

Chapter 2 – Finite Automata

Outline

- **Finite Automata**
 - **Deterministic Finite Automata (DFA)**
 - **Nondeterministic Finite Automata (NFA)**
- **Equivalence Between DFA and NFA**
- **Minimization of DFA**

Finite Automata

- A finite automata has a **finite set of states** and a **control unit**.
- The **control unit** moves from state to state in response to **external inputs**.
- Depending on the state transition there are two types of finite automata:
 1. **Deterministic Finite Automata (DFA)** – if the automaton cannot be in more than one state at any time.
 2. **Nondeterministic Finite Automata (NFA)** – if the automaton can be in several states at once.
- Let us first discuss about DFA, then NFA.

Deterministic Finite Automata (DFA)

- A deterministic finite automaton is the one that is in a single state after reading any sequence of inputs.
- The term '**deterministic**' refers to the fact that on each input there is one and only one state to which the automaton can transition from its current state.

Deterministic Finite Automata (DFA)

- ◆ A Deterministic Finite Automata (DFA) is a 5-tuple

$(Q, \Sigma, \delta, q_0, F)$ where

- ◆ Q is a finite set of states
- ◆ Σ (sigma) is an alphabet
- ◆ $\delta: Q \times \Sigma \rightarrow Q$ is a transition function
- ◆ $q_0 \in Q$ is the initial state
- ◆ $F \subseteq Q$ is a set of accepting states (or final states).

The Transition Function

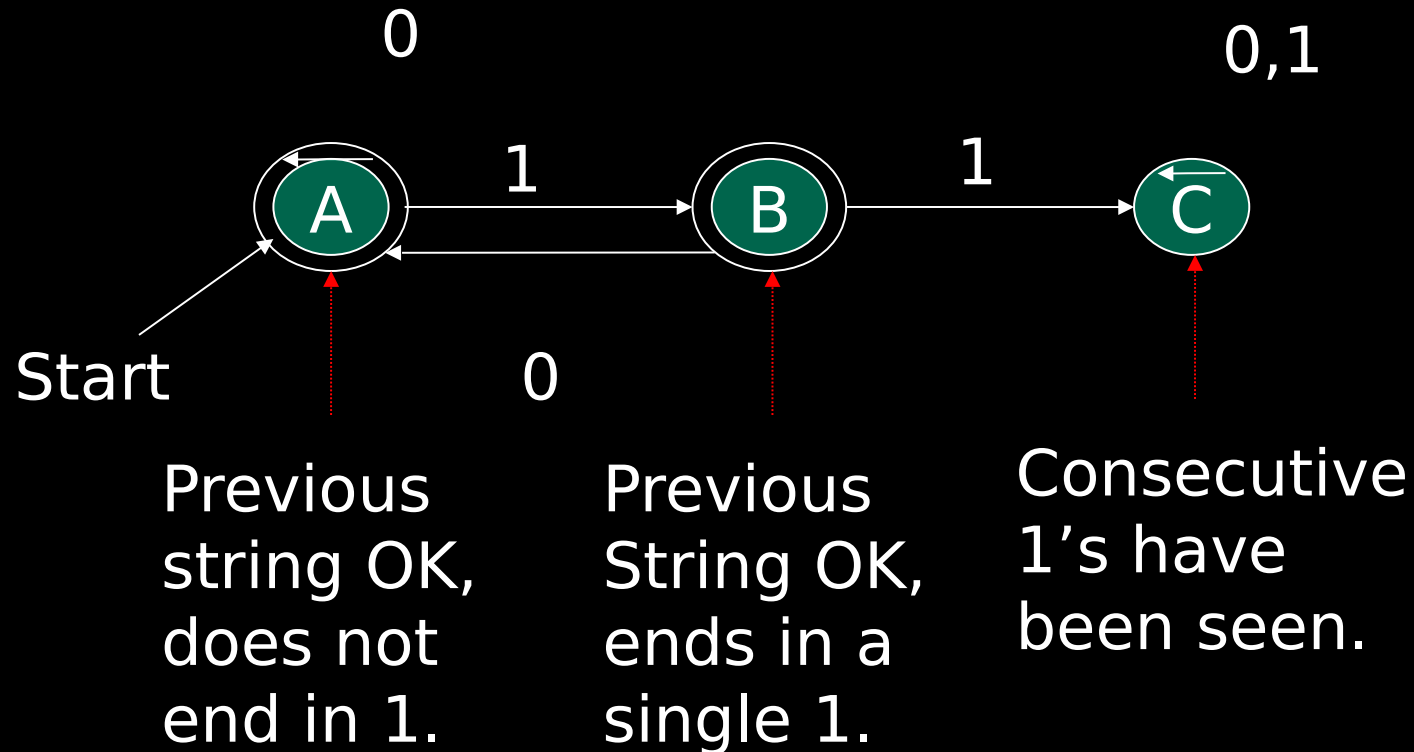
- ◆ It takes two arguments: a state and an input symbol.
- ◆ $\delta(q, a)$ = the state that the DFA goes to when it is in state q and input a is received.
- ◆ DFA do not allow non-deterministic state transitions. There can not be multiple state transition from state q with the same input a .

Graph Representation of DFA's

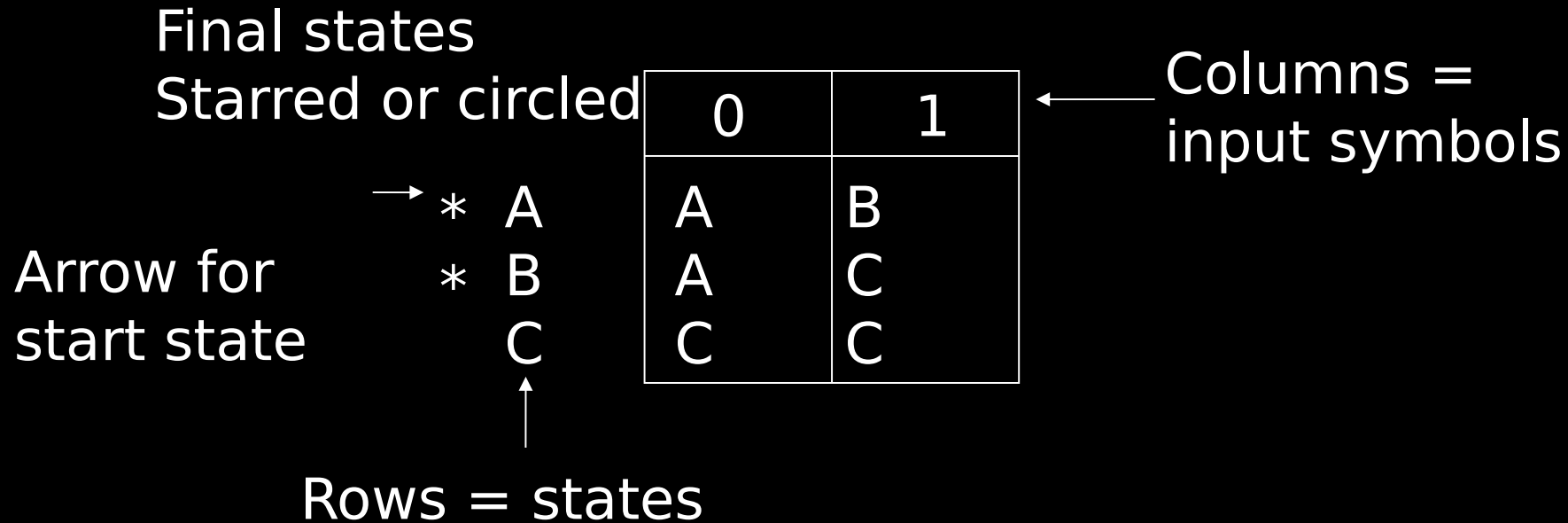
- ◆ Nodes = states.
- ◆ Edges represent transition function.
 - ◆ Edge from state p to state q labeled by all those input symbols that have transitions from p to q .
- ◆ Edge labeled “Start” to the start state.
- ◆ Final states indicated by double circles.
- ◆ DFA do not allow non-deterministic edges. i.e., there can not be more than one edge leaving any state with the same label.

Example: Graph of a DFA

Accepts all strings without two consecutive 1's.



Alternative Representation: Transition Table



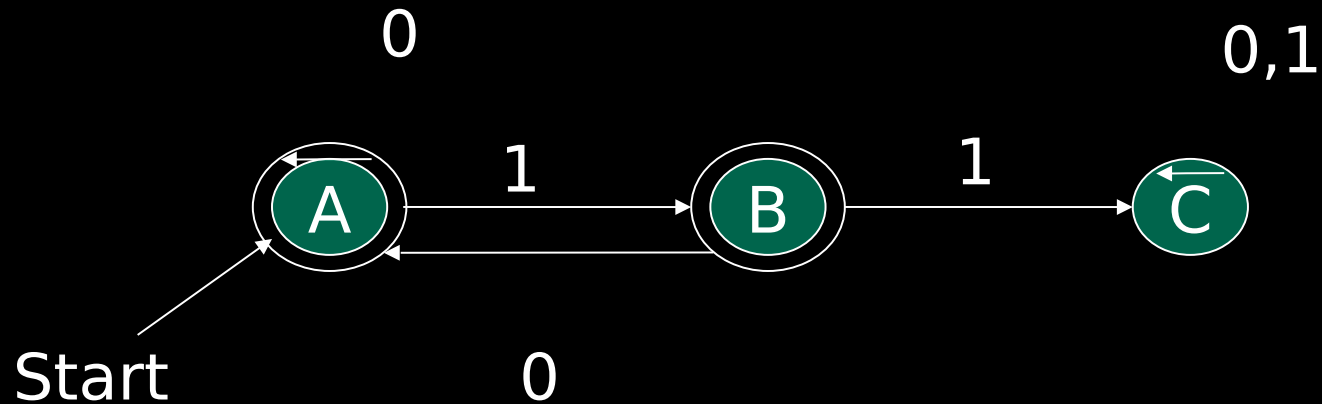
Language of a DFA

- ◆ Automata of all kinds define languages.
- ◆ If A is an automaton, $L(A)$ is its language.
- ◆ For a DFA A , $L(A)$ is the set of strings labeling paths from the start state to a final state.
- ◆ Formally: $L(A) =$ the set of strings w such that $\delta(q_0, w)$ is in F .

Example: String in a Language

String 101 is in the language of the DFA below.

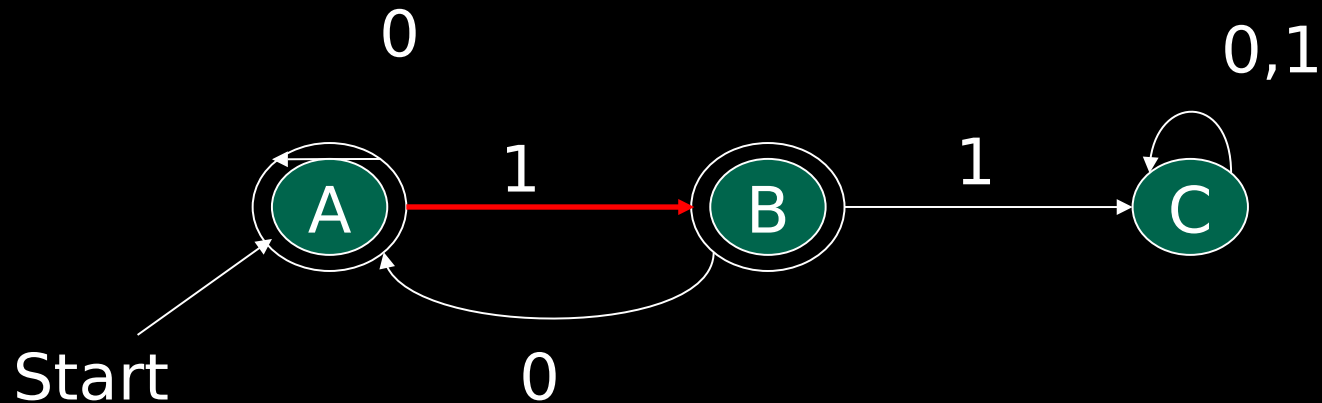
Start at A.



Example: String in a Language

String 101 is in the language of the DFA below.

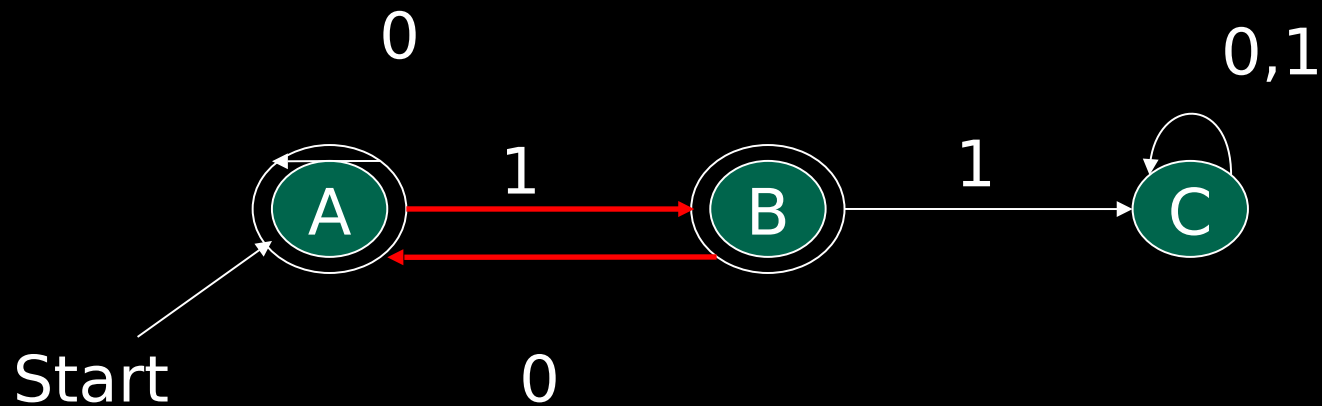
Follow arc labeled 1.



Example: String in a Language

String 101 is in the language of the DFA below.

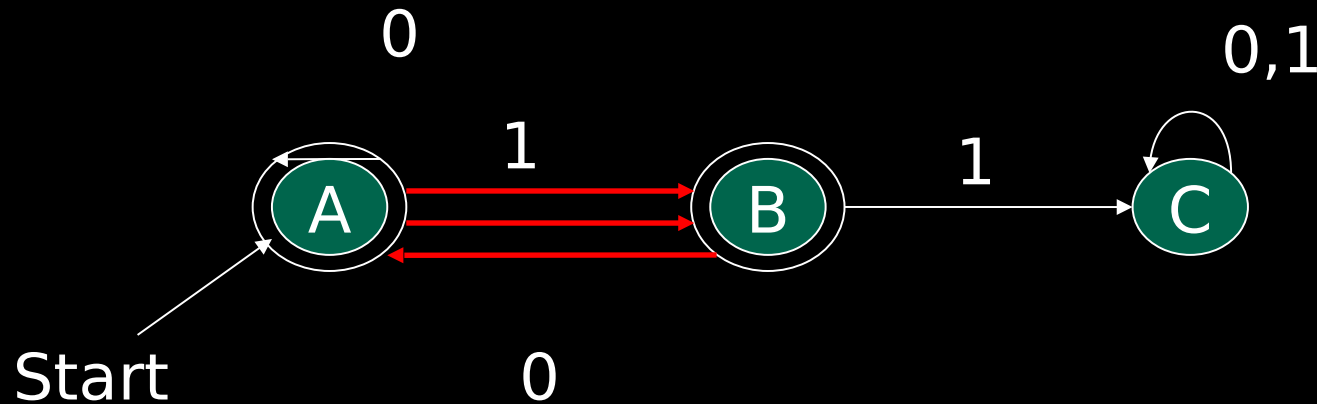
Then arc labeled 0 from current state B.



Example: String in a Language

String 101 is in the language of the DFA below.

Finally arc labeled 1 from current state A. Result is an accepting state, so 101 is in the language.



Example – Concluded

- ◆ The language of our example DFA is:
 $\{w \mid w \text{ is in } \{0,1\}^* \text{ and } w \text{ does not have two consecutive 1's}\}$

Such that...

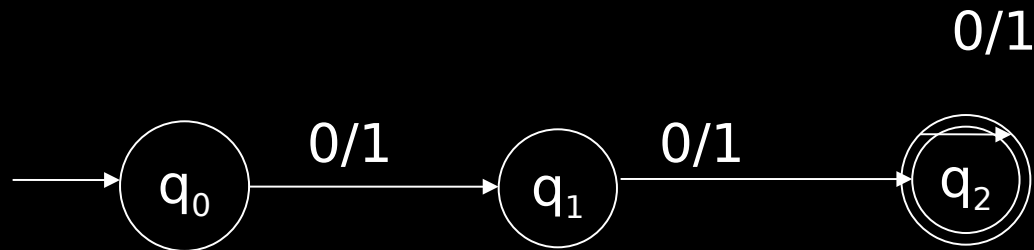
These conditions
about w are true.

Read a *set former* as
“The set of strings w ...”

More Examples on DFA

1. Given a DFA, M such that:

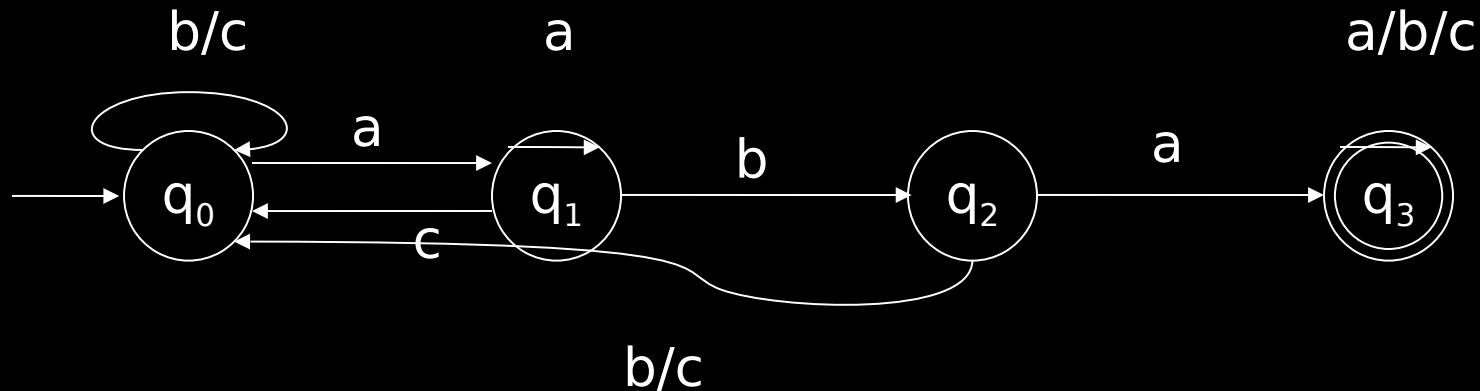
$L(M) = \{x \mid x \text{ is a string of 0's and 1's and } |x| \geq 2\}$



More Examples on DFA

2. Given a DFA, M such that:

$L(M) = \{x \mid x \text{ is in } \{a,b,c\}^* \text{ and } x \text{ contains the substring } aba\}$



More Examples on DFA

3. Let M be a DFA given by:

$M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_0\})$ and δ is given as:

$$\delta(q_0, a) = q_0$$

$$\delta(q_0, b) = q_1$$

$$\delta(q_1, a) = q_1$$

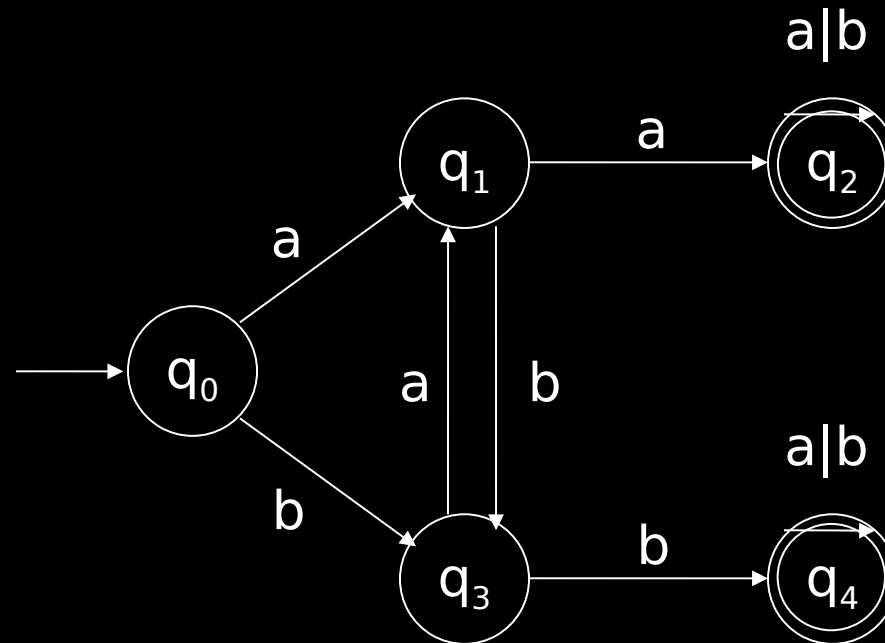
$$\delta(q_1, b) = q_0$$

Construct transition diagram and table for the given DFA and determine the language

More Examples on DFA

4. Given a DFA, M such that:

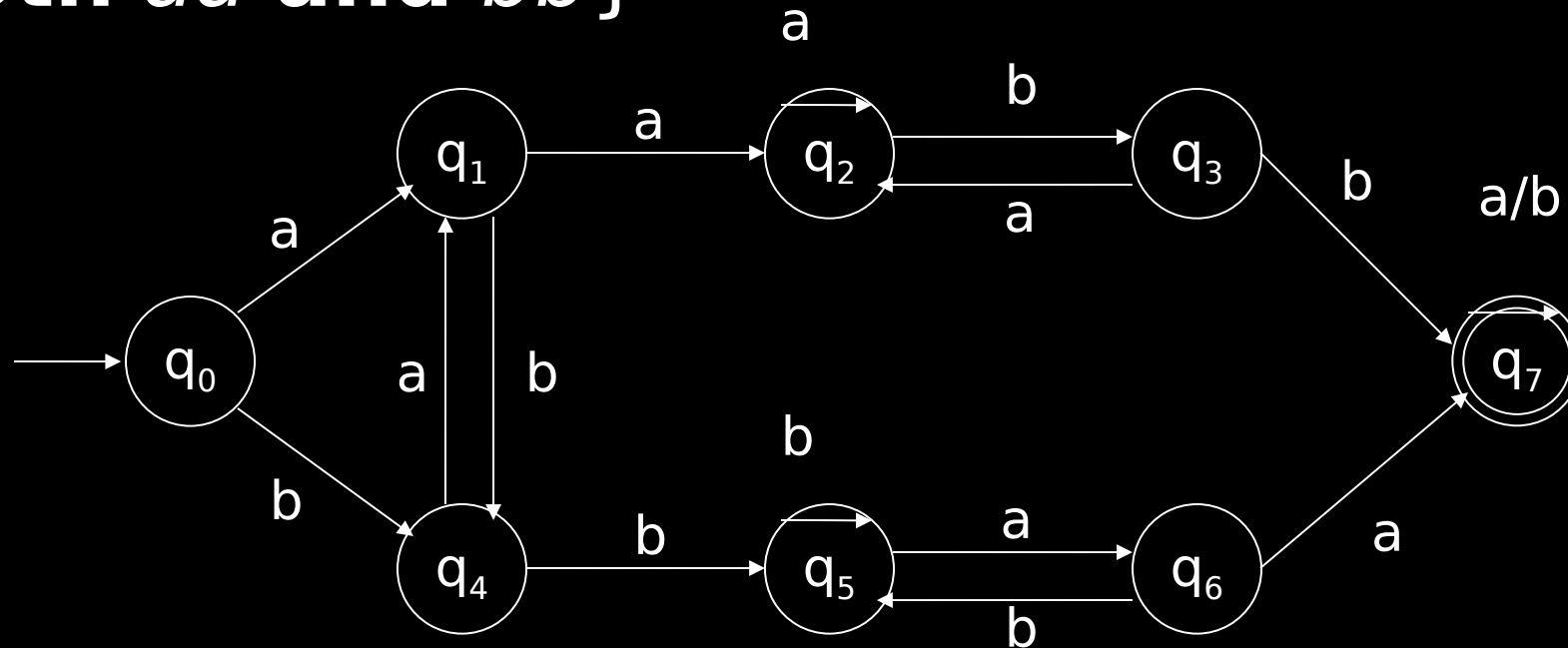
$L(M) = \{x \mid x \text{ is in } \{a,b\}^* \text{ and } x \text{ contains } aa \text{ or } bb\}$



More Examples on DFA

5. Given a DFA, M such that:

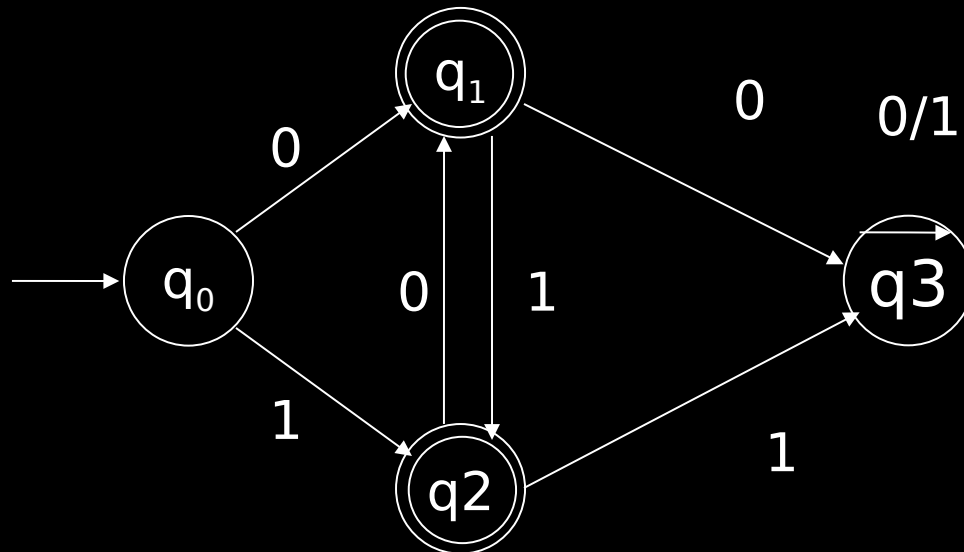
$L(M) = \{x \mid x \text{ is in } \{a,b\}^* \text{ and } x \text{ contains both } aa \text{ and } bb\}$



More Examples on DFA

6. Given a DFA, M such that:

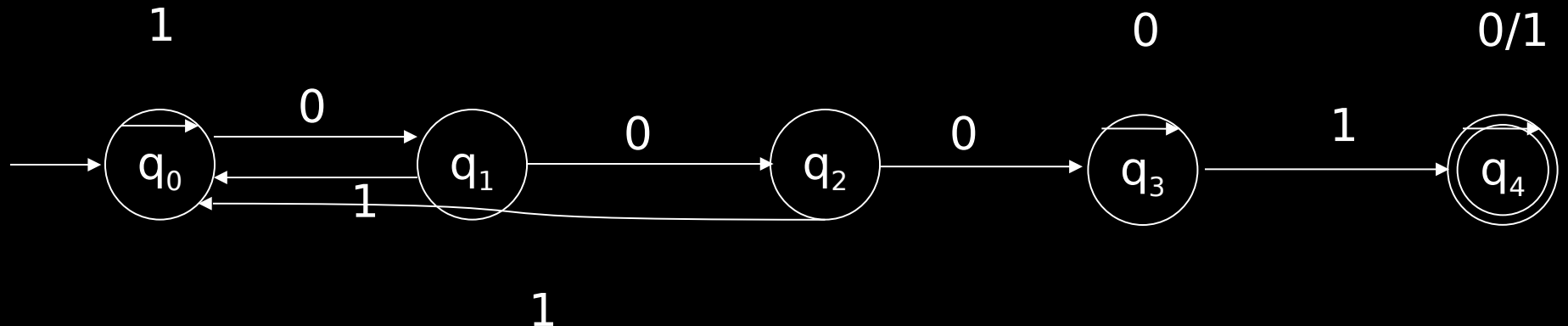
$L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and } x \text{ has **neither** } 11 \text{ **nor** } 00 \text{ as a substring}\}$



More Examples on DFA

7. Given a DFA, M such that:

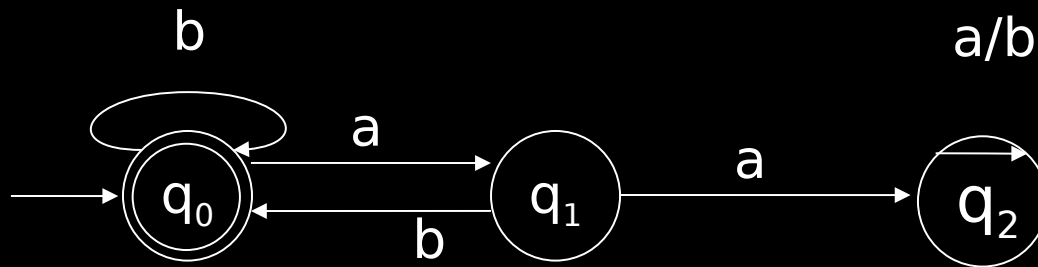
$L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and } x \text{ contains } 0001 \text{ as a substring}\}$



More Examples on DFA

8. Given a DFA, M such that:

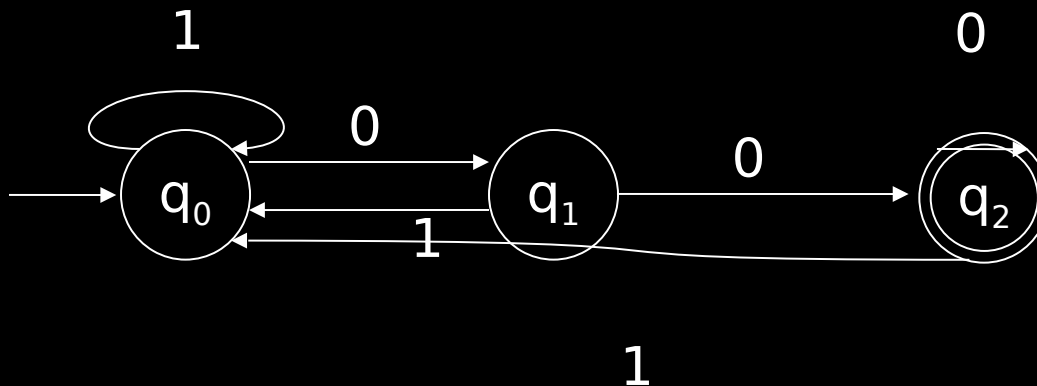
$L(M) = \{x \mid x \text{ is in } \{a,b\}^* \text{ and } a \text{ is immediately followed by } b\}$



More Examples on DFA

9. Given a DFA, M such that:

$L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and } x \text{ contains strings ending in } 00\}$



Exercise on DFA

10. For $\Sigma = \{a,b\}$, construct DFA's that accept the sets consisting of:

- (a) all strings with exactly one a
- (b) all strings with at least one a,
- (c) all strings with no more than three a's
- (d) all strings with at least one a and exactly two b's.
- (e) all the strings with exactly two a's and more than two b's

Exercise on DFA

11. Given a DFA, M such that:

a) $L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and } x \text{ has a substring } 011\}$

b) $L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and the second symbol from the right is a } 1\}$

c) $L(M) = \{x \mid x \text{ is in } \{0,1\}^* \text{ and Either begin or end (or both) with } 01 \}$

Nondeterministic Finite Automata (NFA)

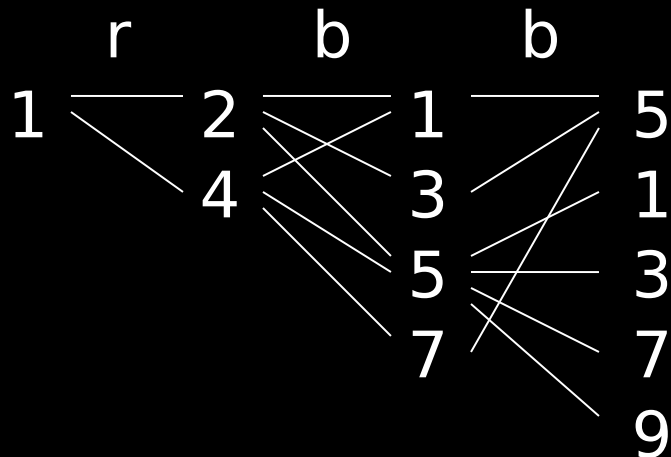
Nondeterminism

- ◆ A *nondeterministic finite automaton* has the ability to be in several states at once.
- ◆ Transitions from a state on an input symbol can be to any set of states.
- ◆ Start in one start state.
- ◆ Accept if any sequence of choices leads to a final state.
- ◆ *Intuitively*: the NFA always “guesses right.”

Example: Chessboard – (2)

1	2 red	3
4 red	5	6 red
7	8 red	9

Check rbb



	r	b
1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

Accept, since final state reached ²⁹

Example: Accepted Moves

1	2	3
4	5	6
7	8	9

- ◆ For our chessboard NFA we saw that rbb is accepted.
- ◆ If the input consists of only b's, the set of accessible states alternates between {5} and {1,3,7,9}, so only even-length, nonempty strings of b's are accepted.
- ◆ What about strings with at least one r?

Formal Definition of NFA

- ◆ An NFA is a five-tuple:

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

Q A finite set of states

Σ A finite input alphabet

q_0 The initial/starting state, q_0 is in Q

F A set of final/accepting states, which is a subset of Q

δ A transition function, which is a total function from $Q \times \Sigma$ to 2^Q

$\delta: (Q \times \Sigma) \rightarrow 2^Q$ - 2^Q is the power set of Q , the set of all subsets of Q

NFA Differences with DFA

- Three major differences

1. The range of δ is in the power set 2^Q
2. ϵ (empty string) transitions are possible in NFA. NFA can make a transition without consuming an input symbol.
3. In an NFA, the set $\delta(q_i, a)$ may be empty; there is no transition defined for this specific situation.

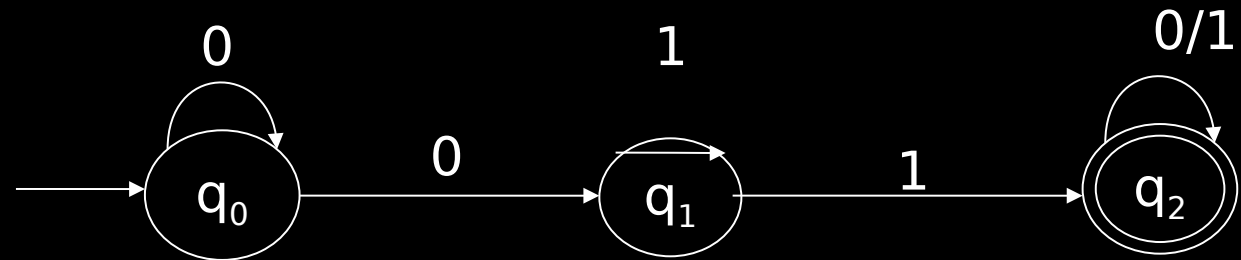
Example #1: some 0's followed by some 1's

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

Start state is q_0

$F = \{q_2\}$



$\delta:$	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{\}$
q_1	$\{\}$	$\{q_1, q_2\}$
q_2^*	$\{q_2\}$	$\{q_2\}$

◆ Example #2: pair of 0's or pair of 1's

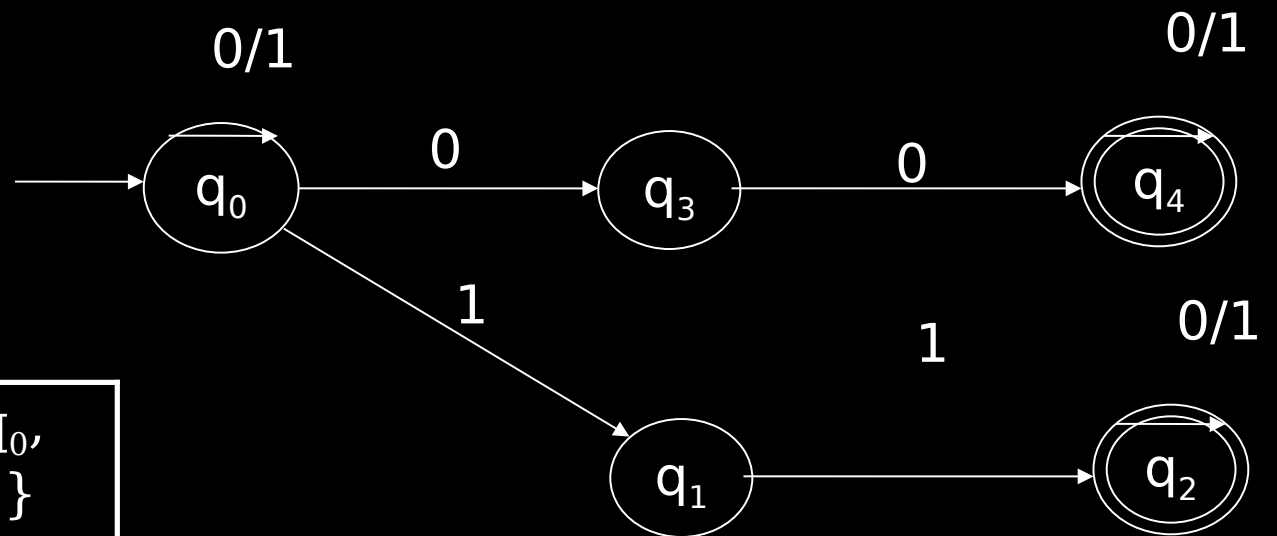
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

Start state is q_0

$F = \{q_2, q_4\}$

$\delta:$	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\{\}$	$\{q_2\}$
q_2^*	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	$\{\}$
q_4^*	$\{q_4\}$	$\{q_4\}$



Language of an NFA

- ◆ A string w is accepted by an NFA if $\delta(q_0, w)$ contains at least one final state.
- ◆ The language of the NFA is the set of strings it accepts.

Equivalence of DFA's, NFA's

Equivalence of DFA's, NFA's

- ◆ A DFA can be turned into an NFA that accepts the same language.
- ◆ If $\delta_D(q, a) = p$, let the NFA have $\delta_N(q, a) = \{p\}$.
- ◆ Then the NFA is always in a set containing exactly one state – the state the DFA is in after reading the same input.

Equivalence

- ◆ Surprisingly, for any NFA there is a DFA that accepts the same language.
- ◆ Proof is the *subset construction*.
- ◆ The number of states of the DFA may have exponentially more states than the NFA.
- ◆ Thus, NFA's accept exactly the regular languages like DFA's.

Subset Construction

- ◆ Given an NFA with states Q , inputs Σ , transition function δ_N , start state q_0 , and final states F , construct equivalent DFA with:
 - ▶ States 2^Q (Set of subsets of Q).
 - ▶ Inputs Σ .
 - ▶ Start state $\{q_0\}$.
 - ▶ Final states = all those with a member of F .

Critical Point

- ◆ The DFA states have *names* that are sets of NFA states.
- ◆ But as a DFA state, an expression like $\{p,q\}$ must be read as a single symbol, not as a set.

Subset Construction

- ◆ The transition function δ_D is defined by:
 $\delta_D(\{q_1, \dots, q_k\}, a)$ is the union over all $i = 1, \dots, k$ of $\delta_N(q_i, a)$.
- ◆ **Example:** We'll construct the DFA equivalent of our “chessboard” NFA.

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
* 8	4,6	5,7,9
9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}		
{5}		

Alert: What we're doing here is the *lazy* form of DFA construction, where we only construct a state if we are forced to.

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}		
{2,4,6,8}		
{1,3,5,7}		

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

*

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}		
{1,3,5,7}		
{1,3,7,9}		

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
{1,3,5,7}		
* {1,3,7,9}		
* {1,3,5,7,9}		

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
* {1,3,7,9}		
* {1,3,5,7,9}		

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
* {1,3,7,9}	{2,4,6,8}	{5}
* {1,3,5,7,9}		

Example: Subset Construction

	r	b
→ 1	2,4	5
2	4,6	1,3,5
3	2,6	5
4	2,8	1,5,7
5	2,4,6,8	1,3,7,9
6	2,8	3,5,9
7	4,8	5
8	4,6	5,7,9
* 9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
* {1,3,7,9}	{2,4,6,8}	{5}
* {1,3,5,7,9}	{2,4,6,8}	{1,3,5,7,9}

Example 1 : Convert NFA to DFA

$M = (\{A, B\}, \{0, 1\}, \delta, A, \{A\})$ where δ is given by:

* →

	0	1
A	A	B
B	B	A, B

* →

*

	0	1
A	A	B
B	B	{A, B}
{A, B}	{A, B}	{A, B}

Example 2 : Convert NFA to DFA

$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$ where δ is given by:

	a	b
→ q ₀	q ₀ , q ₁	q ₂
q ₁	q ₀	q ₁
* q ₂	ϕ	q ₀ , q ₁

	a	b
→ q ₀	{q ₀ , q ₁ }	q ₂
* q ₂	ϕ	{q ₀ , q ₁ }
{q ₀ , q ₁ }	{q ₀ , q ₁ }	{q ₁ , q ₂ }
* {q ₁ , q ₂ }	q ₀	{q ₀ , q ₁ }
ϕ	ϕ	ϕ

Example 3 : Convert NFA to DFA

$M = (\{P, Q, R, S\}, \{0, 1\}, \delta, P, \{S\})$ where δ is given by:

→

	0	1
P	P, Q	P
Q	R	R
R	S	ϕ
S	S	S

*

→

	0	1
P	{P, Q}	P
{P, Q}	{P, Q, R}	{P, R}
{P, R}	{P, Q, S}	P
{P, Q, R}	{P, Q, R, S}	{P, R}
{P, Q, S}	{P, Q, R, S}	{P, R, S}
{P, R, S}	{P, Q, S}	{P, S}
{P, S}	{P, Q, S}	{P, S}
{P, Q, R, S}	{P, Q, R, S}	{P, R, S}

*
*
*
*

Example 4 : Convert NFA to DFA

$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_2, q_3\})$ where δ is given by:

	a	b
q0	q0,q1	q0
q1	q2	q1
q2	q3	q3
q3	ϕ	q2

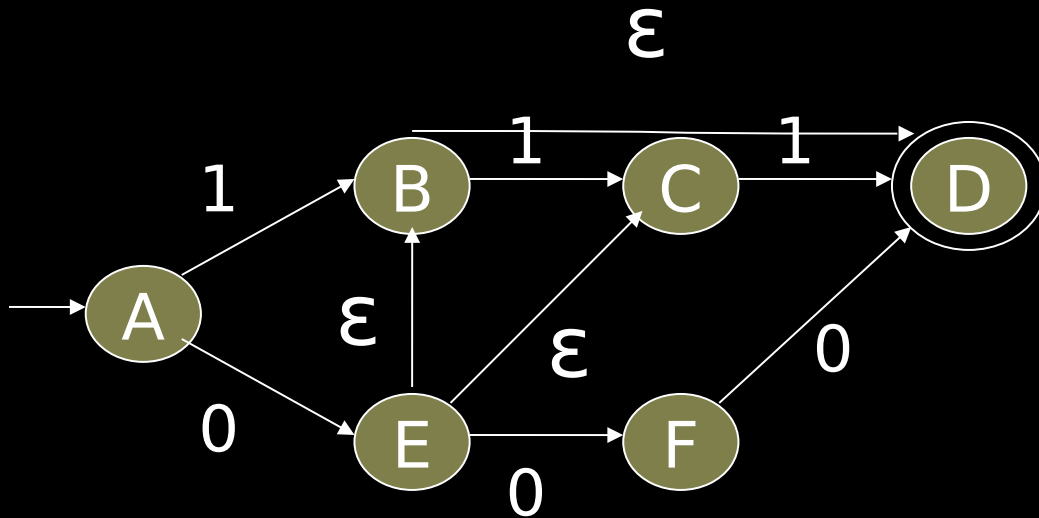
→

	a	b
q0	{q0,q1}	q0
{q0,q1}	{q0,q1,q2}	{q0,q1}
{q0,q1,q2}	{q0,q1,q2,q3}	{q0,q1,q3}
{q0,q1,q3}	{q0,q1,q2}	{q0,q1,q2}
{q0,q1,q2,q3}	{q0,q1,q2,q3}	{q0,q1,q2,q3}

NFA's With ϵ -Transitions

- ◆ We can allow state-to-state transitions on ϵ input.
- ◆ These transitions are done spontaneously, without looking at the input string.
- ◆ A convenience at times, but still only regular languages are accepted.

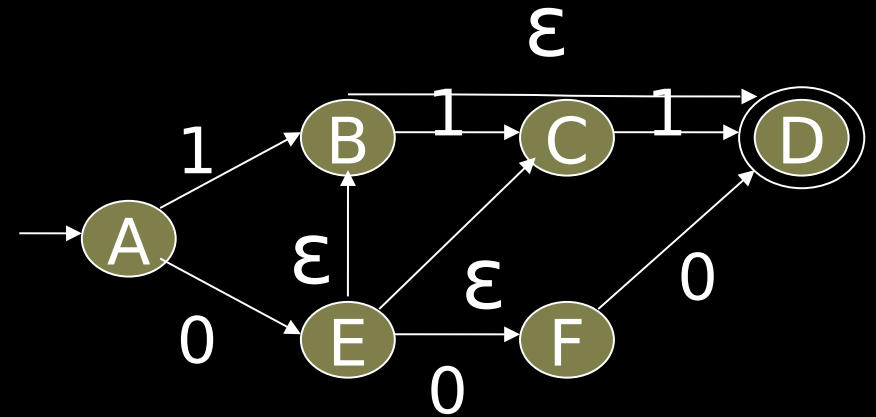
Example: ϵ -NFA



	0	1	ϵ
→ A	{E}	{B}	$\emptyset$
B	$\emptyset$	{C}	{D}
C	$\emptyset$	{D}	$\emptyset$
* D	$\emptyset$	$\emptyset$	$\emptyset$
E	{F}	$\emptyset$	{B, C}
F	{D}	$\emptyset$	$\emptyset$

Closure of States

- ◆ $CL(q)$ = set of states you can reach from state q following only arcs labeled ϵ .
- ◆ **Example:** $CL(A) = \{A\}$;
 $CL(E) = \{B, C, D, E\}$.



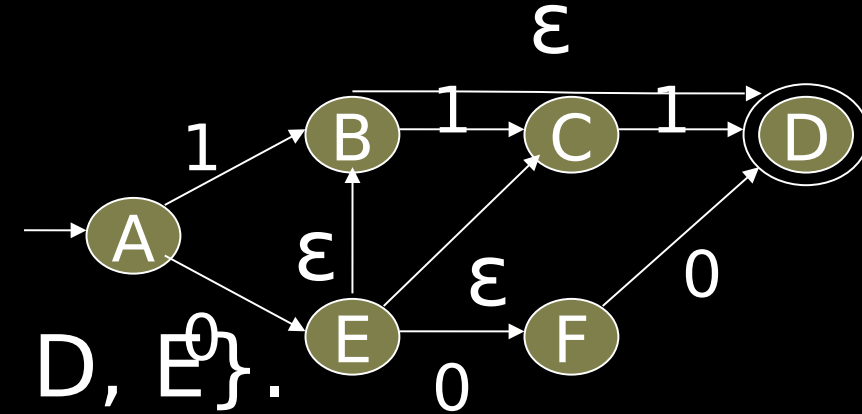
- ◆ Closure of a set of states = union of the closure of each state.

Extended Delta $\hat{\delta}$

- ◆ **Basis:** $\hat{\delta}(q, \varepsilon) = CL(q)$.
- ◆ **Induction:** $\hat{\delta}(q, xa)$ is computed as follows:
 1. Start with $\hat{\delta}(q, x) = S$.
 2. Take the union of $CL(\delta(p, a))$ for all p in S .
- ◆ **Intuition:** $\hat{\delta}(q, w)$ is the set of states you can reach from q following a path labeled w .

Example: Extended Delta

- ◆ $\delta(A, \epsilon) = CL(A) = \{A\}$.
- ◆ $\delta(A, 0) = CL(\{E\}) = \{B, C, D, E\}$.
- ◆ $\delta(A, 01) = CL(\{C, D\}) = \{C, D\}$.
- ◆ *Language* of an ϵ -NFA is the set of strings w such that (q_0, w) contains a final state.



Equivalence of NFA, ϵ -NFA

- ◆ Every NFA **is** an ϵ -NFA.
 - ◆ It just has no transitions on ϵ .
- ◆ Converse requires us to take an ϵ -NFA and construct an NFA that accepts the same language.
- ◆ We do so by combining ϵ -transitions with the transition on a real input.

Equivalence – (2)

- ◆ Start with an ε -NFA with states Q , inputs Σ , start state q_0 , final states F , and transition function δ_E .
- ◆ Construct an “ordinary” NFA with states Q , inputs Σ , start state q_0 , final states F' , and transition function δ_N .

Equivalence - (3)

Procedure to convert ϵ -NFA into NFA

Step 1: Find Closure of all states

Step 2: Extended Transition Function for all states:

$$\delta(q_0, a) = \text{CL}(\delta(\text{CL}(q_0), a))$$

Step 3: Set of all final states F' consists of all states whose E-Closure contains a final state in F

Interesting
closures:

$CL(B) = \{B, D\}$:

$CL(E) = \{B, C, D, E\}$

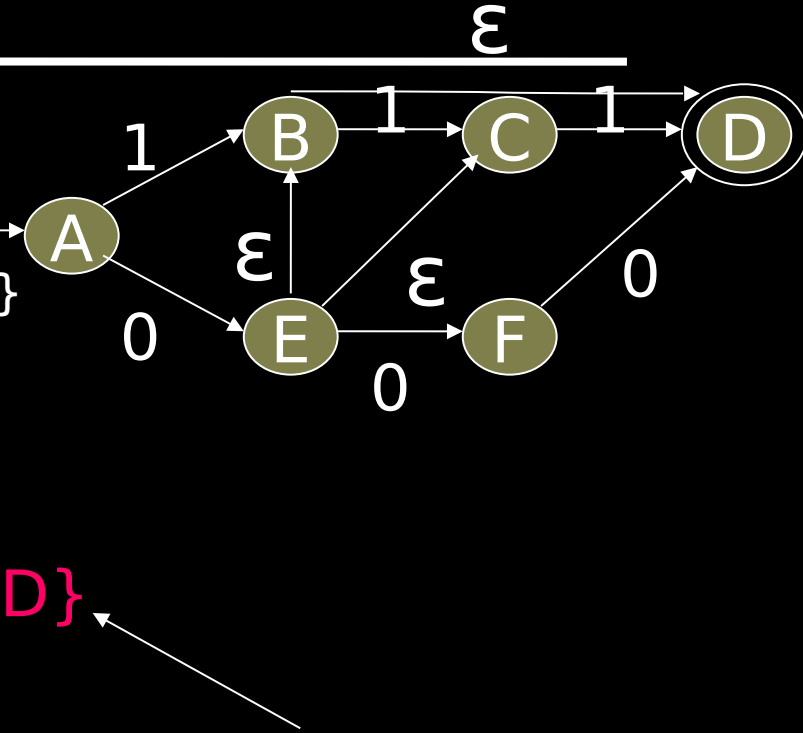
Example: ϵ -NFA-to-NFA

	0	1	ϵ
→ A	{E}	{B}	$\emptyset$
B	$\emptyset$	{C}	{D}
C	$\emptyset$	{D}	$\emptyset$
* D	$\emptyset$	$\emptyset$	$\emptyset$
* E	{F}	$\emptyset$	{B, C}
F	{D}	$\emptyset$	$\emptyset$

ϵ -NFA

	0	1
→ A	{B, C, D, E}	{B, D}
* B	$\emptyset$	{C}
C	$\emptyset$	{D}
* D	$\emptyset$	$\emptyset$
* E	{F}	{C, D}
F	{D}	$\emptyset$

NFA without ϵ -Transition

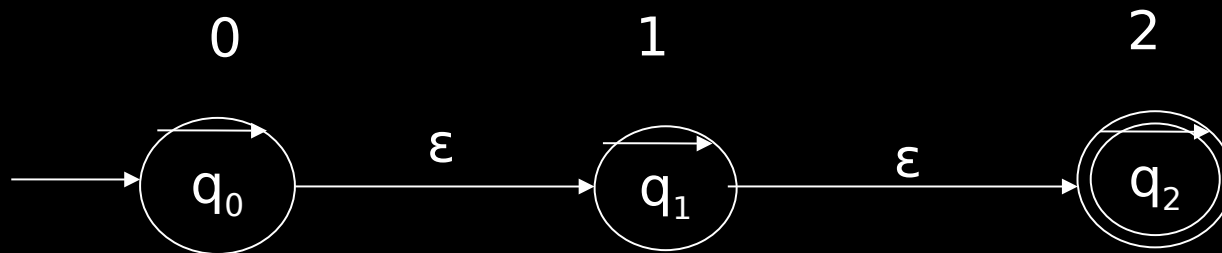


Since closures of B and E include final state D.

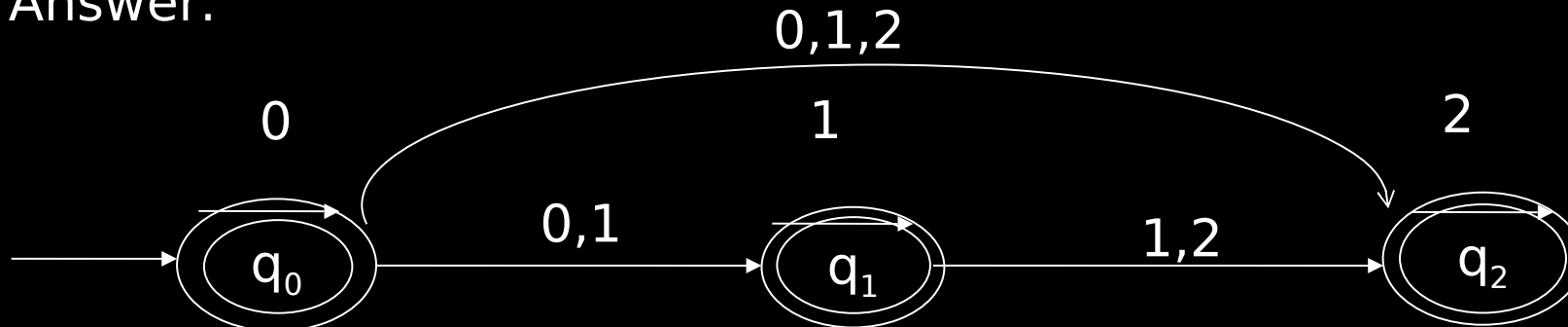
Since closure of E includes B and C; which have transitions on 1 to C and D.

Additional Examples ...

1. Convert the following NFA with ϵ -transitions into NFA without ϵ -transitions.



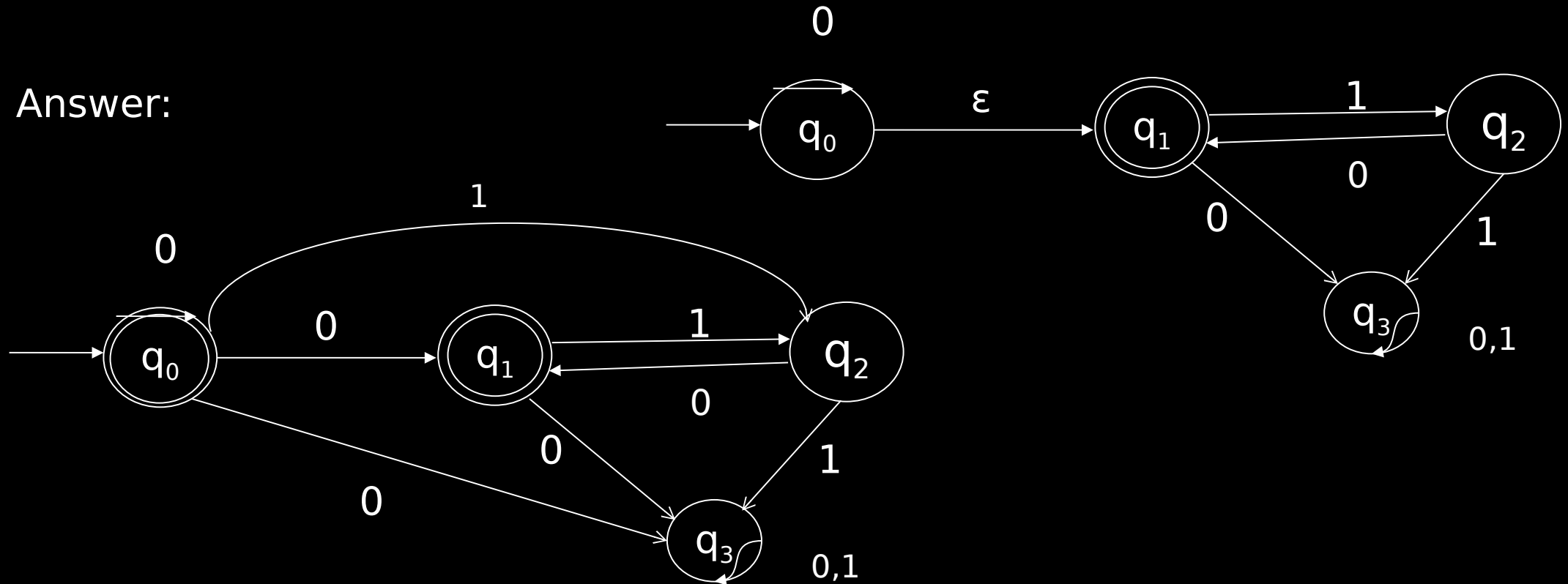
Answer:



Additional Examples ...

2. Convert the following NFA with ϵ -transitions into NFA without ϵ -transitions.

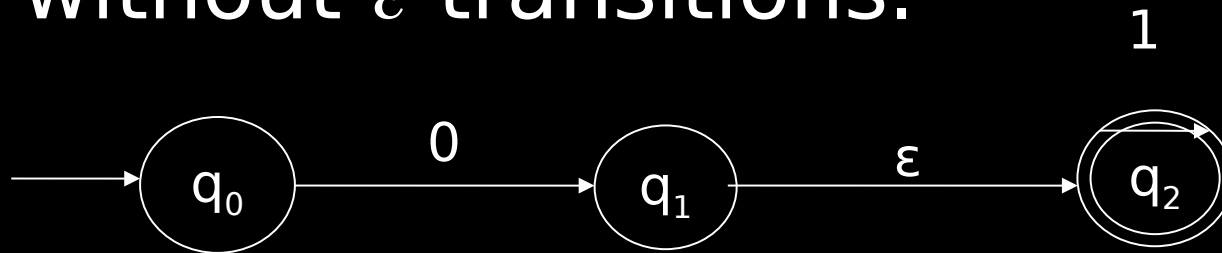
Answer:



Exercise

Convert the following NFA with ε -transitions into NFAs without ε -transitions.

a)



β) ε -NFA with the following transition table

	0	1	ε
→ A	{A}	$\emptyset$	{B}
B	$\emptyset$	{C}	{D}
C	{D}	$\emptyset$	$\emptyset$
* D	$\emptyset$	{C}	{B}

Summary

- ◆ DFA's, NFA's, and ϵ -NFA's all accept exactly the same set of languages: the **regular languages**.
- ◆ The NFA types are easier to design and may have exponentially fewer states than a DFA.
- ◆ But only a DFA can be **implemented!**

Minimization of DFA

◆ Steps in constructing minimum automata

Step 1: Remove all inaccessible states.

- ▶ This can be done by enumerating all simple paths of the graph from the initial to the final. Any state which is not part of some path is inaccessible.

Step 2: Construct partition zero (π_0)

- ▶ Partition the states of an automata as a set of final states and non-final states

$$\pi_0 = \{Q1, Q2\}$$

Minimization ...

Step 3: Construction of $\pi_1, \pi_2, \dots, \pi_n$

- Subsets of π_{i+1} are obtained by partitioning π_i
- If $\delta(q1, a) = X$ and $\delta(q2, a) = Y$ for $a \in \Sigma$
 - If X and Y are in same subset of π_i , $q1$ and $q2$ are equivalent, else not.
 - If $q1$ and $q2$ are not equivalent, separate them.

REPEAT STEP 3 Until all pairs are marked.

STOP Partitioning if $\pi = \pi$

Minimization ...

Step 4: Find the set of all equivalent states and for each set, create a state labeled by the set.

Step 5: For each transition rule of M of the form

$\delta(q_r, a) = q_p$, find the sets to which q_r and q_p belong.

Let q_r is in set X and q_p in Y , then add a transition rule

$\delta(x, a) = Y$

- ◆ The initial state is the state whose label includes q_0 .
- ◆ The states that contain a final state are final states.

Examples

Minimize the following Automata

a)

	0	1
- >q0	q1	q3
q1	q2	q4
q2	q1	q4
q3	q2	q4
*q4	q4	q4

$$\pi_0 = \{q4\}, \{q0, q1, q2, q3\}$$

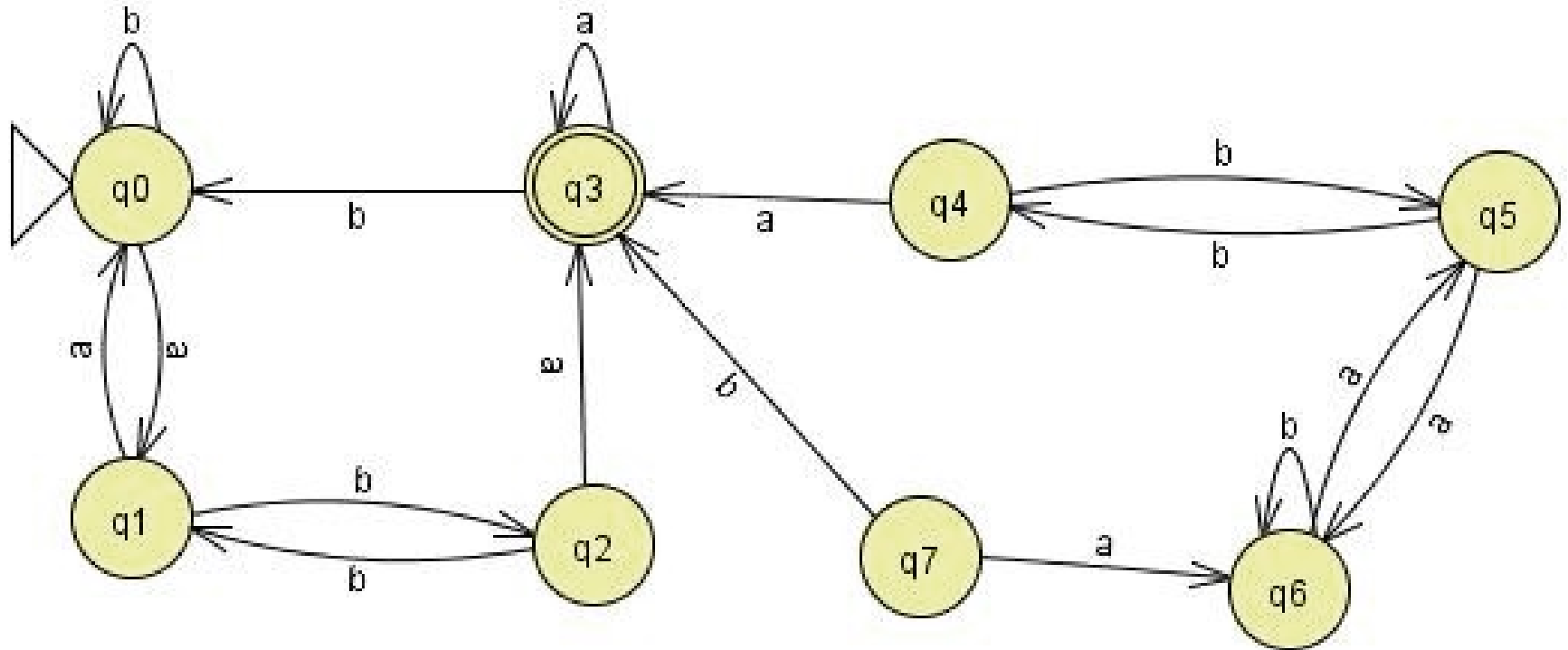
$$\pi_1 = \{q4\}, \{q0\}, \\ \{q1, q2, q3\}$$

$$\pi_2 = \{q4\}, \{q0\}, \\ \{q1, q2, q3\}$$

Examples ...

Minimize the following Automata

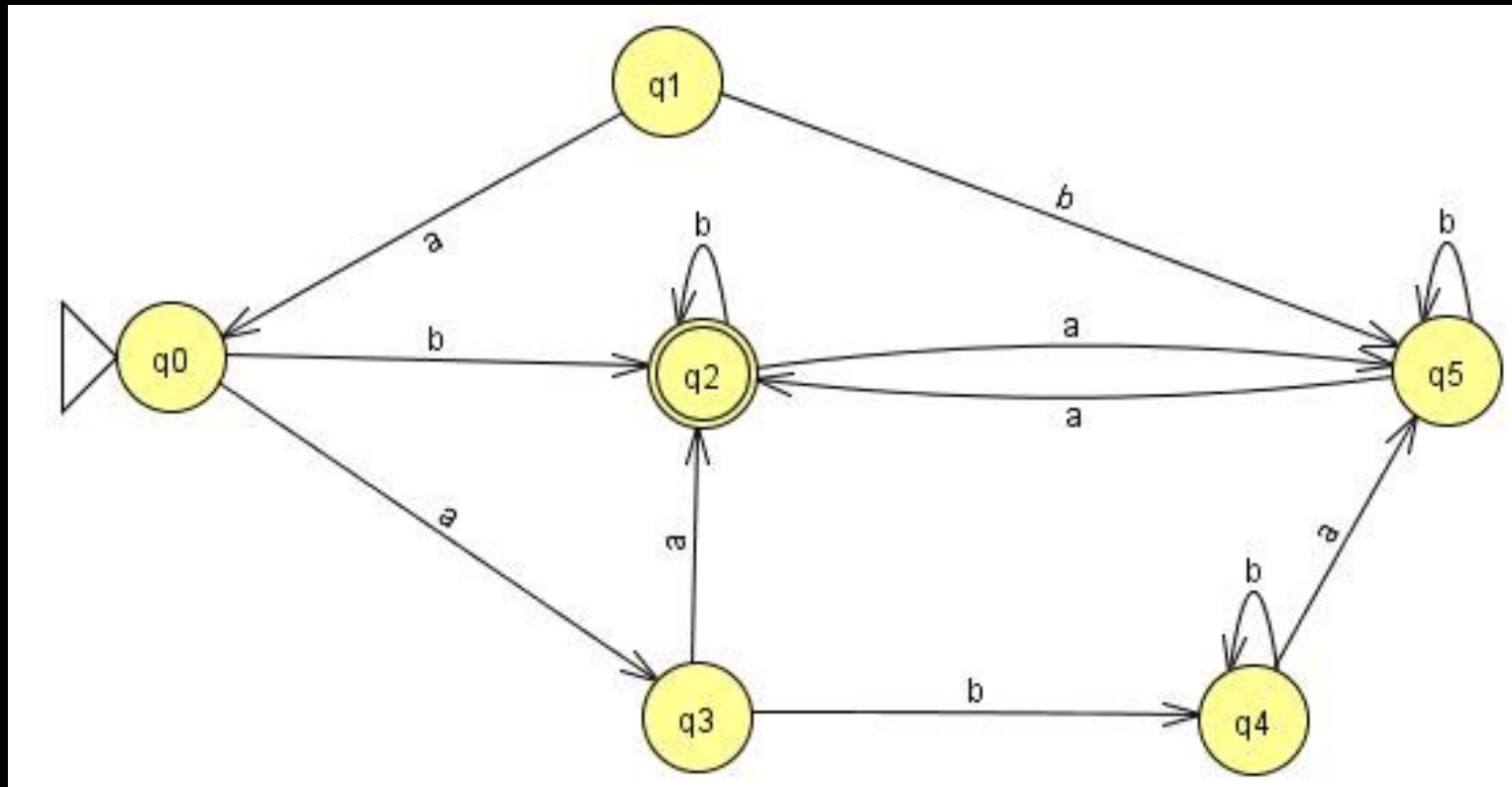
b)



Examples ...

Minimize the following Automata

c)



Examples ...

Minimize the following Automata

d)

	0	1
- >q0	q1	q4
q1	q2	q3
q2	q2	q3
*q3	q5	q6
q4	q5	q6
q5	q2	q3
q6	q5	q6