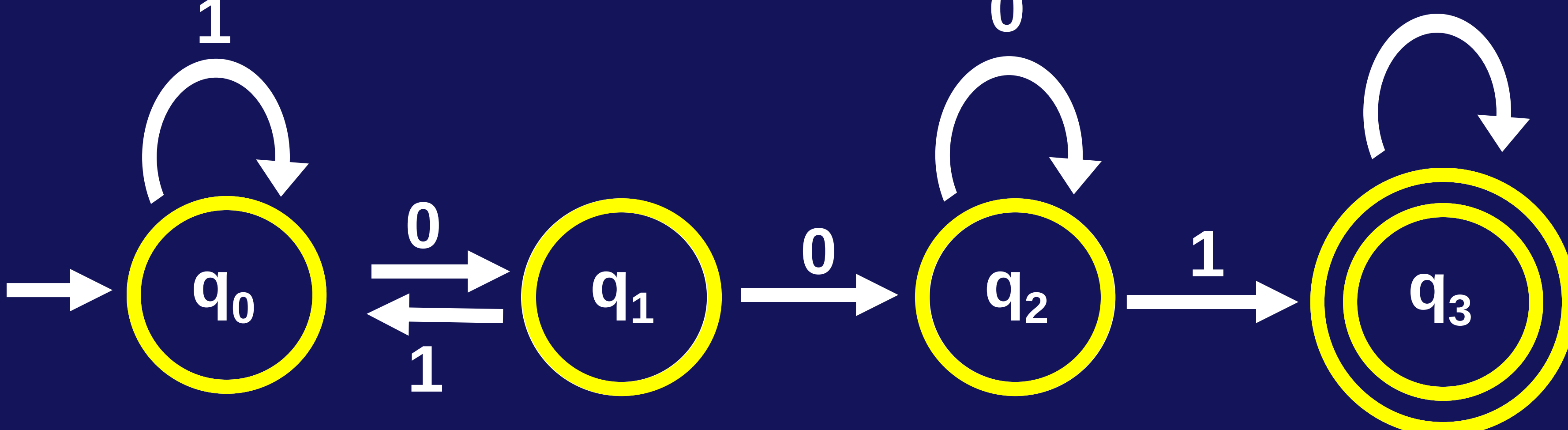


Base Case: p accepts and q rejects $\Rightarrow p \not\equiv q$

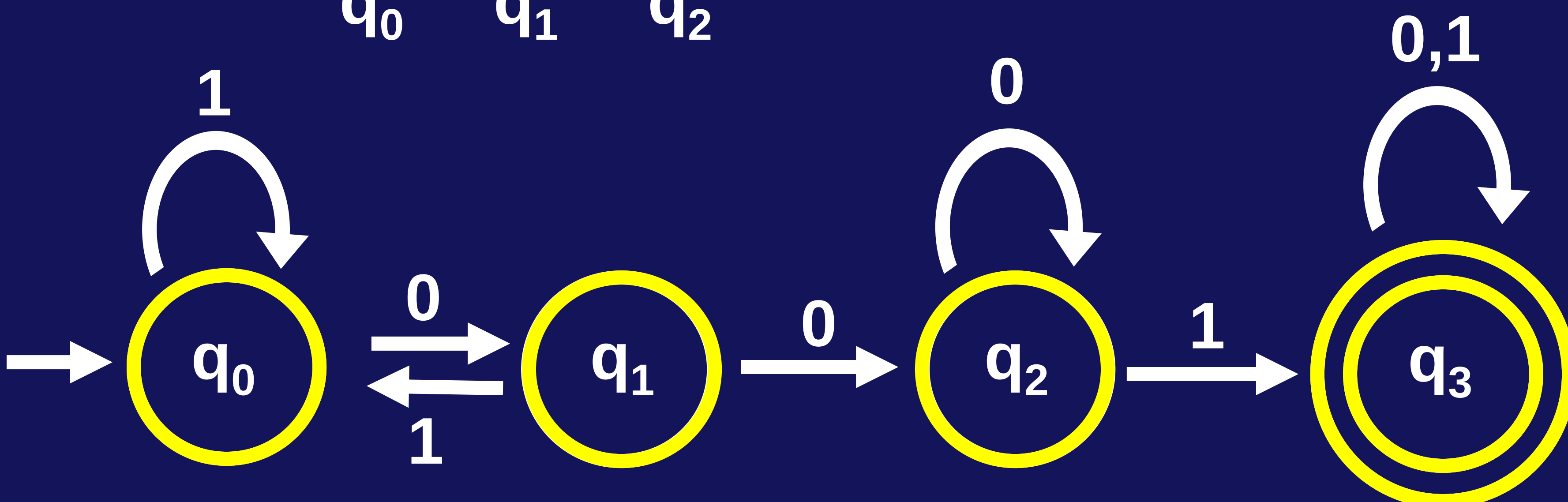
Recursion:

$$p \xrightarrow{a} r, \quad q \xrightarrow{a} s, \quad r \not\equiv s \Rightarrow p \not\equiv q$$

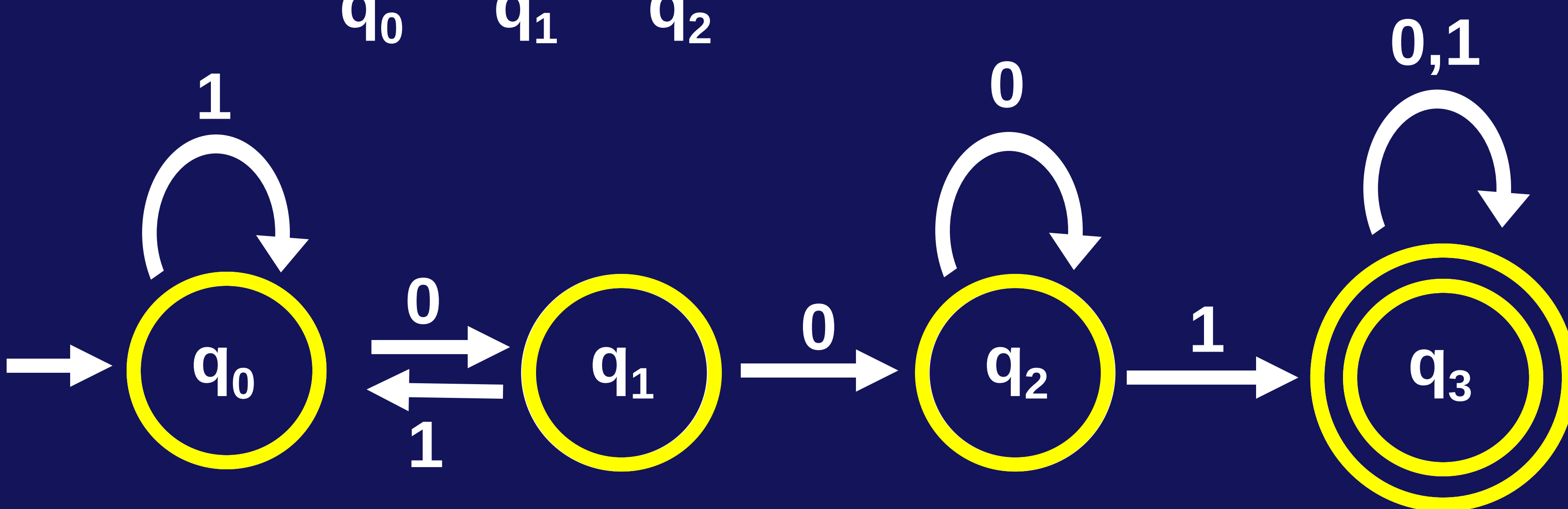
q_1			
q_2			
q_3	X_0	X_0	X_0
	q_0	q_1	q_2



	q_0	q_1	q_2
q_1			
q_2	X_1	X_1	
q_3	X_0	X_0	X_0

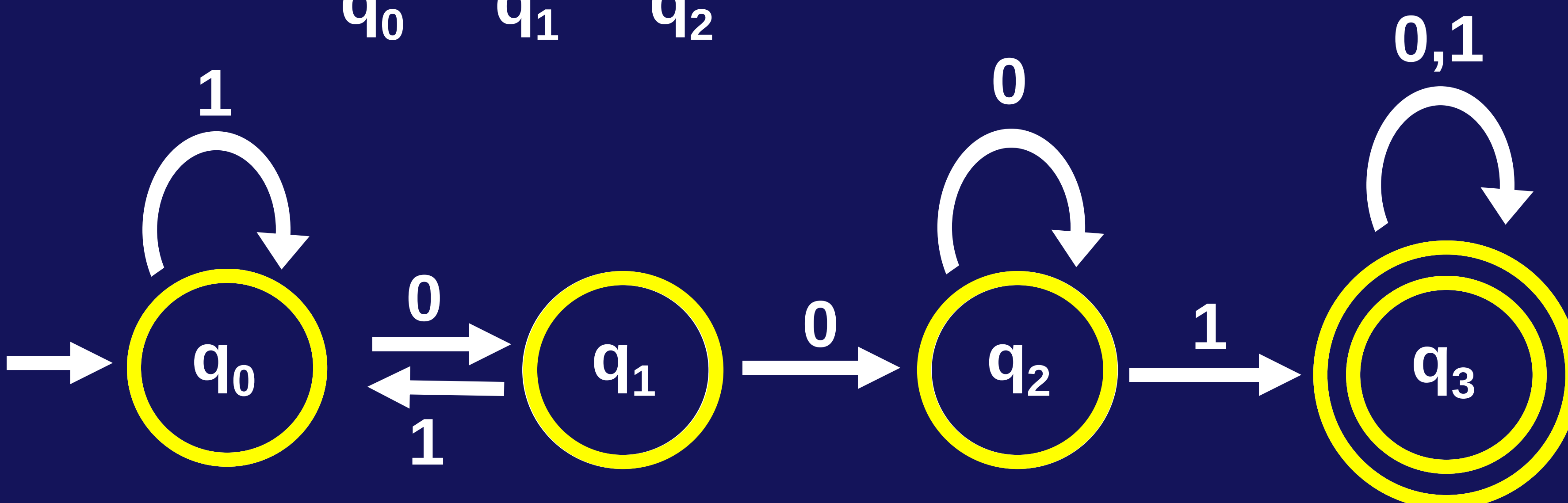


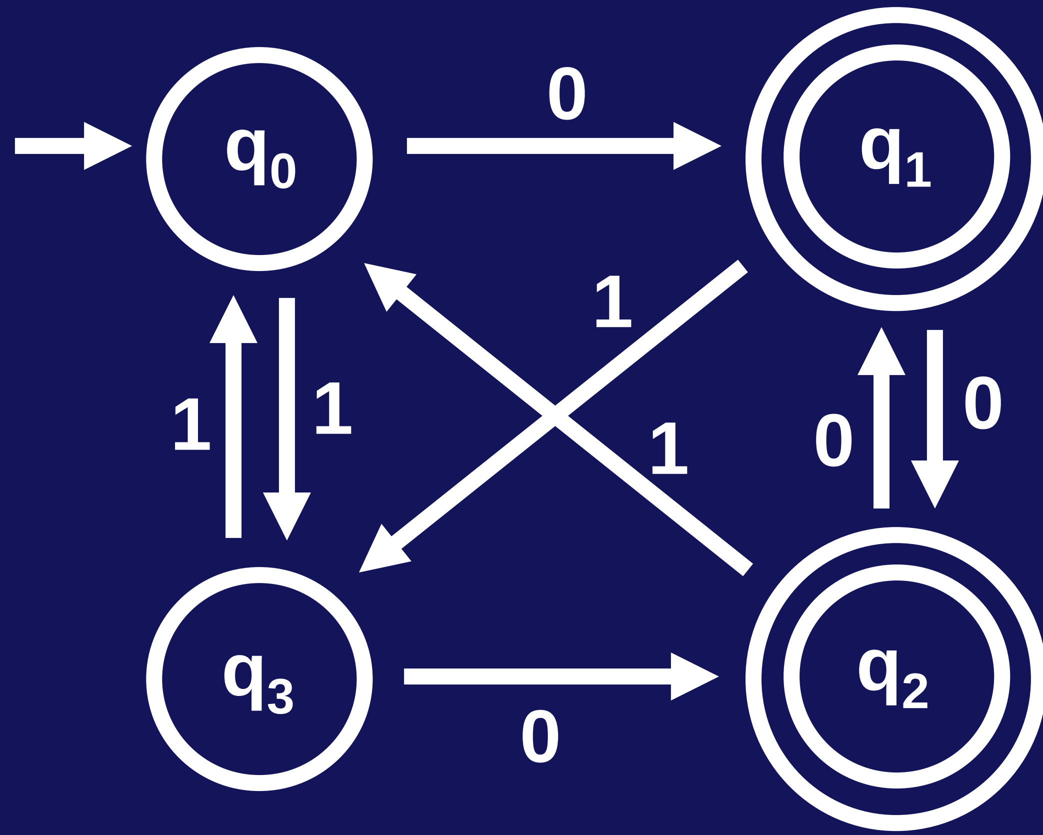
	q_0	q_1	q_2
q_1	X_2		
q_2	X_1	X_1	
q_3	X_0	X_0	X_0



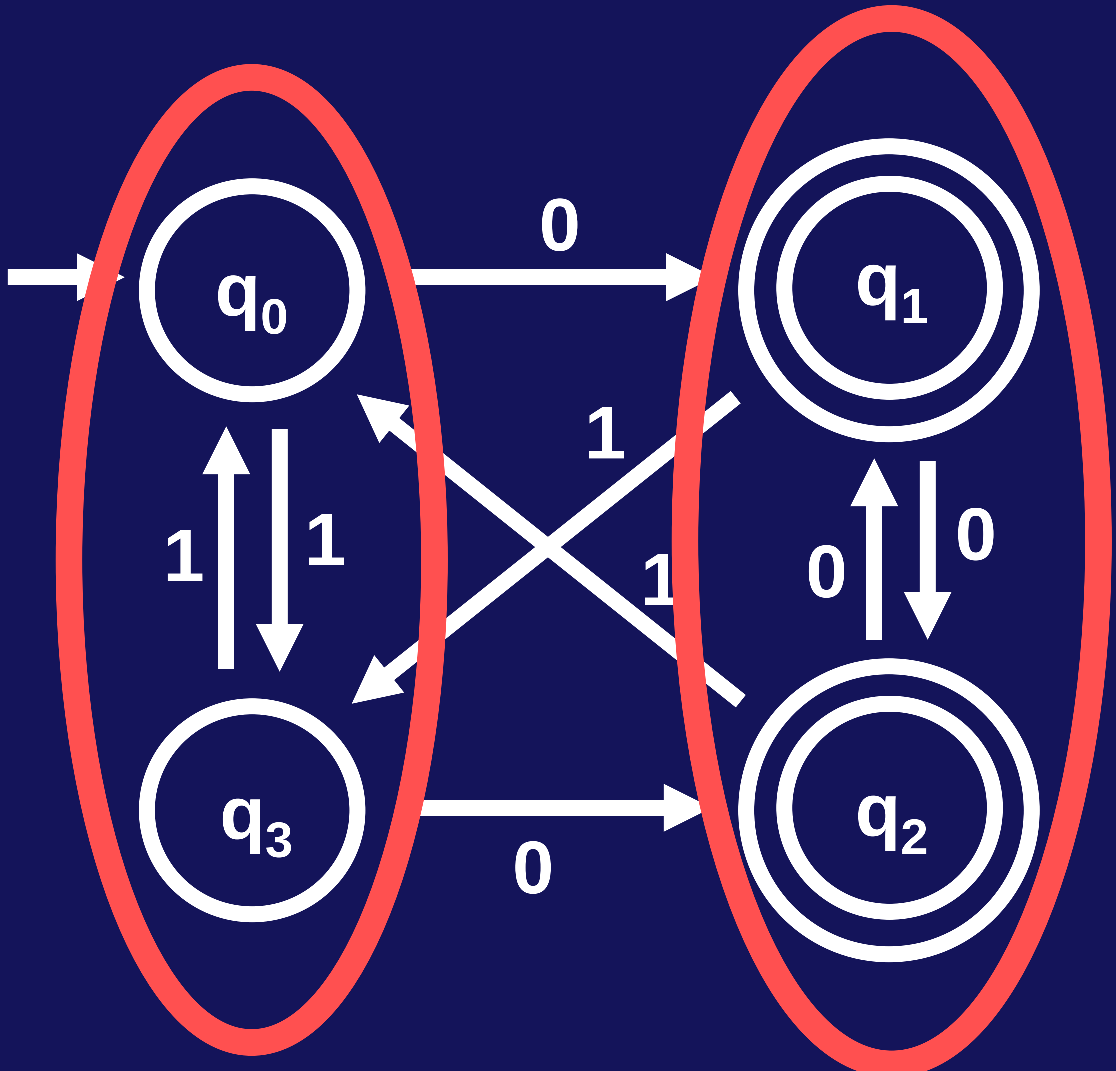
	q_0	q_1	q_2
q_1	X_2		
q_2	X_1	X_1	
q_3	X_0	X_0	X_0

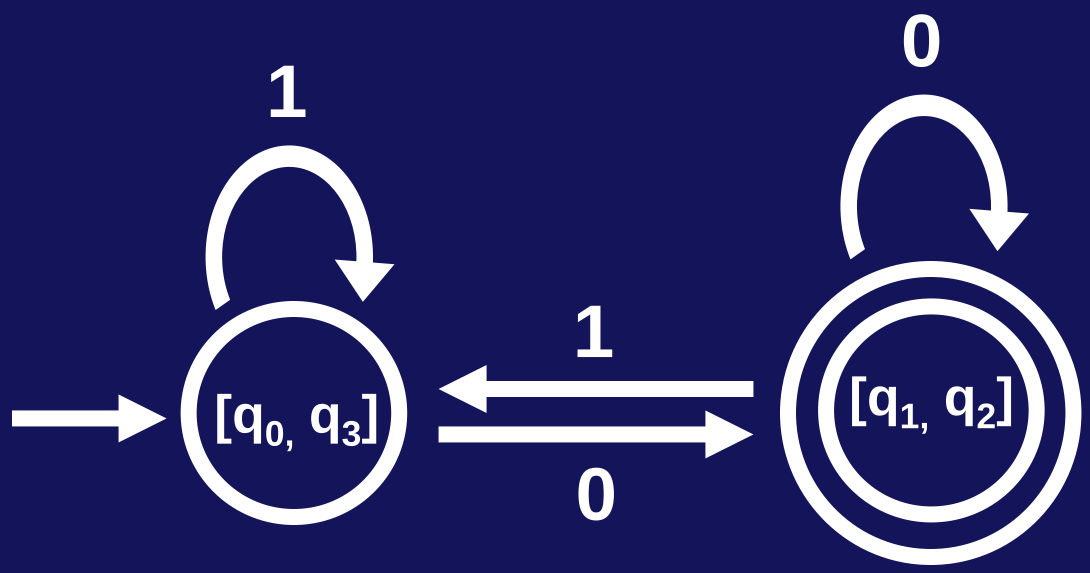
Have you noticed the reason for putting symbols X_1, X_2



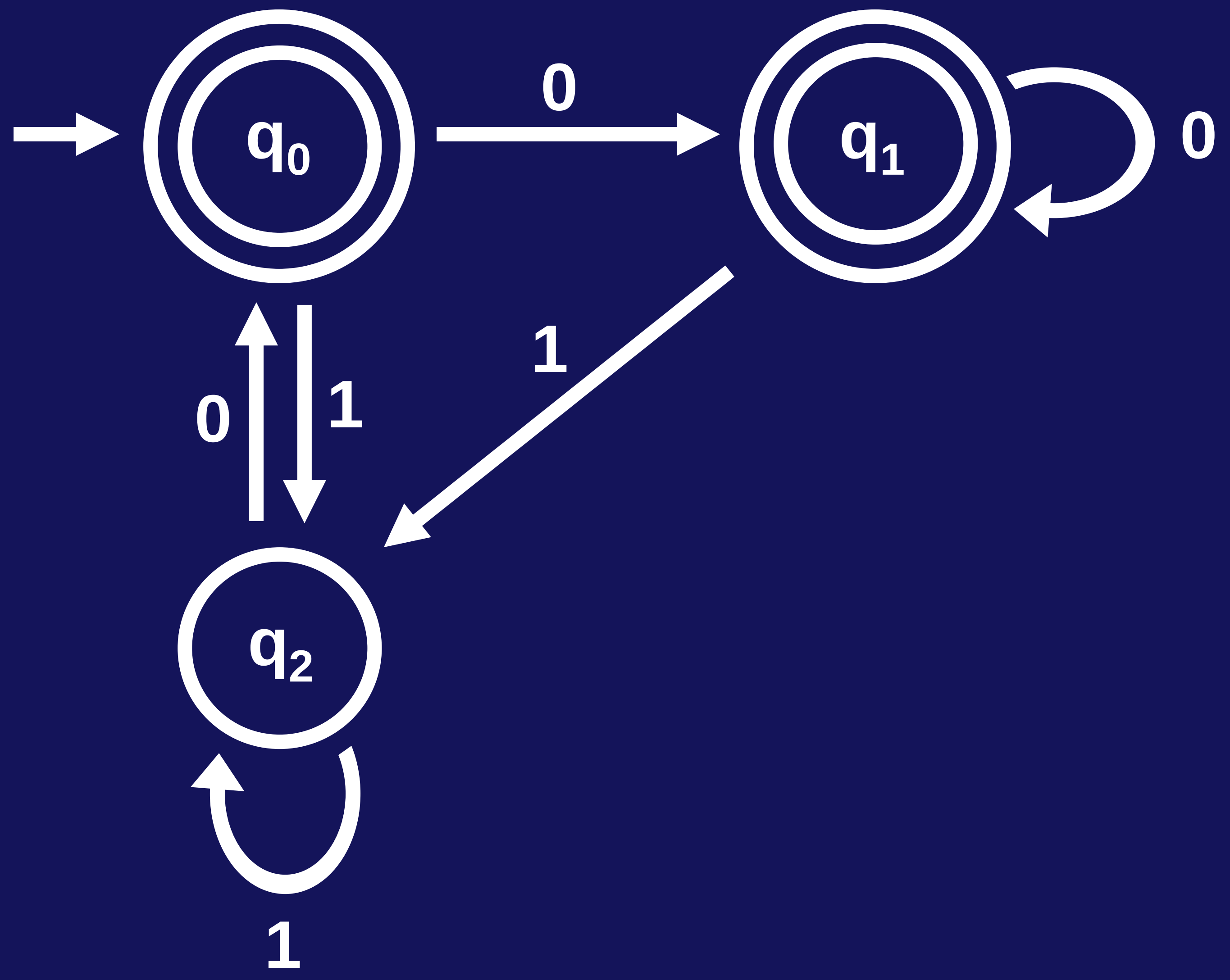


q_1	X_0		
q_2	X_0		
q_3		X_0	X_0
	q_0	q_1	q_2

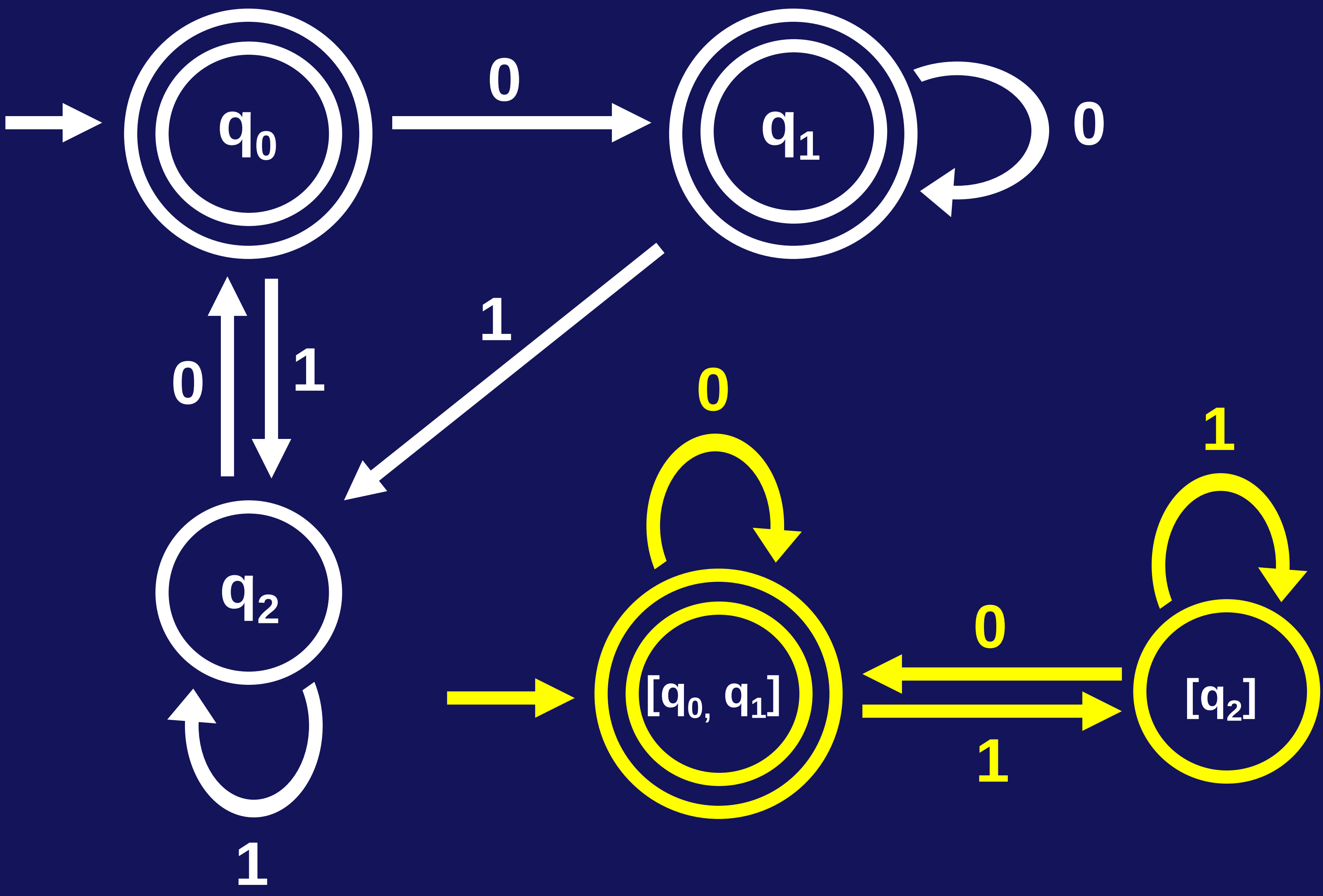

q0
q1
q2
q3
0
1
1
1
1
0
0
0

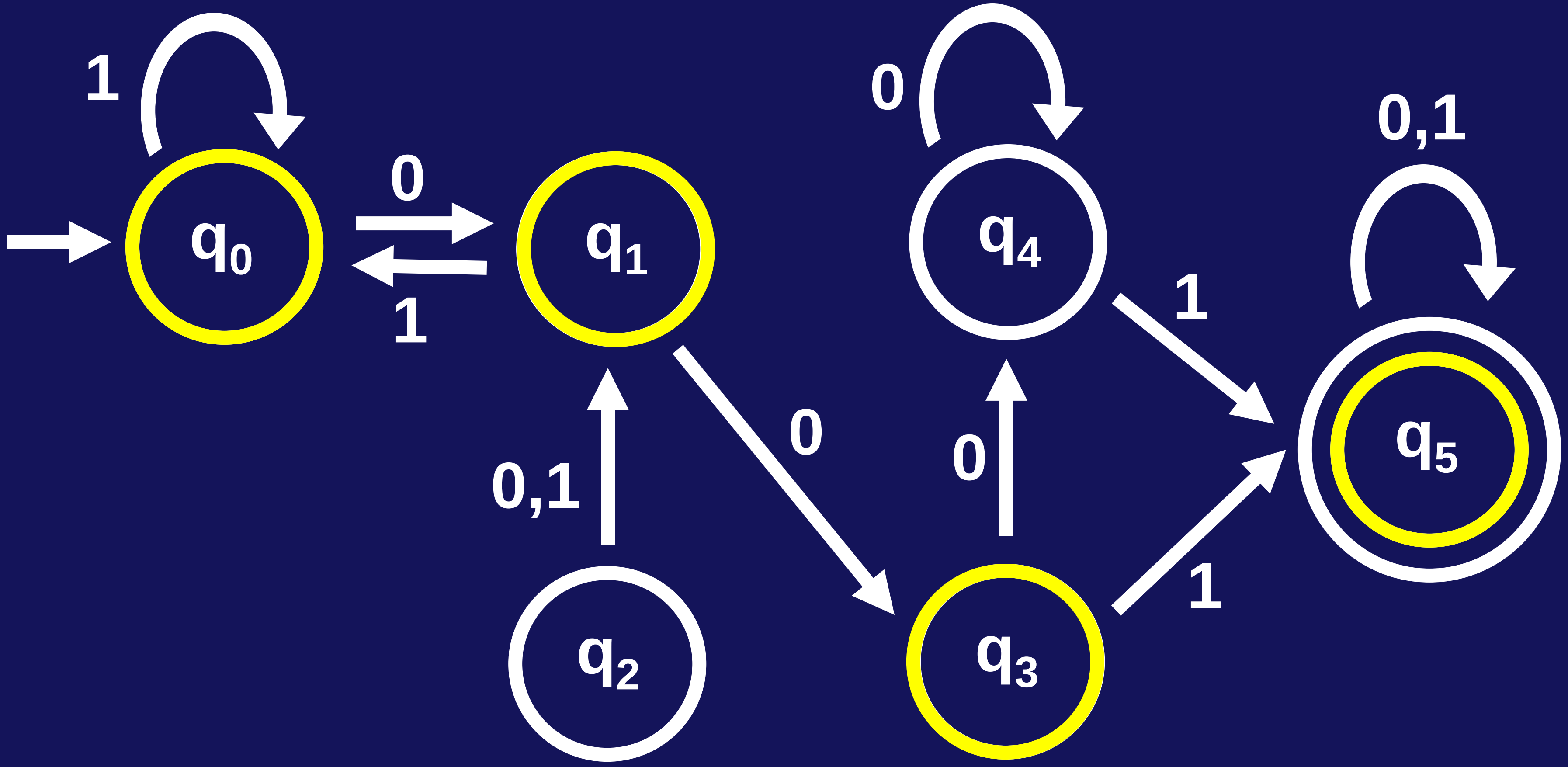

1
0
1
[q0, q3]
0
[q1, q2]

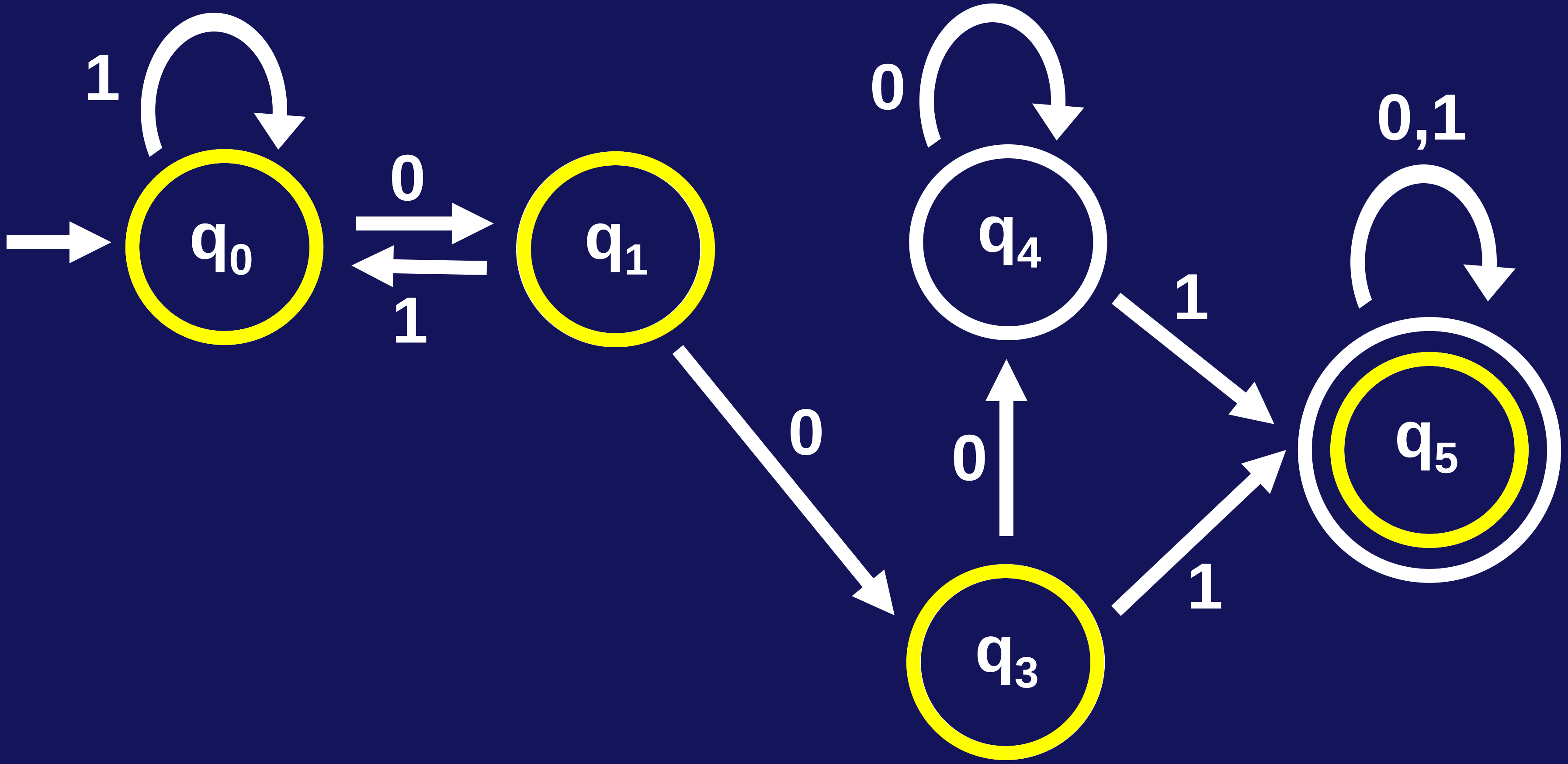
MINIMIZE



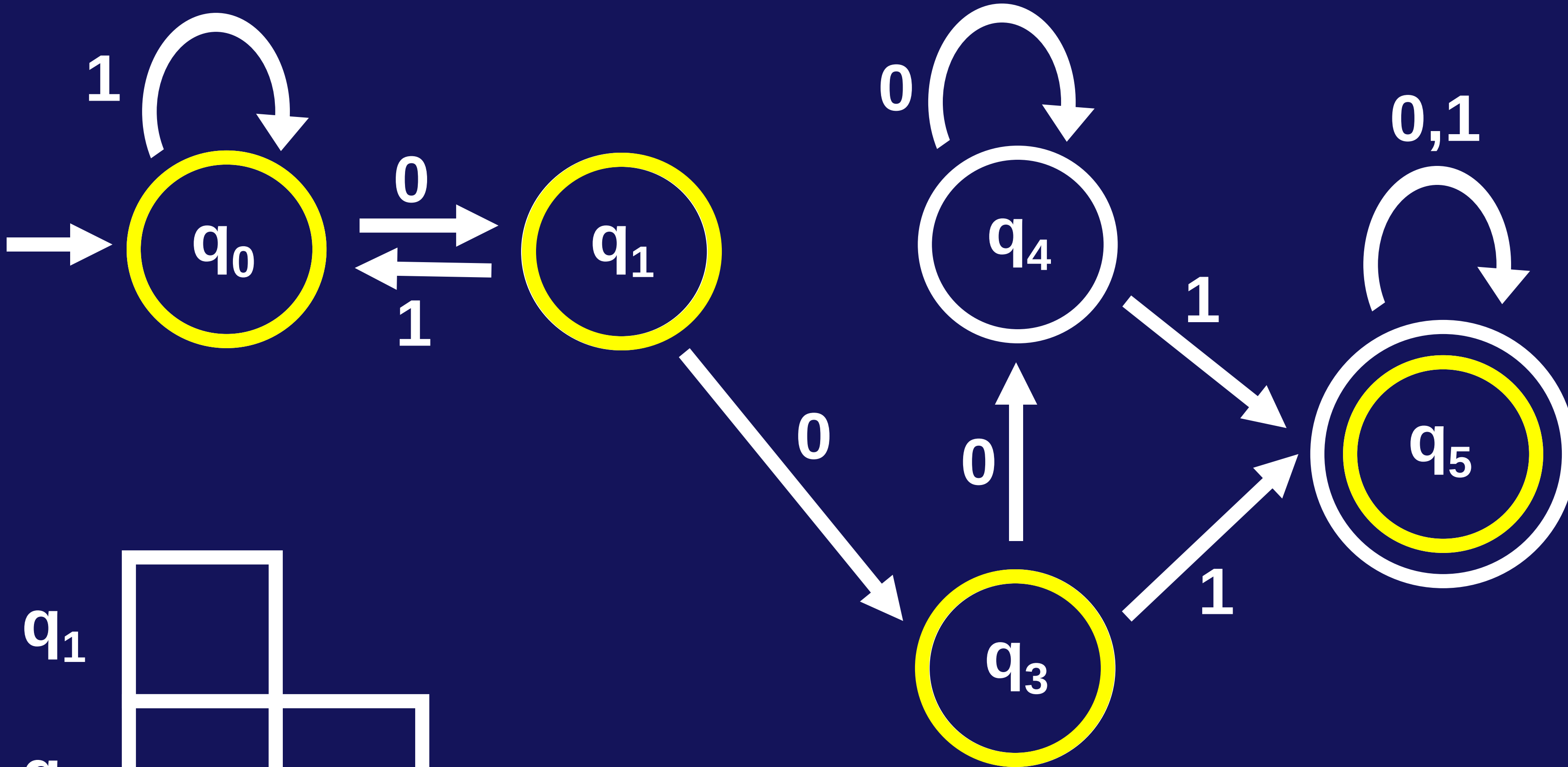
MINIMIZE



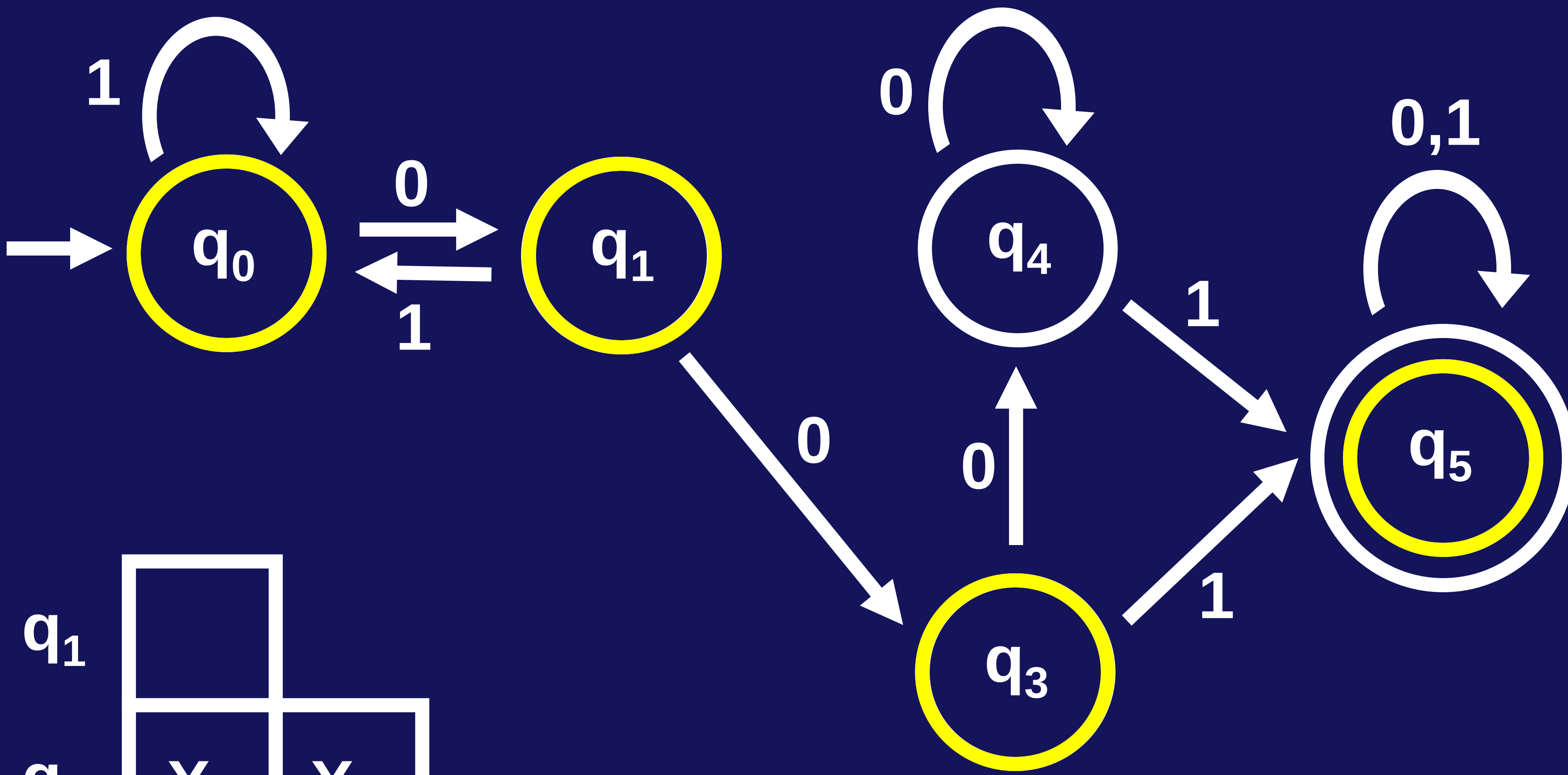

1
0
1
0
0,1
0
1
0,1
0
1
q_0
q_1
q_2
q_3
q_4
q_5



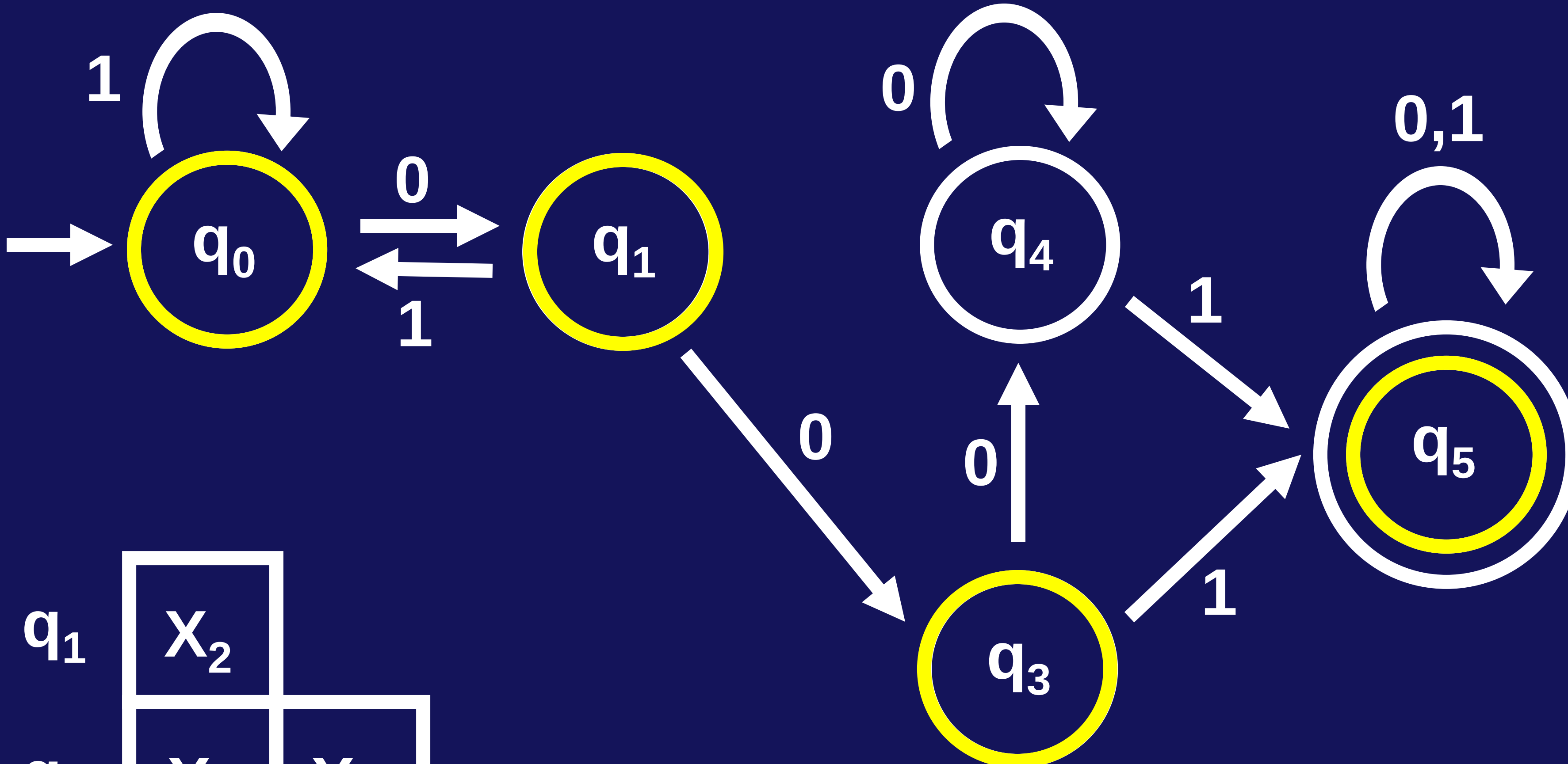
1
0
1
0
0,1
q0
q1
q4
1
0
0
q5
1
q3



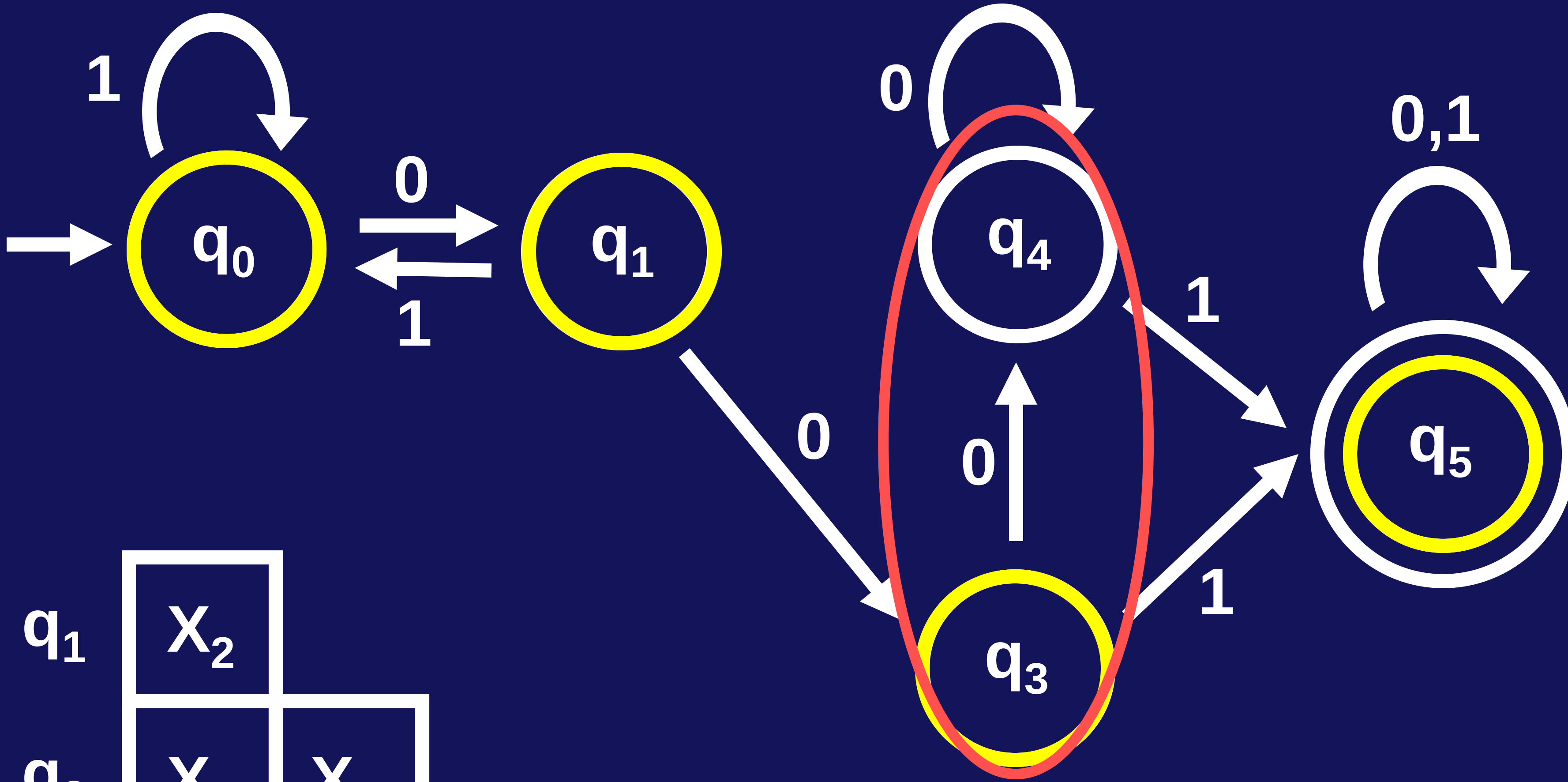
	q_0	q_1	q_3	q_4
q_1				
q_3				
q_4				
q_5	X_0	X_0	X_0	X_0



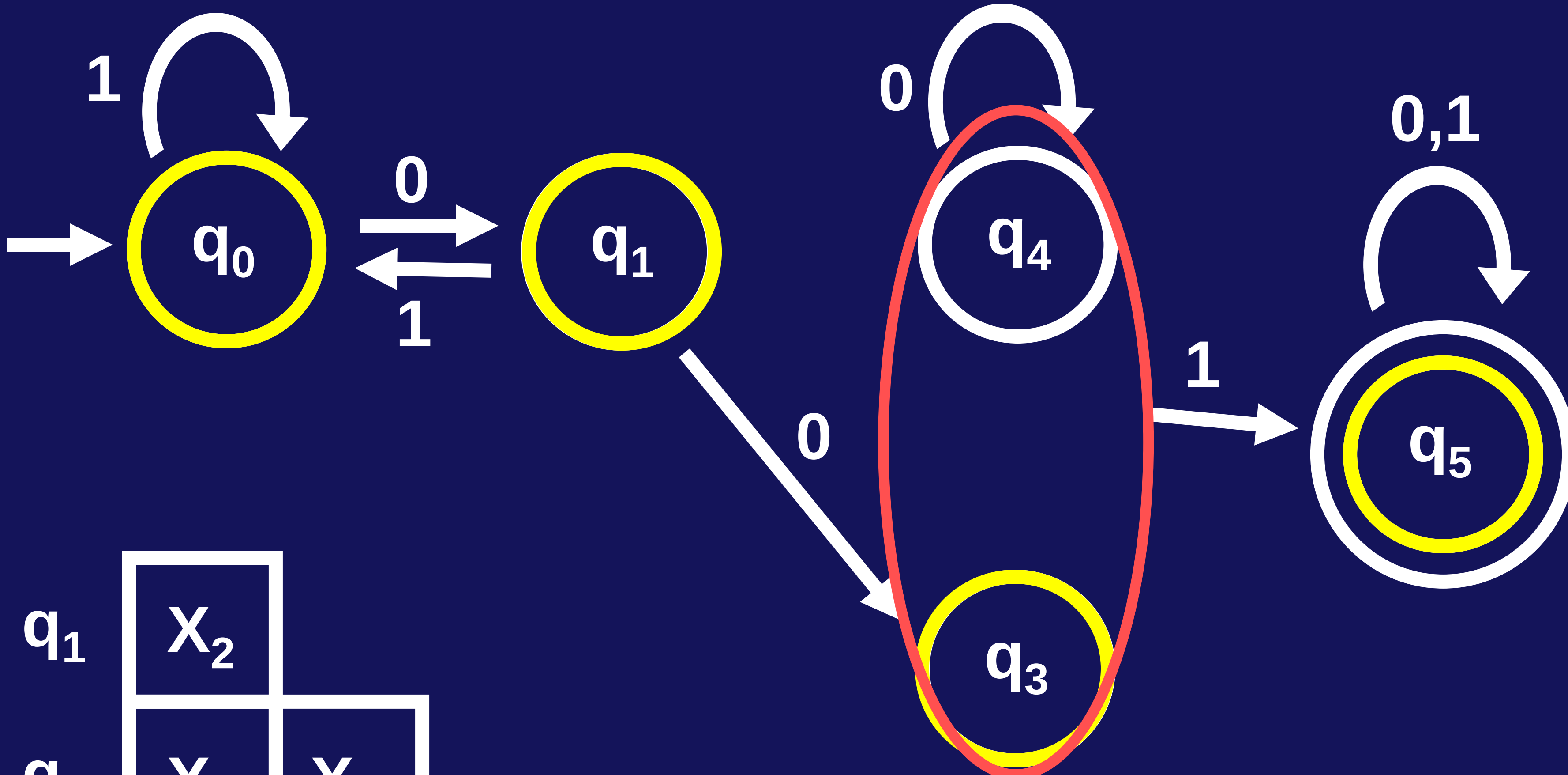
	q_0	q_1	q_3	q_4
q_1				
q_3	X_1	X_1		
q_4	X_1	X_1		
q_5	X_0	X_0	X_0	X_0



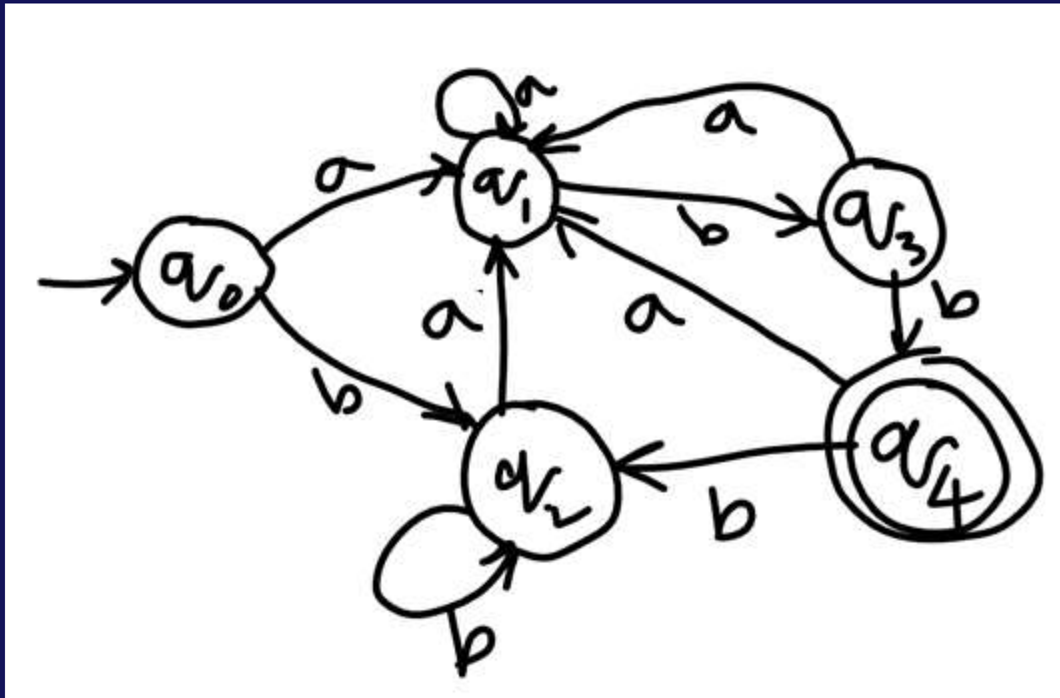
	q_0	q_1	q_3	q_4
q_1	X_2			
q_3	X_1	X_1		
q_4	X_1	X_1		
q_5	X_0	X_0	X_0	X_0



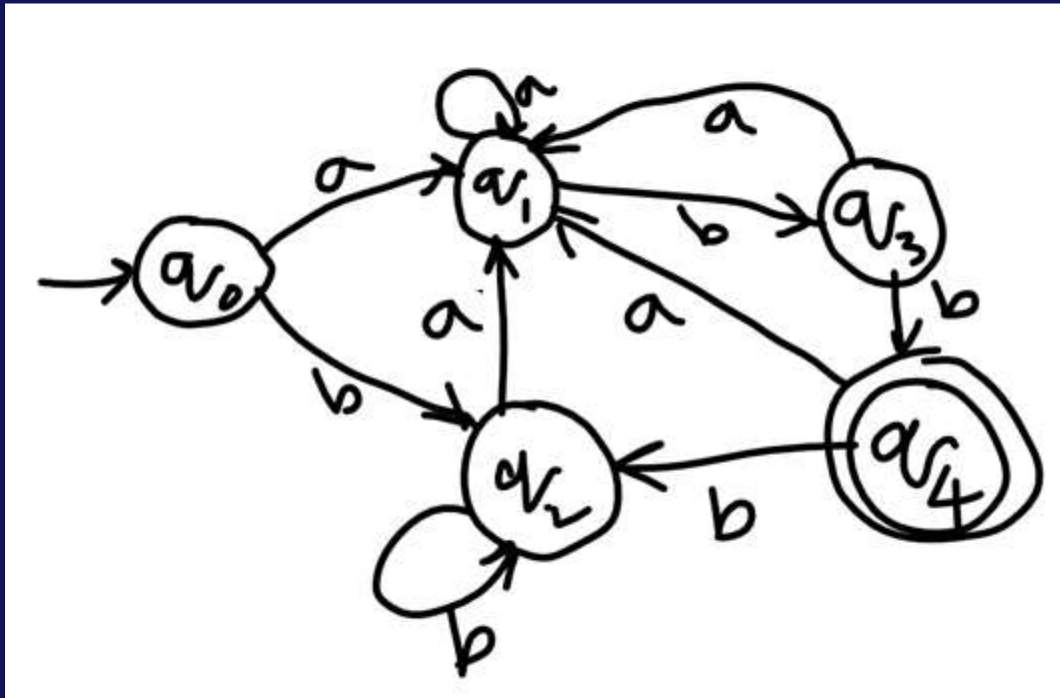
	q_0	q_1	q_3	q_4
q_1	X_2			
q_3	X_1	X_1		
q_4	X_1	X_1		
q_5	X_0	X_0	X_0	X_0



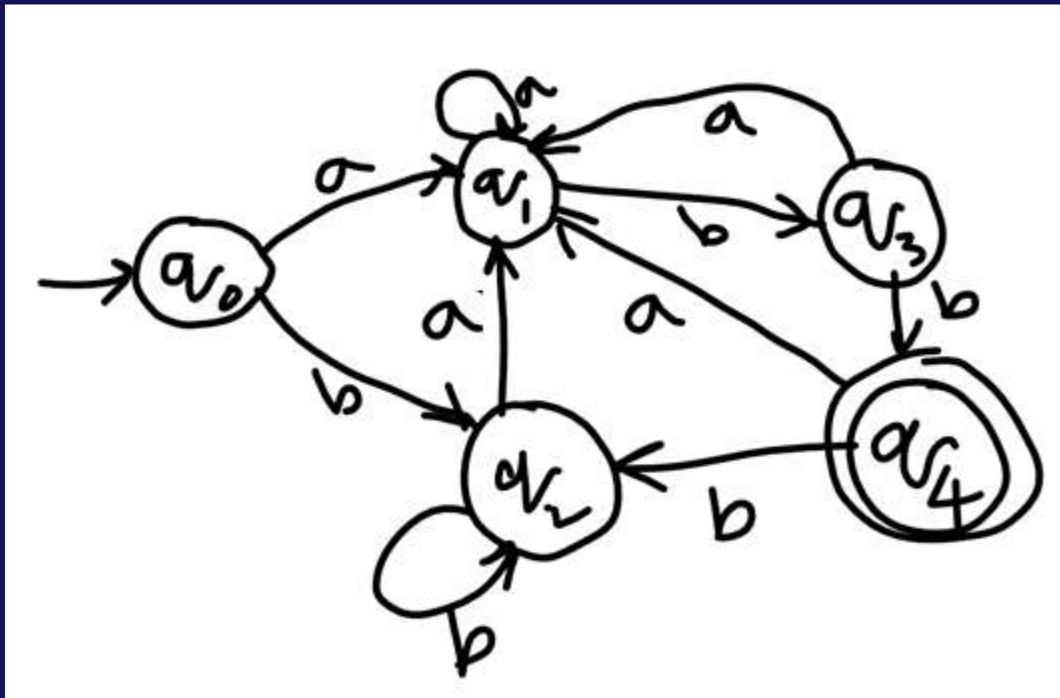
	q_0	q_1	q_3	q_4
q_1	X_2			
q_3	X_1	X_1		
q_4	X_1	X_1		
q_5	X_0	X_0	X_0	X_0



a
a
a
b
a
a
b
b
b
b
q_0
q_1
q_2
q_3
q_4

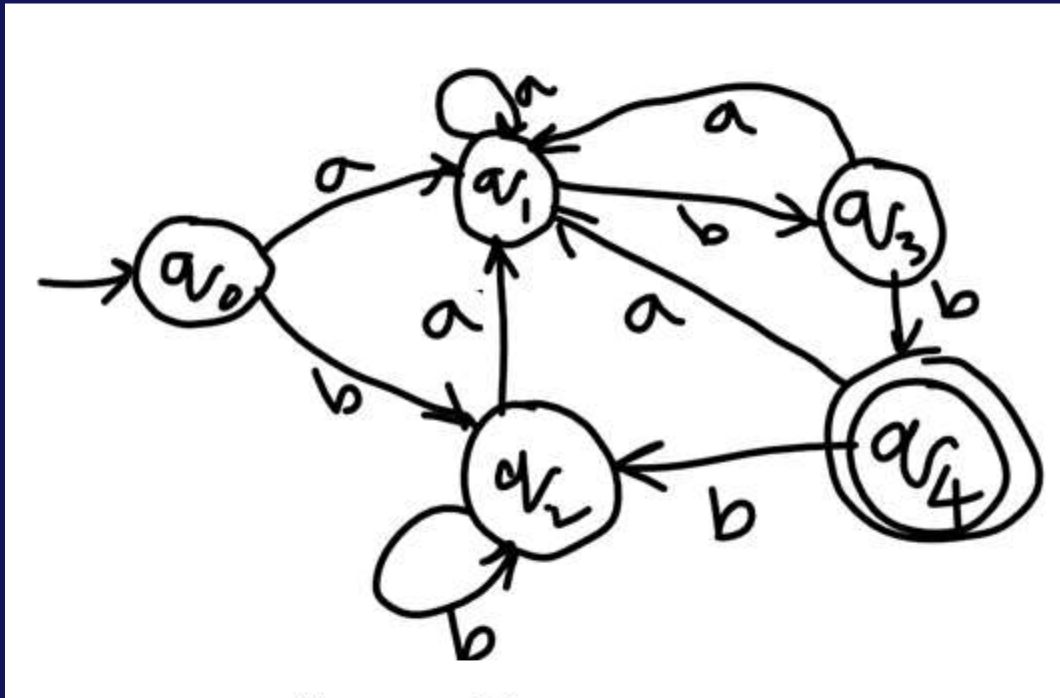


	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
$*q_4$	q_1	q_2



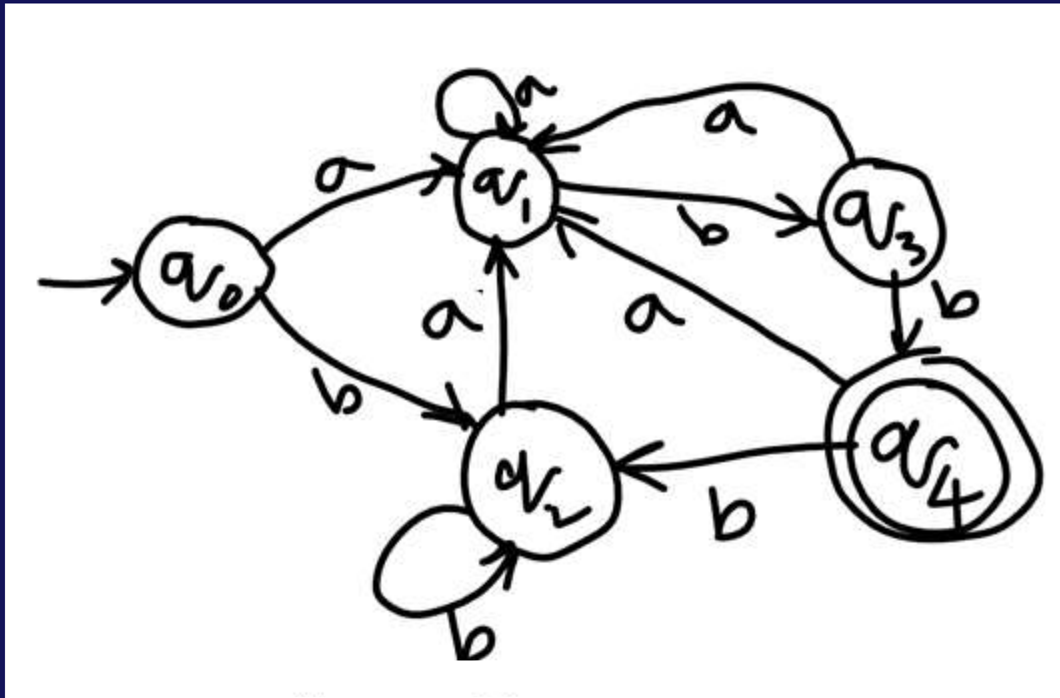
	q4	q3	q2	q1
q0	X0			
q1	X0			
q2	X0			
q3	X0			

	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_4	q_3	q_2	q_1
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_4	q_3	q_2	q_1
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
Removed ~~q_3~~	~~q_2~~	~~q_6~~
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

Step 1: Identify unreachable states

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

$\boxed{q_3 \text{ is not reachable}}$

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_4
	q_7	q_6	q_2

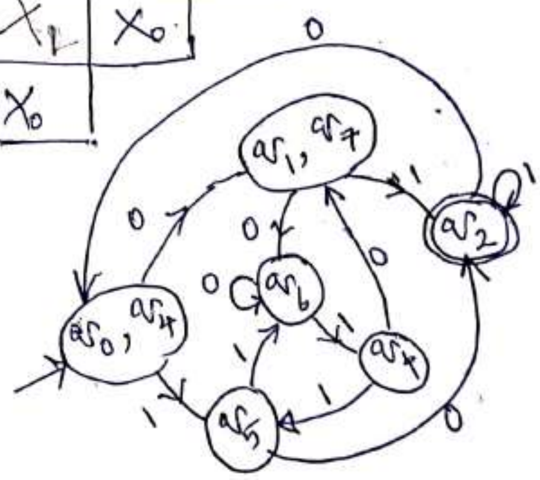
step 1: Identify unreachable states

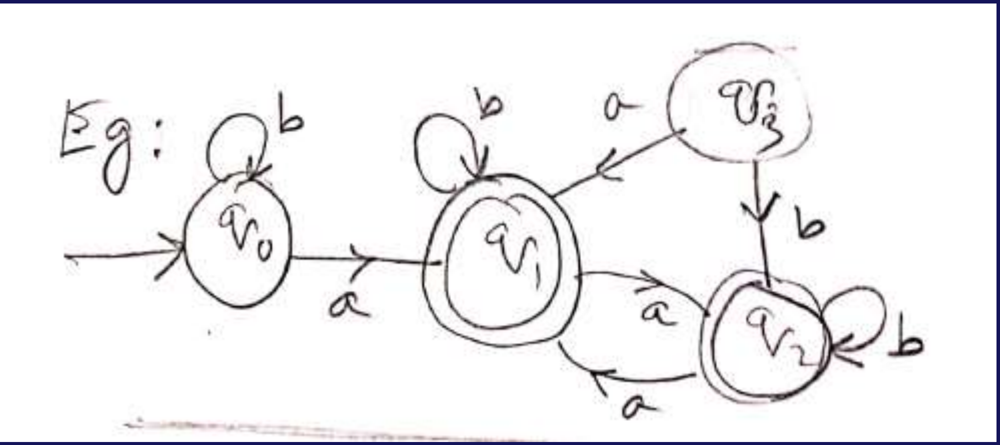
$\{q_0\}^+$

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

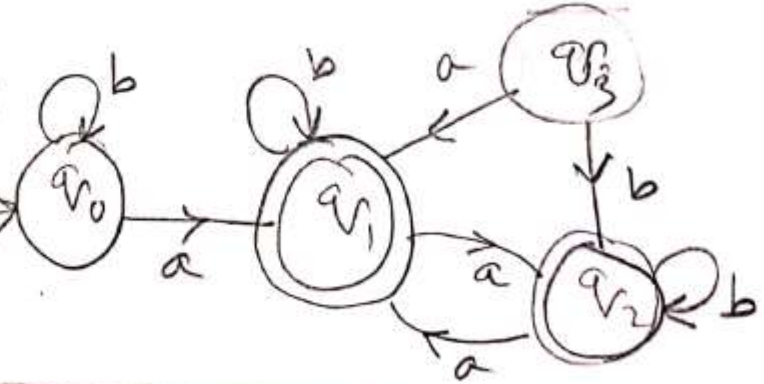
$\boxed{q_3 \text{ is not reachable}}$

	q_7	q_6	q_5	q_4	q_2	q_1
q_0	X_1	X_2	X_1	✓	X_0	X_1
q_1	✓	X_1	X_1	X_1	X_0	
q_2	X_0	X_0	X_0	X_0		
q_4	X_1	X_2	X_1			
q_5	X_1	X_1				
q_6	X_1					

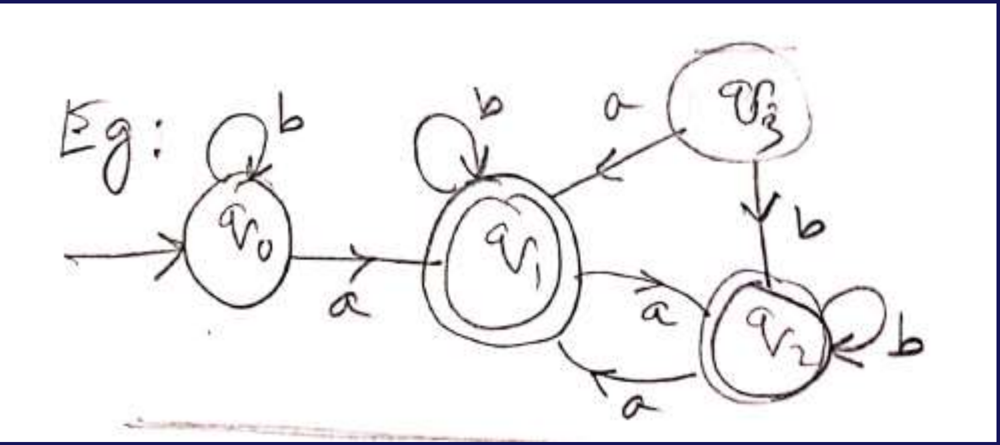



Eg:
b
b
a
q_3
q_0
q_1
b
a
a
q_2
b
a

Eg:

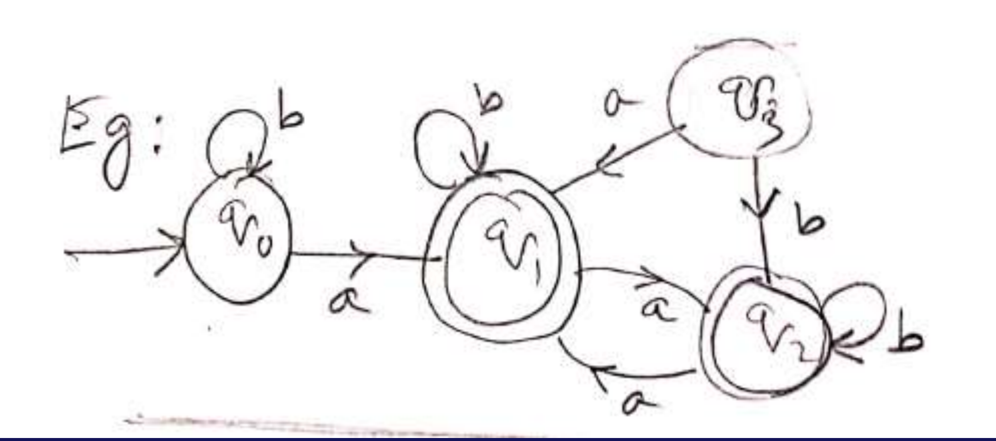


Unreachable = q_3



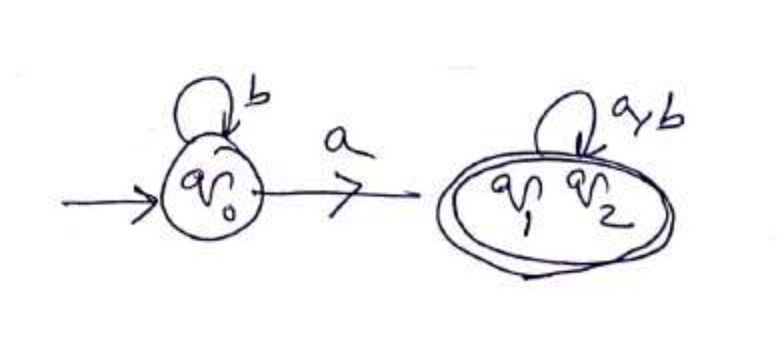
Unreachable = q_3

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 DFAs 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 $L \neq M$.

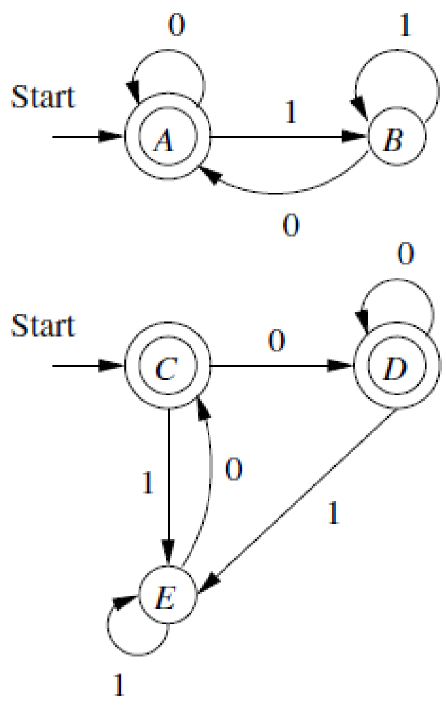


Figure 4.10: Two equivalent DFA's