23MT2014

THEORY OF COMPUTATION

Topic:

NFA-DFA: EQUIVALENCE OF MACHINES

Session - 5

AIM OF THE SESSION

The course aims to provide students with a deep understanding of NFAs and their ability to recognize and accept regular languages.

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. To familiarize students with the concept of regular languages and their relationship with NFAs.
2. To enable students to design and construct NFAs that recognize and accept specific regular languages.
3. To provide students with the ability to analyze and prove the equivalence of NFAs and regular expressions.

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Understand the concept of regular languages and their properties.
2. Design and construct NFAs that recognize and accept specific regular languages.
3. Analyze and prove the equivalence of NFAs and regular expressions in recognizing regular languages.

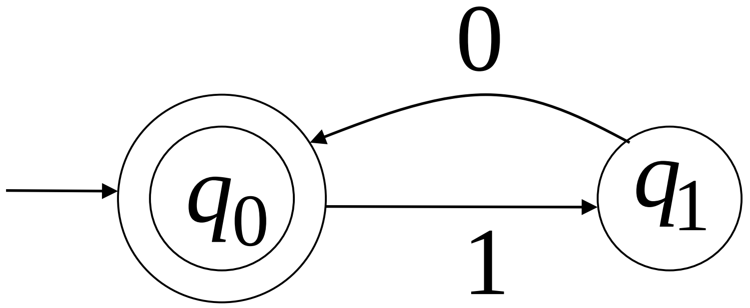
Equivalence of Machines

- Definition for Automata:
- Machine M_1 is equivalent to machine M_2
- if $L(M_1) = L(M_2)$

Example of equivalent machines

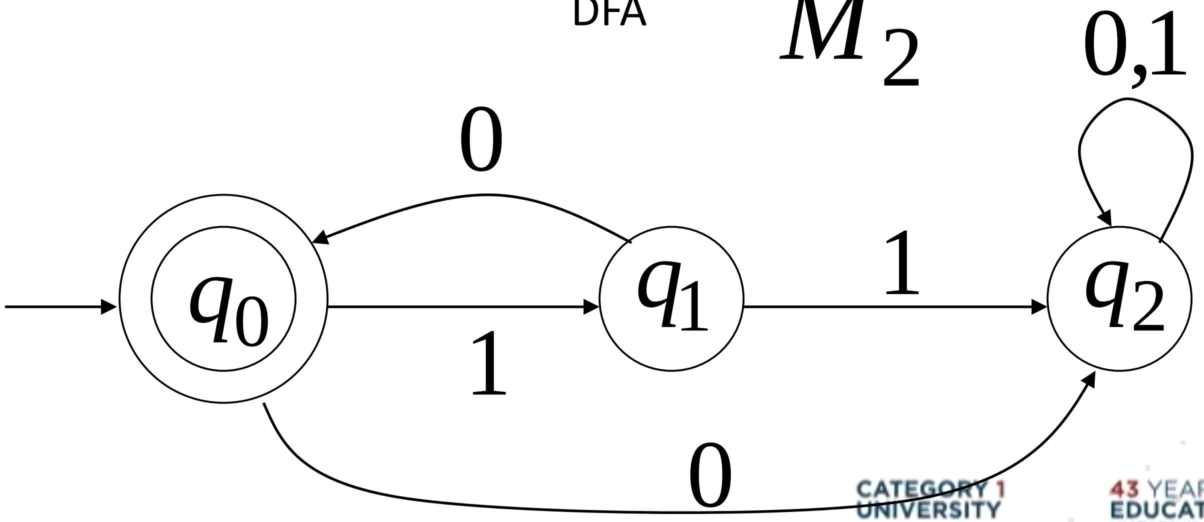
$$L(M_1)=\{10\}^*$$

NFA M_1



$$L(M_2)=\{10\}^*$$

DFA M_2



We will prove:

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{accepted} \\ \text{by NFAs} \end{array} \right\} = \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Languages
accepted
by DFAs

NFAs and DFAs have the same computation power

Step 1

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{accepted} \\ \text{by NFAs} \end{array} \right\} \supseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Proof: Every DFA is trivially an NFA

$\Downarrow$

Any language L accepted by a DFA is also accepted by an NFA

Step 2

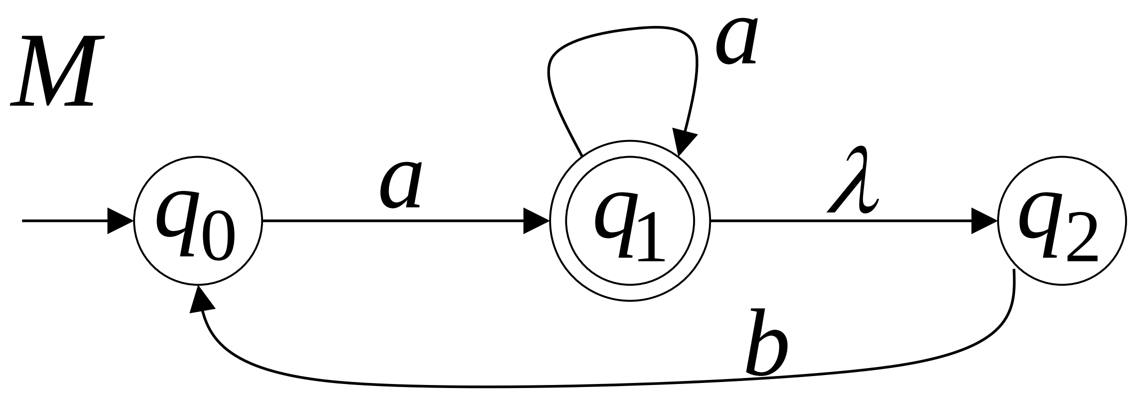
$$\left\{ \begin{array}{l} \text{Languages} \\ \text{accepted} \\ \text{by NFAs} \end{array} \right\} \subseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Proof: Any NFA can be converted to an equivalent DFA

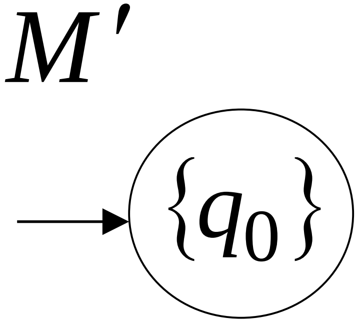
Any language L accepted by an NFA is also accepted by a DFA

Convert NFA to DFA

- NFA

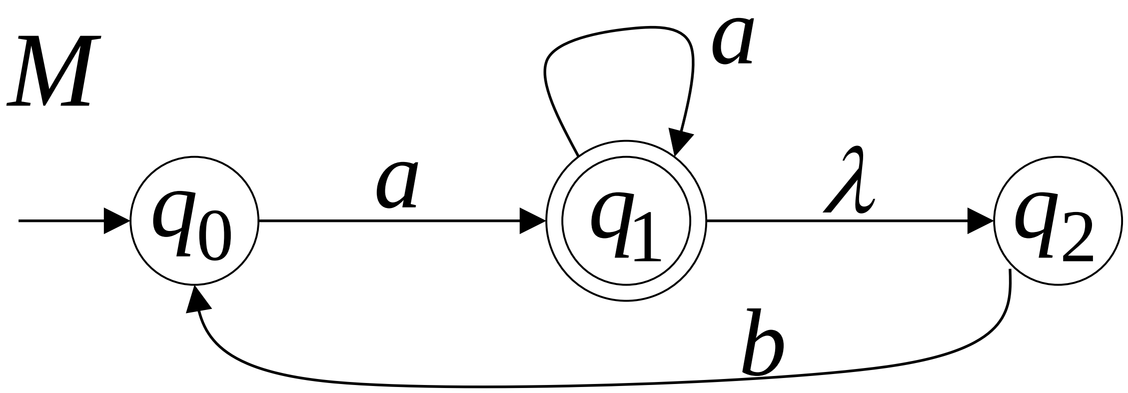


DFA

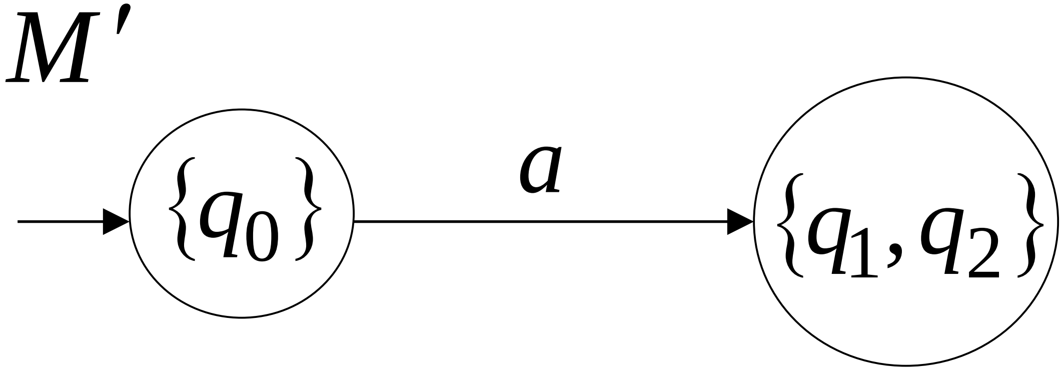


Convert NFA to DFA

- NFA

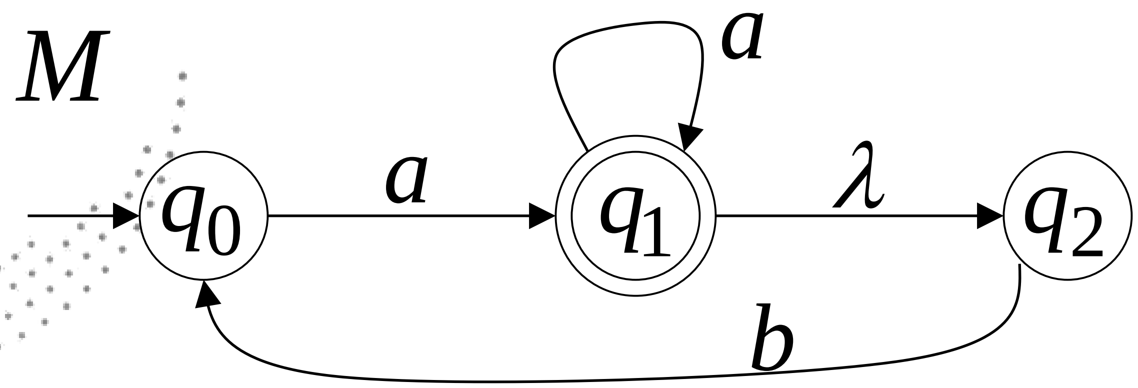


DFA

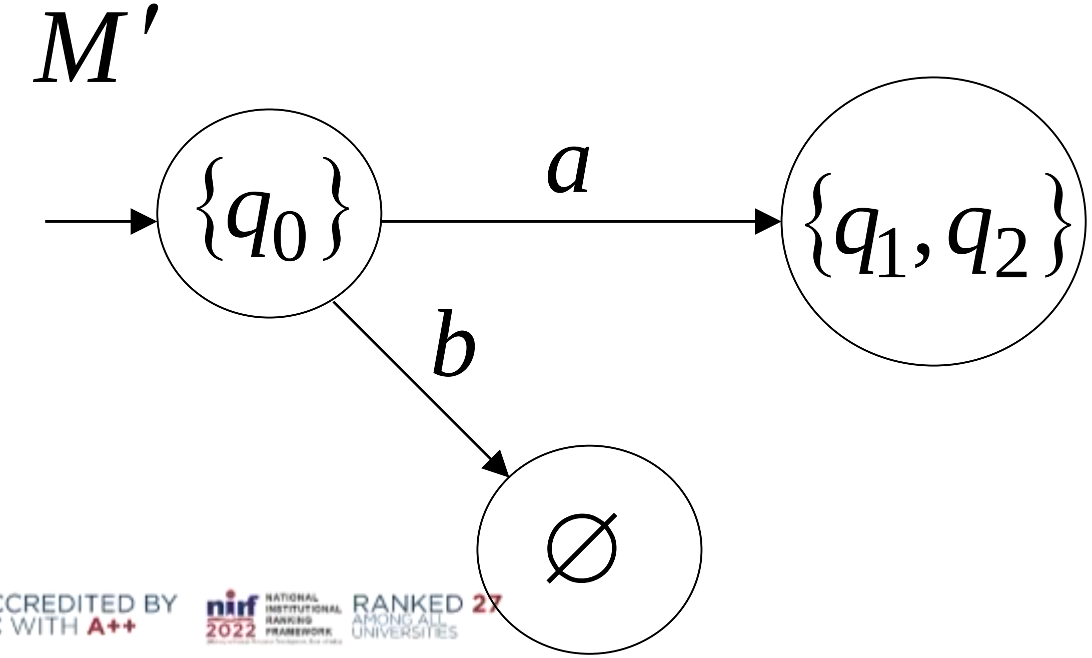


Convert NFA to DFA

- NFA

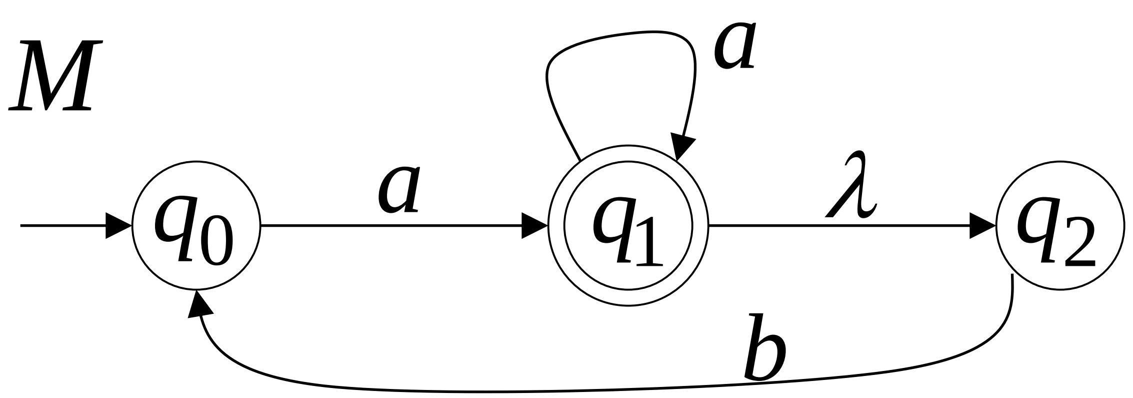


DFA

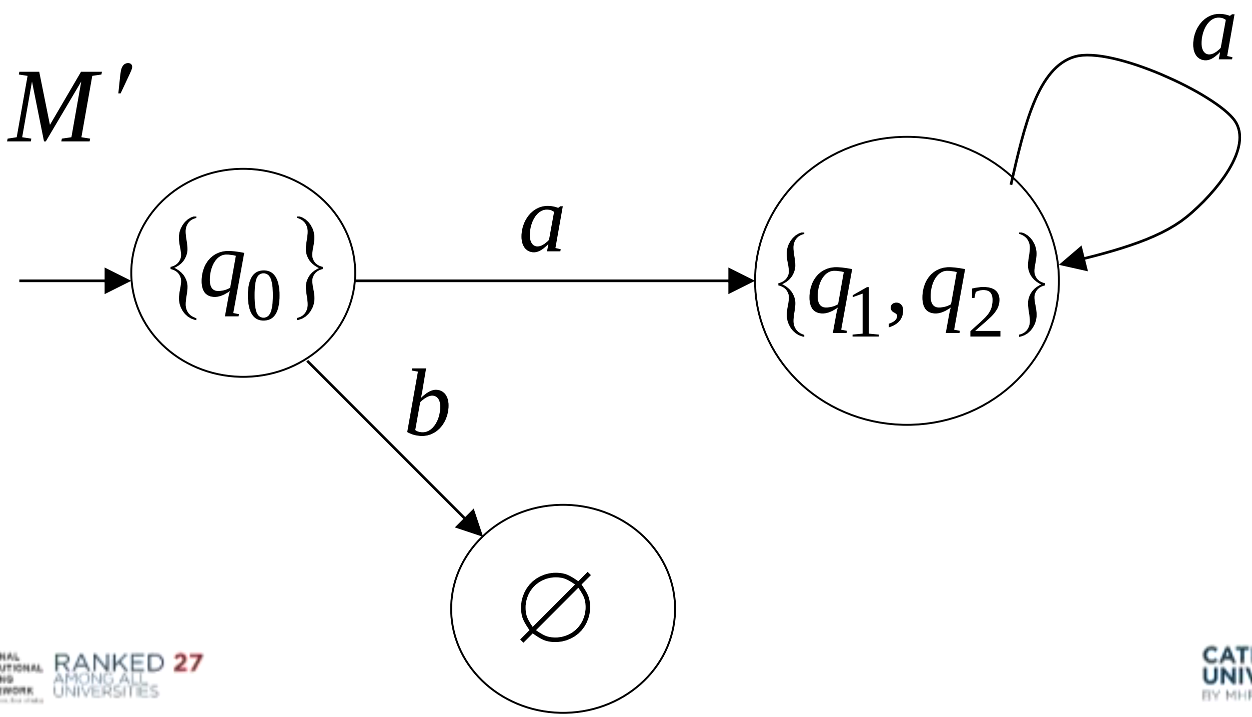


Convert NFA to DFA

- NFA

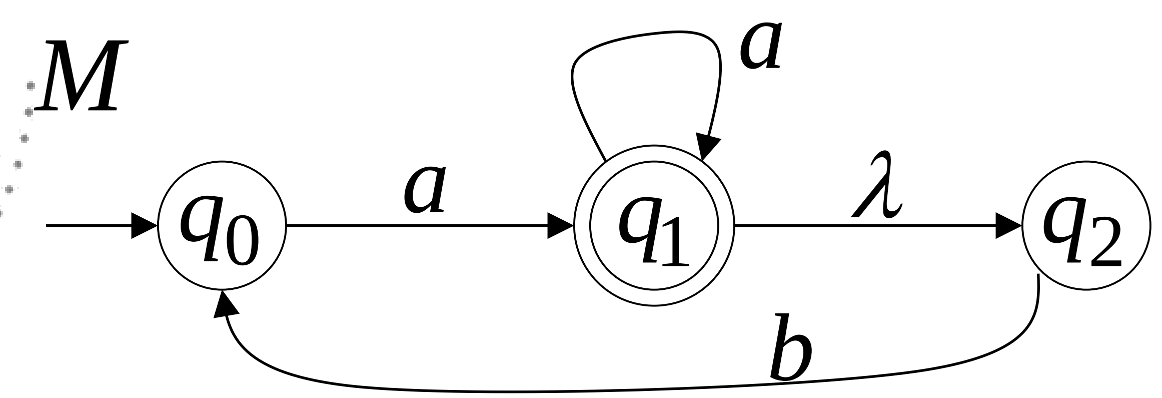


DFA

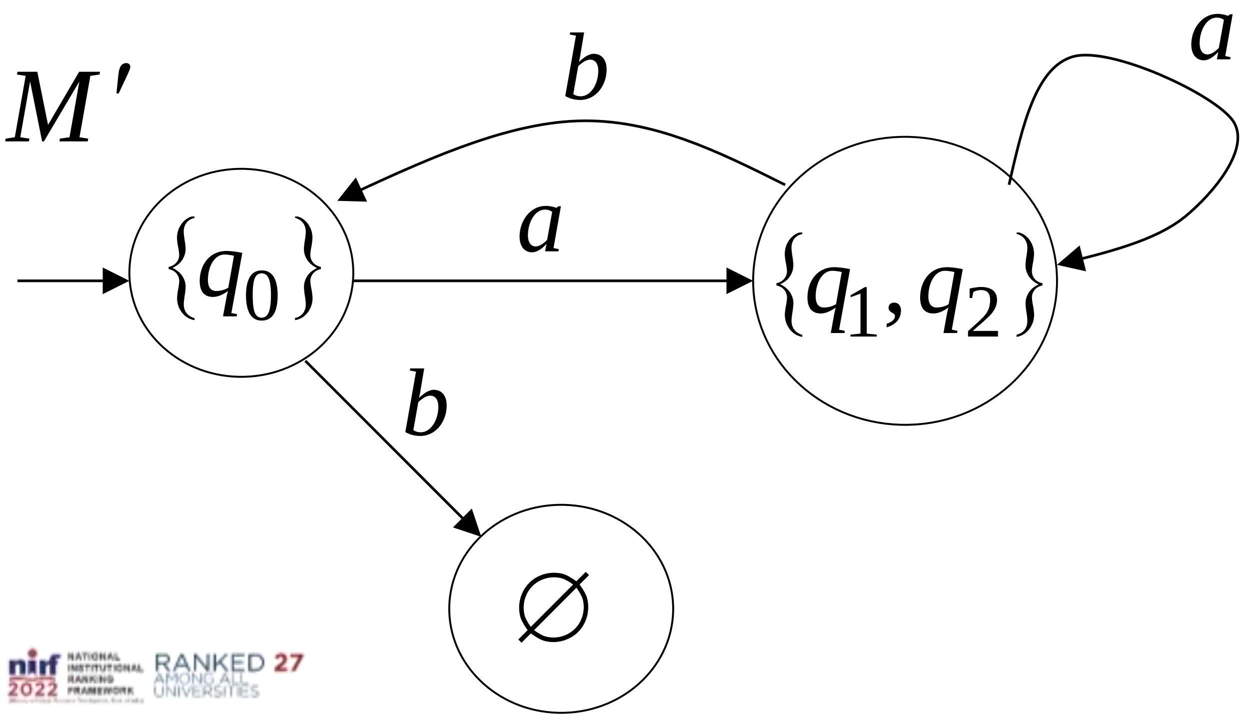


Convert NFA to DFA

- NFA M

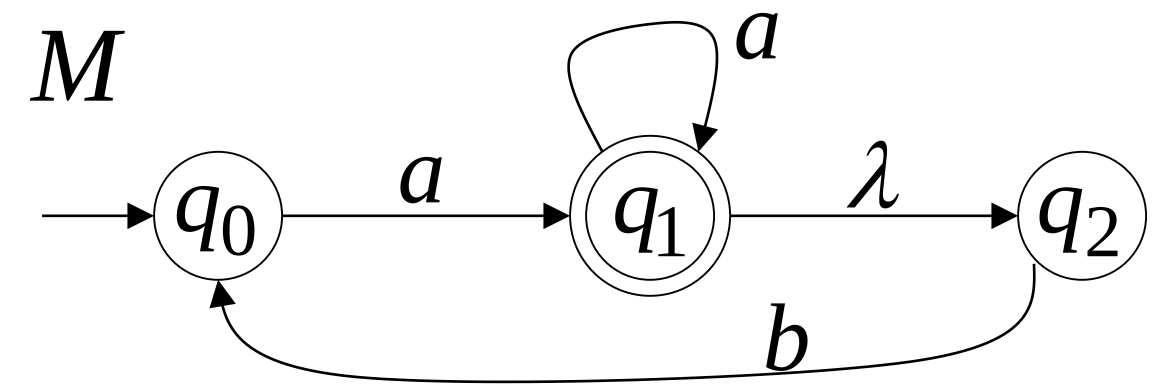


DFA M'

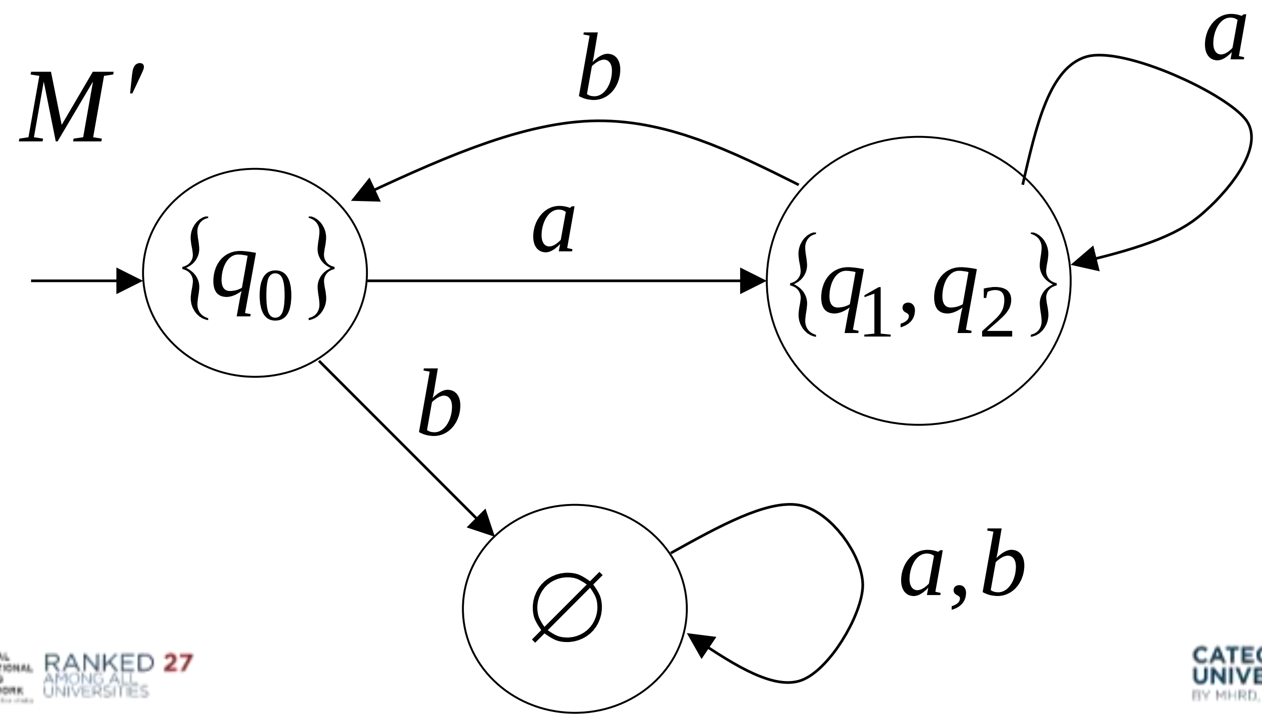


Convert NFA to DFA

- NFA

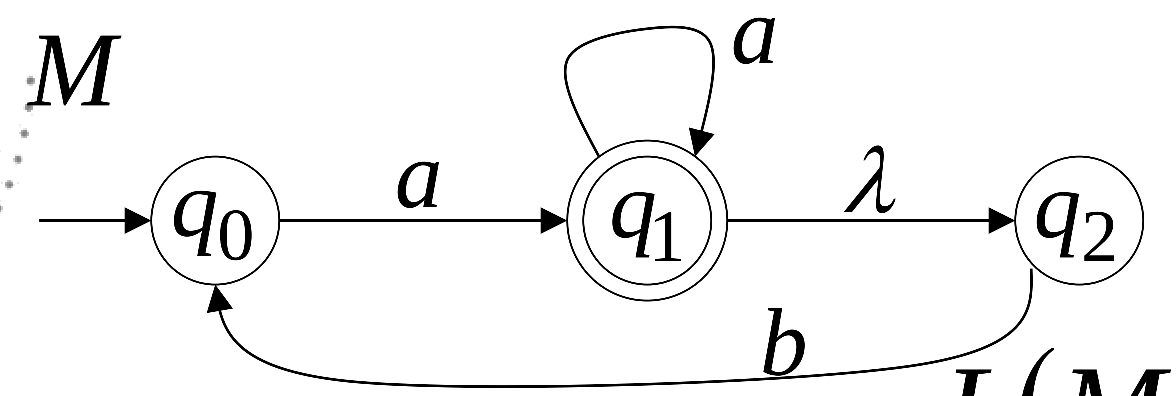


DFA



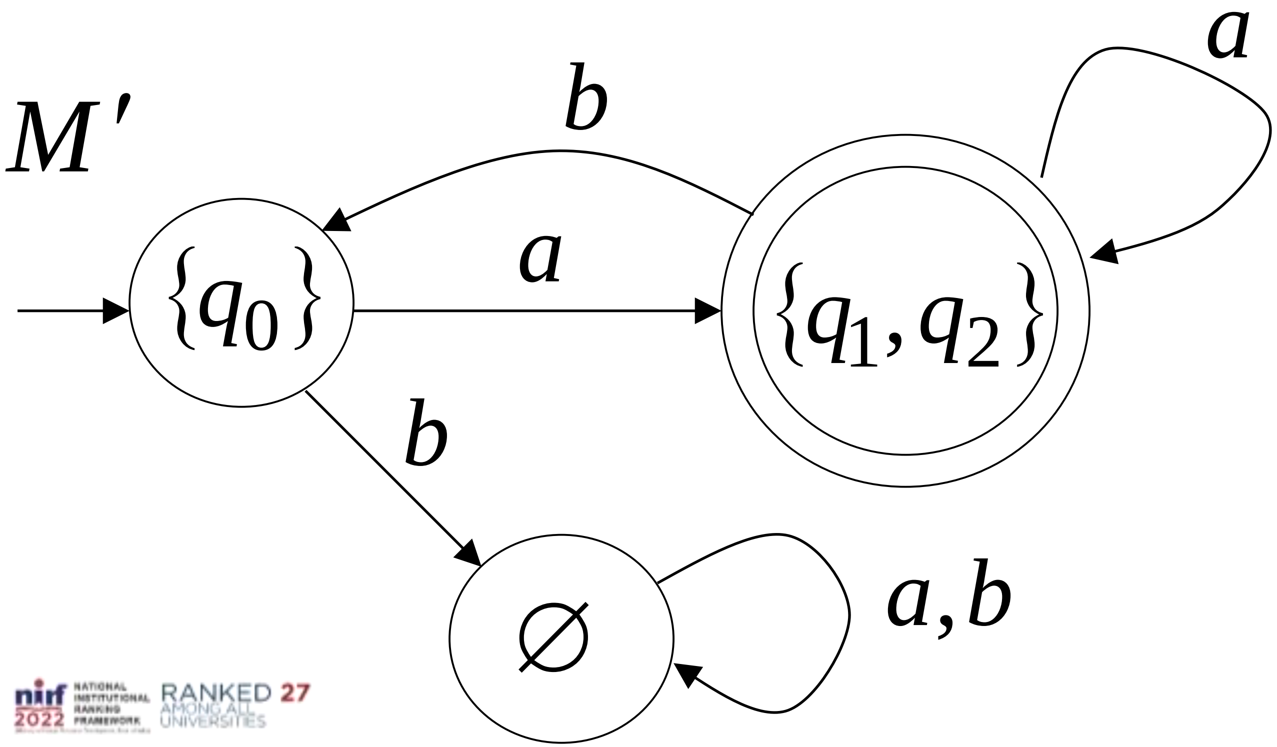
Convert NFA to DFA

- NFA



$$L(M) = L(M')$$

DFA



NFA to DFA: Remarks

- We are given an NFA $M = (Q_N, \Sigma, \delta_N, q_0, F_N)$

- We want to convert it
- to an equivalent DFA $M' = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$

- With

$$L(M) = L(M')$$

- If the NFA has *n* states

$$Q_N = \{q_0, q_1, q_2, \ldots, q_n\}$$

- the DFA has 2^n states in the power set of Q_N
- i.e.

$$\emptyset, \{q_0\}, \{q_1\}, \{q_1, q_2\}, \{q_3, q_4, q_7\}, \ldots.$$

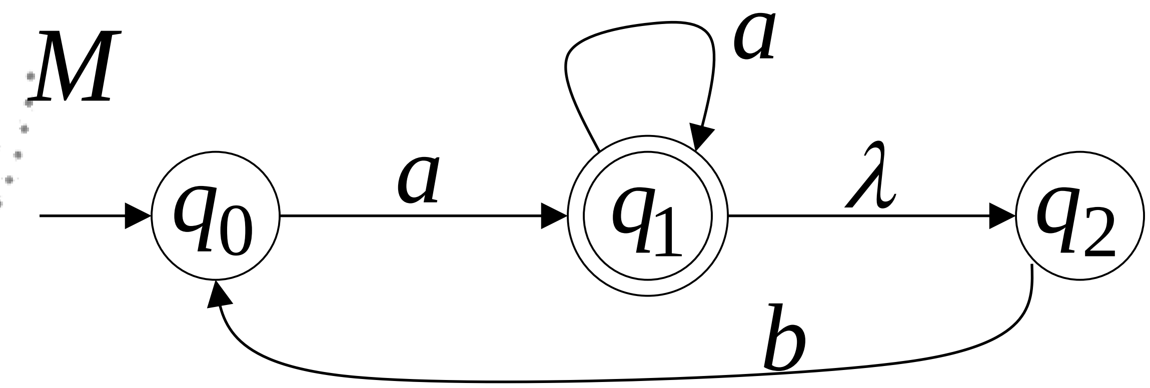
Procedure NFA to DFA

- **1.** Initial state of NFA: q_0

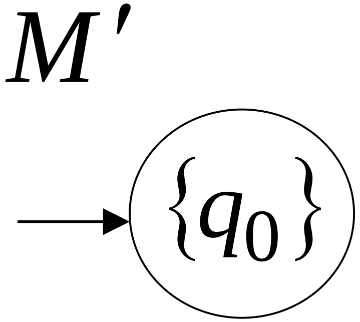
- Initial state of DFA: $\{q_0\}$

Example

- NFA



DFA



Procedure NFA to DFA

- **2.** For every DFA's state $\{q_i, q_j, \ldots, q_m\}$

- Compute in the NFA

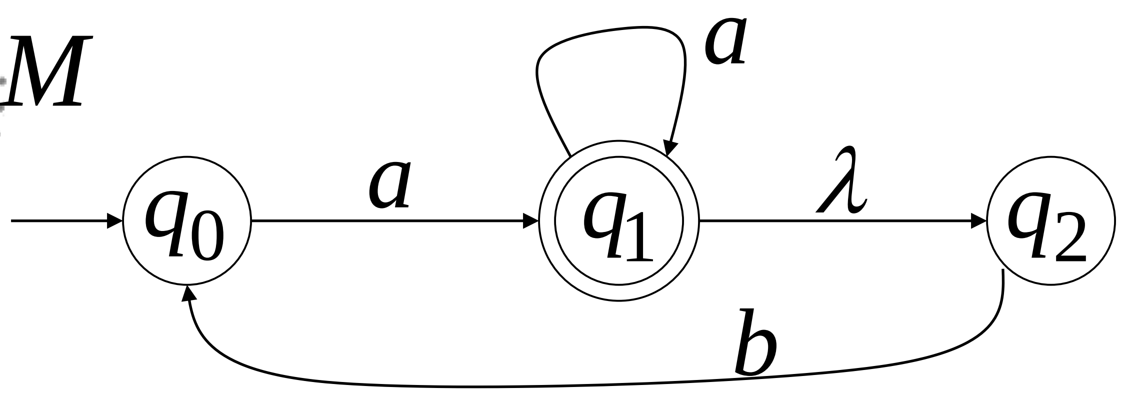
$$\left.\begin{array}{l} \delta^*(q_i, a), \\ \delta^*(q_j, a), \\ \ldots \end{array}\right\} = \{q'_i, q'_j, \ldots, q'_m\}$$

- Add transition to DFA

$$\delta(\{q_i, q_j, \ldots, q_m\},\ a) = \{q'_i, q'_j, \ldots, q'_m\}$$

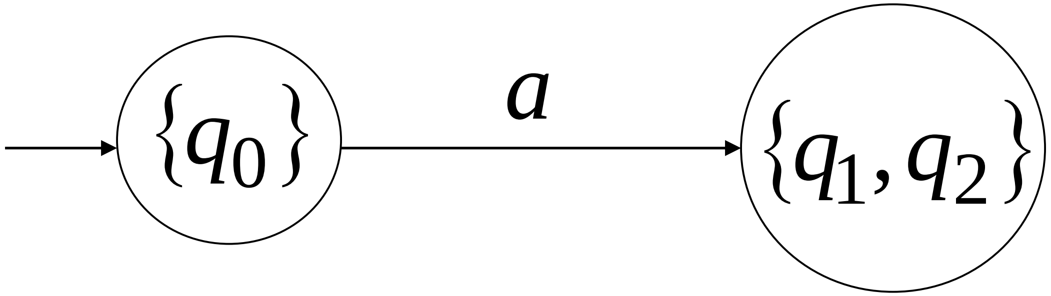
Example

- NFA M



$$\delta^*(q_0, a) = \{q_1, q_2\}$$

DFA M'



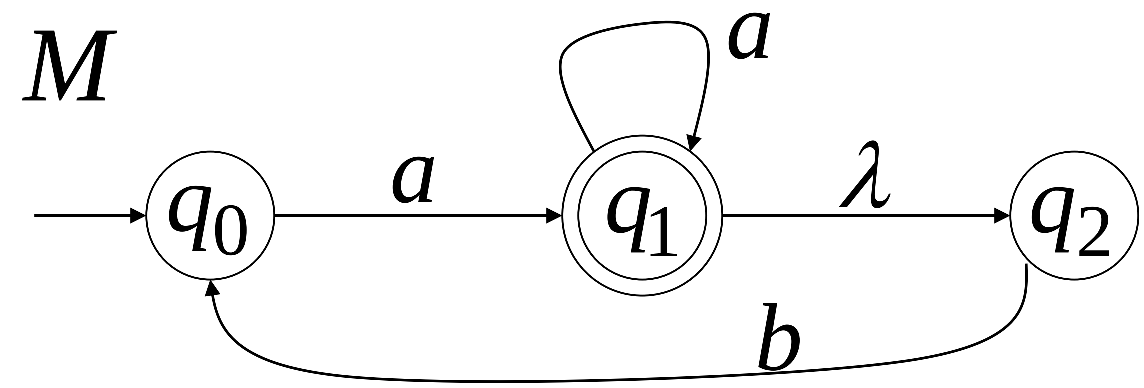
$$\delta(\{q_0\}, a) = \{q_1, q_2\}$$

Procedure NFA to DFA

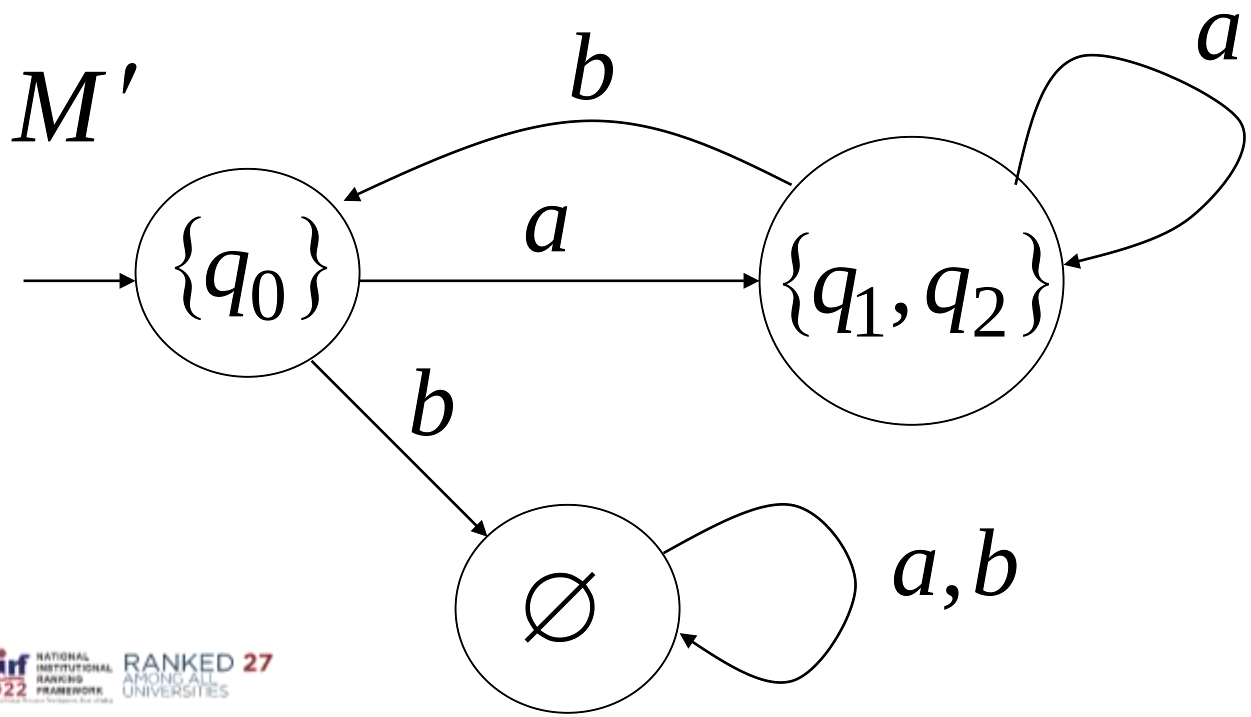
- Repeat Step 2 for all letters in alphabet,
- until
- no more transitions can be added.

Example

- NFA



DFA



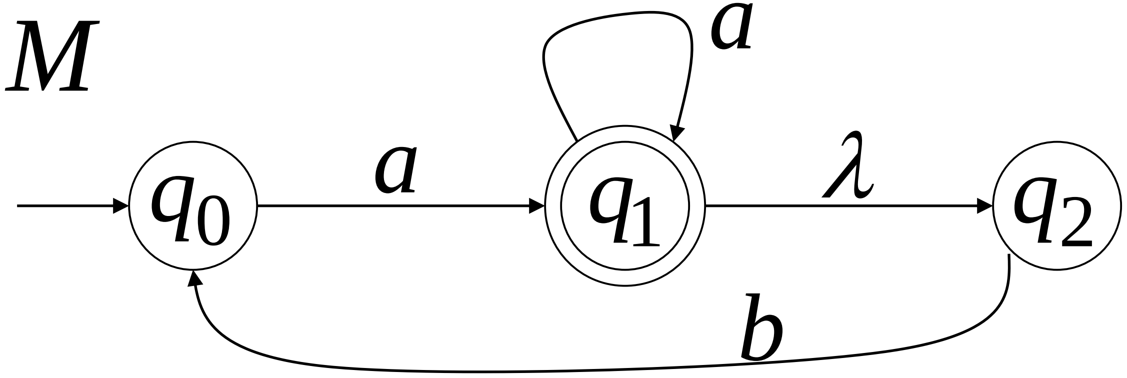
Procedure NFA to DFA

- **3.** For any DFA state $\{q_i, q_j, \ldots, q_m\}$
- If some q_j is a final state in the NFA
- Then,
- $\{q_i, q_j, \ldots, q_m\}$
- is a final state in the DFA

Example

- NFA

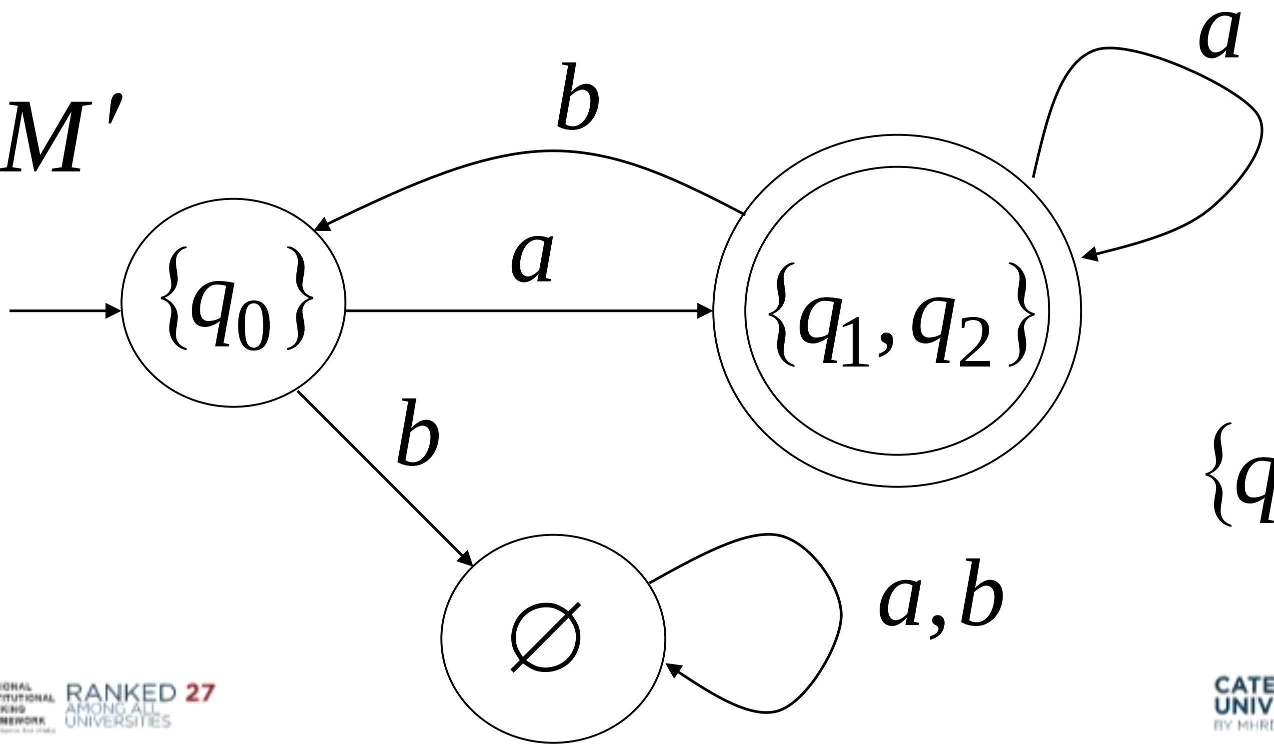
M



$q_1 \in F$

DFA

M'



$\{q_1, q_2\} \in F'$

Theorem

- Take NFA M

Apply procedure to obtain DFA M'

Then M and M' are equivalent :

$$L(M) = L(M')$$

Proof

$$L(M) = L(M')$$

$$\Updownarrow$$

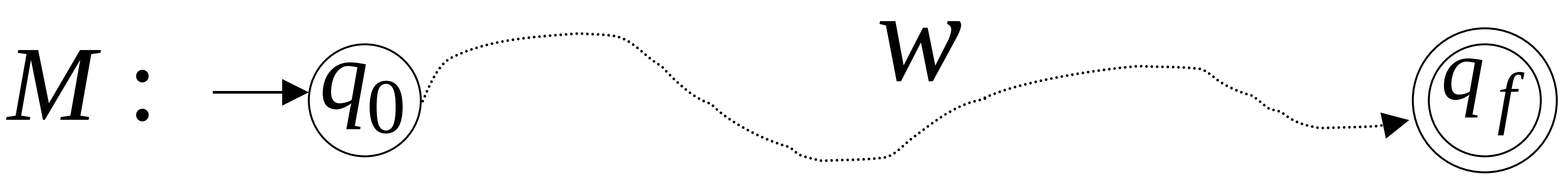
$$L(M) \subseteq L(M') \quad \text{AND} \quad L(M) \supseteq L(M')$$

First we show: $$L(M) \subseteq L(M')$$

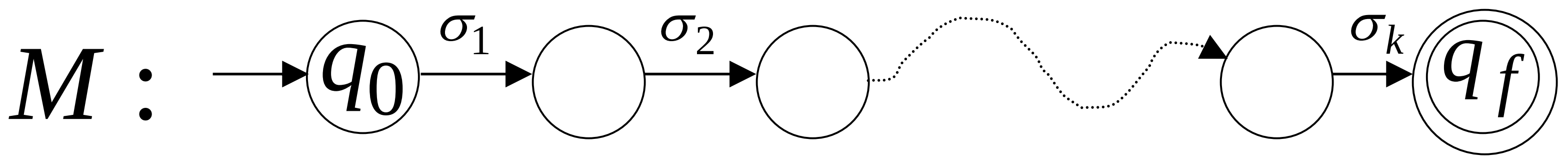
Take arbitrary: $$w \in L(M)$$

We will prove: $$w \in L(M')$$

$$w \in L(M)$$



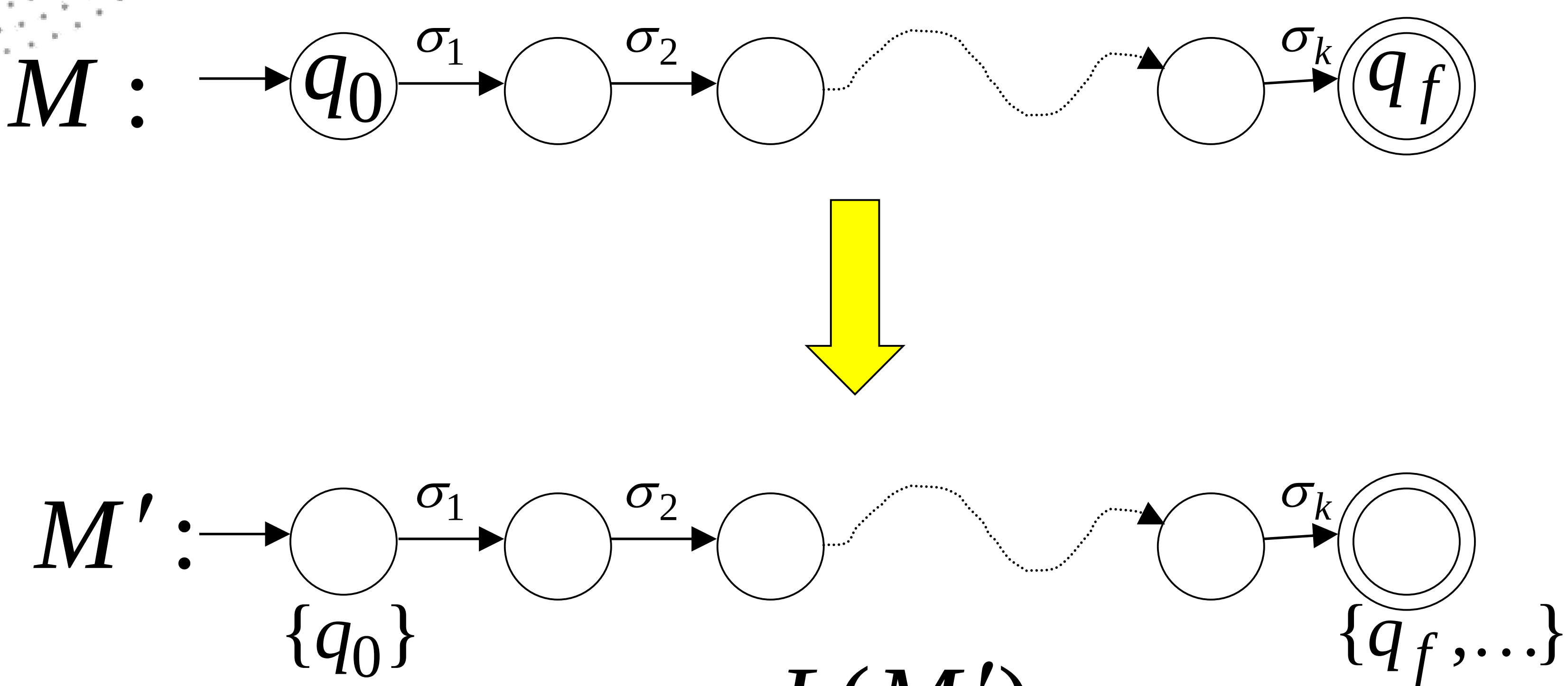
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



We will show that if

$$w \in L(M)$$

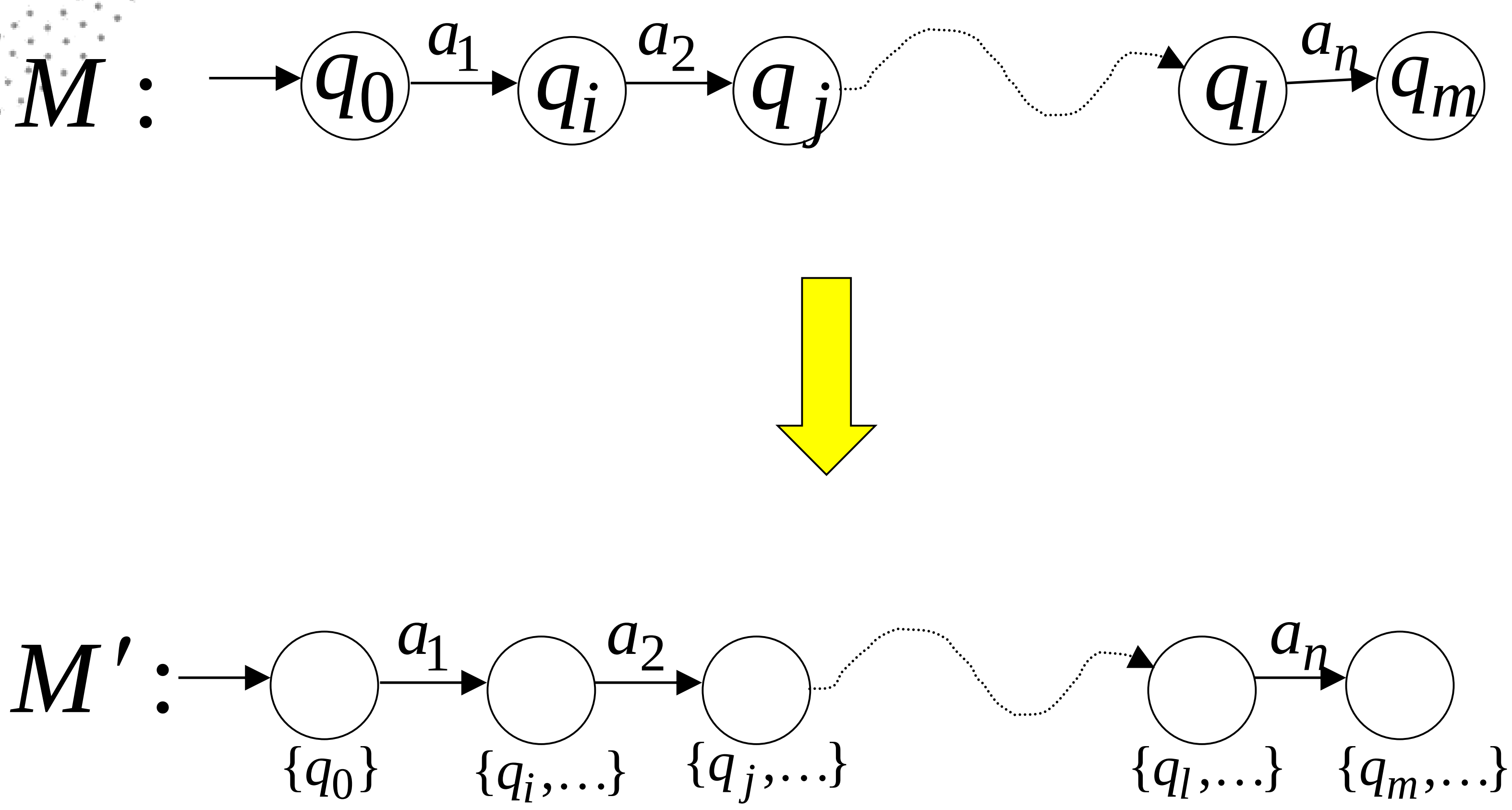
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



$$w \in L(M')$$

More generally, we will show that if in M:

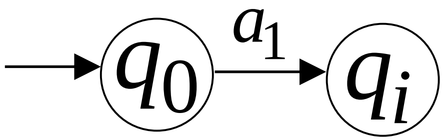
(arbitrary string) $v = a_1 a_2 \cdots a_n$



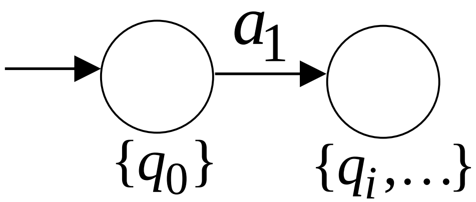
Proof by induction on $|v|$

Induction Basis: $v = a_1$

$M :$



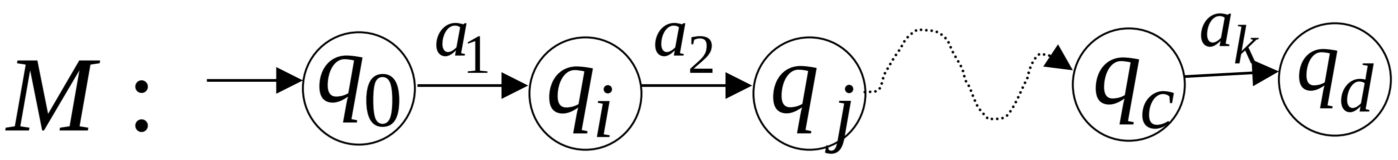
$M' :$

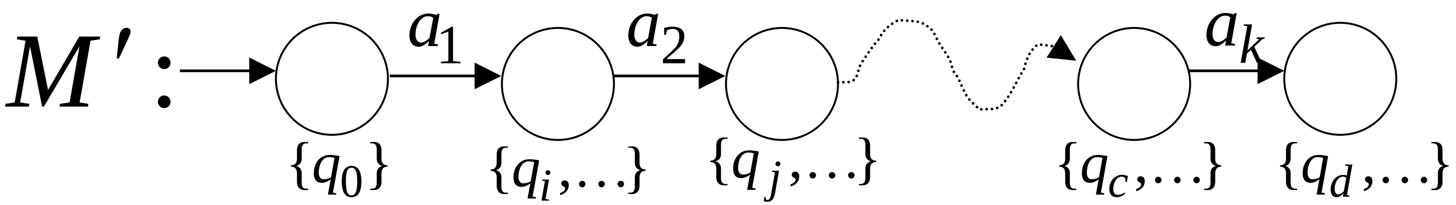


Induction hypothesis:

$$1 \leq |v| \leq k$$

$$v = a_1 a_2 \cdots a_k$$

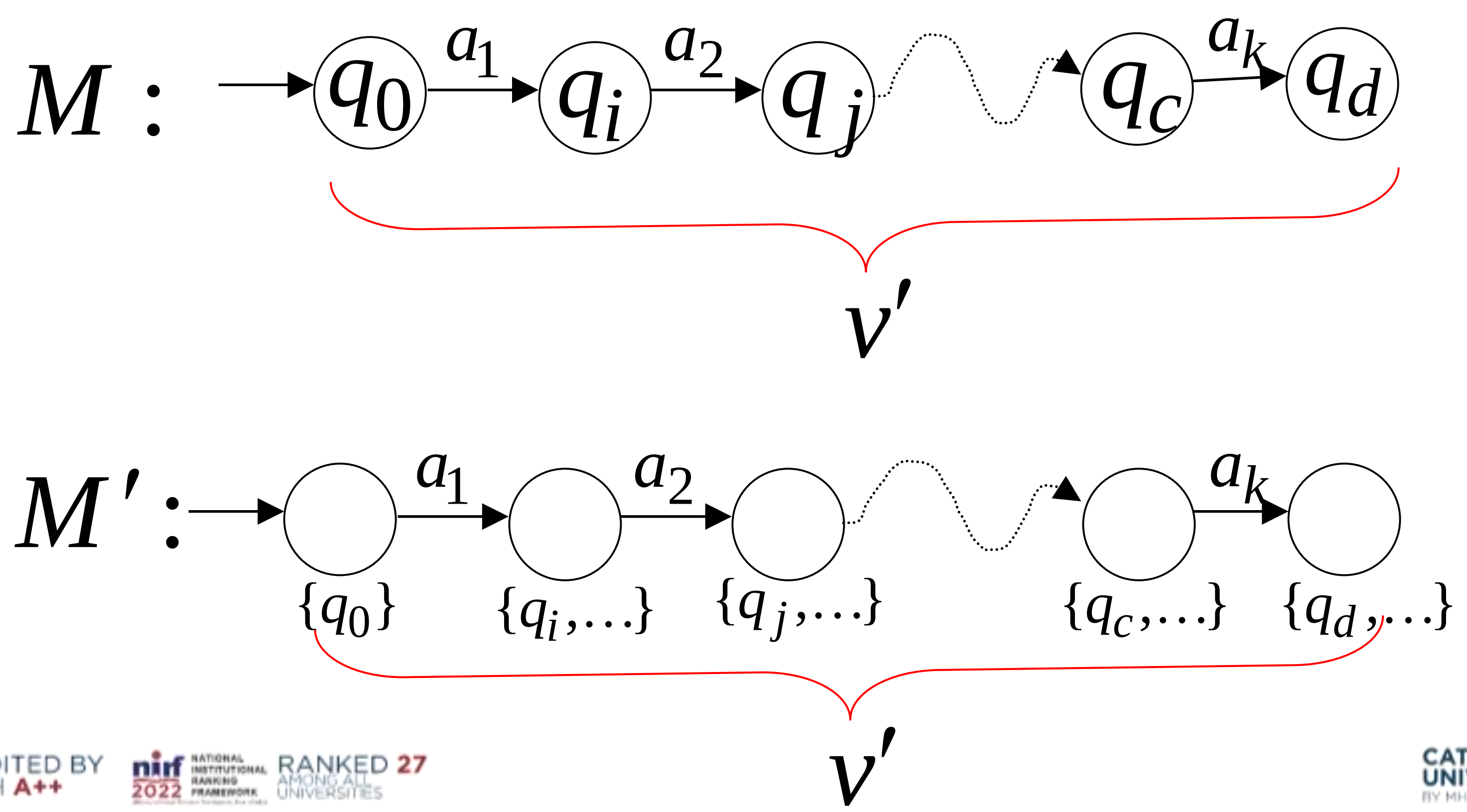




Induction Step:

$$|v| = k+1$$

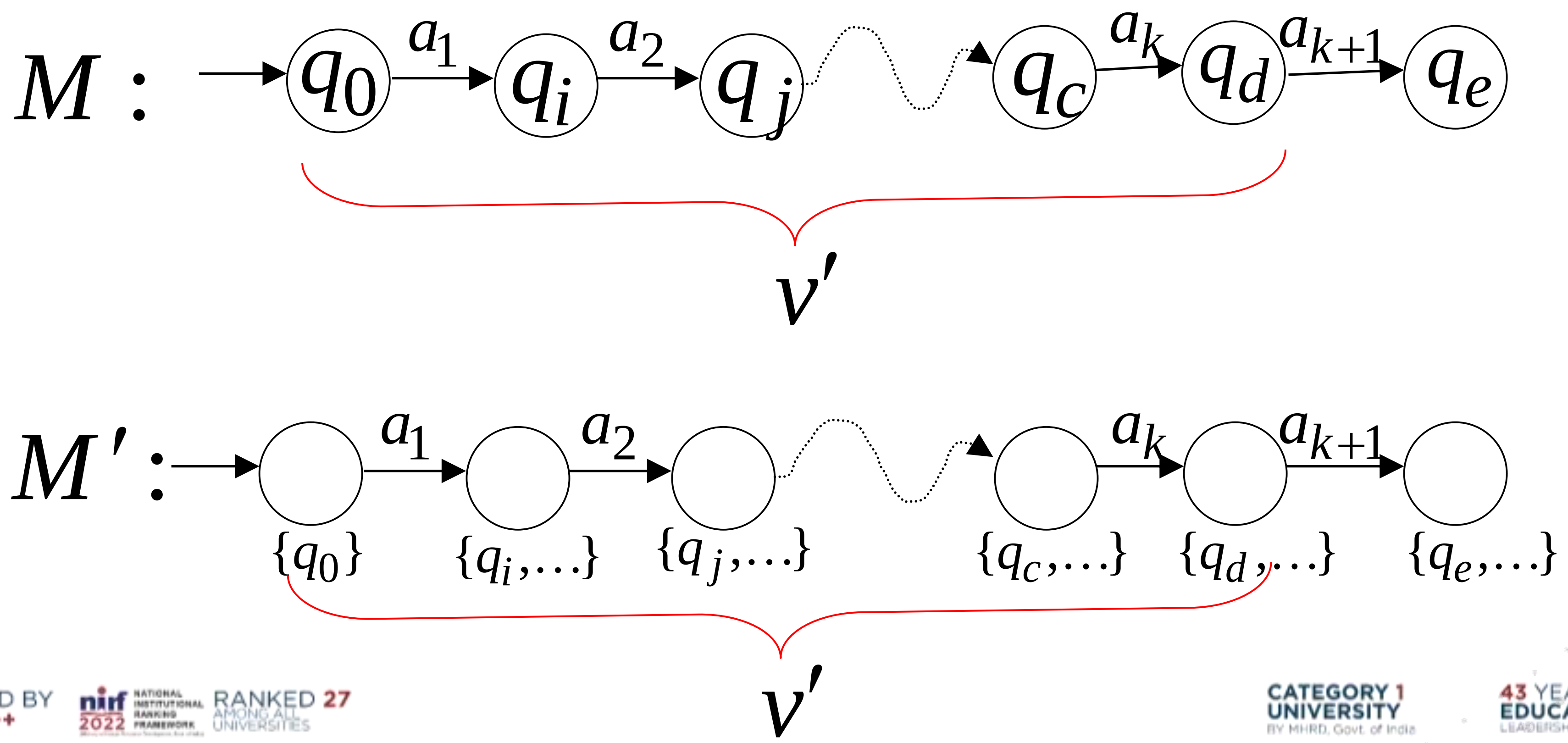
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$



Induction Step:

$$|v| = k+1$$

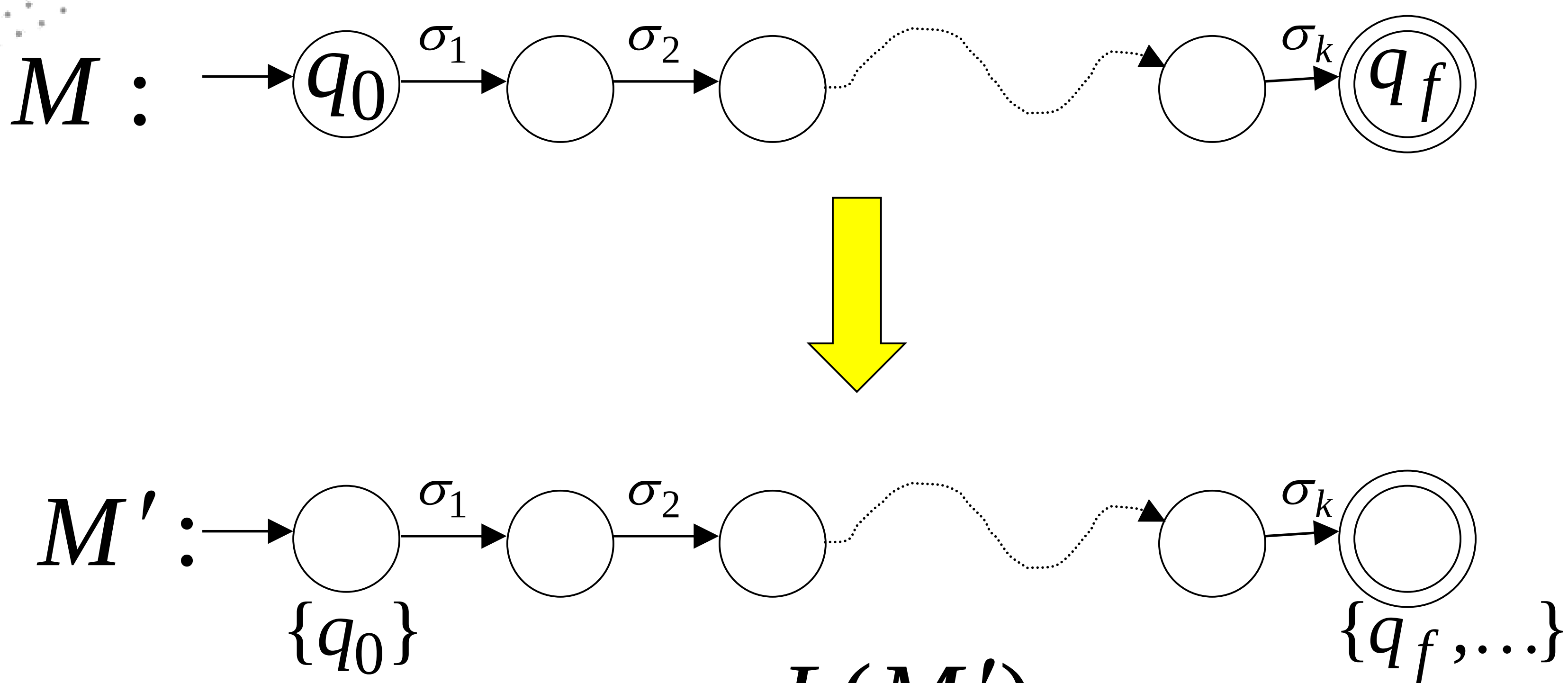
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$



Therefore if

$$w \in L(M)$$

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



$$w \in L(M')$$

We have shown:

$$L(M) \subseteq L(M')$$

We also need to show:

$$L(M) \supseteq L(M')$$

(proof is similar)

QUIZ TIME

Which of the following statements is true regarding NFAs and regular languages?

a) NFAs can accept all regular languages.

b) NFAs can only accept a subset of regular languages.

c) NFAs cannot accept any regular languages.

d) NFAs can accept some regular languages, but not all.

Answer: a) NFAs can accept all regular languages.

QUIZ TIME

Which of the following is a defining characteristic of regular languages that makes them recognizable by NFAs?

a) They can be described using context-free grammars.

b) They can be accepted by Turing machines.

c) They can be recognized by deterministic finite automata.

d) They can be recognized by non-deterministic finite automata.

Answer: d) They can be recognized by non-deterministic finite automata.

QUIZ TIME

Which operation is guaranteed to preserve the regularity of a language recognized by an NFA?

a) Union

b) Intersection

c) Complementation

d) Concatenation

Answer: a) Union

QUIZ TIME

If a language is recognized by both a DFA and an NFA, what can we conclude about the language?

a) The language is not regular.

b) The language is context-free.

c) The language is irregular.

d) The language is regular.

Answer: d) The language is regular.

QUIZ TIME

Which of the following algorithms can be used to convert an NFA into an equivalent DFA?

a) Thompson's construction algorithm

b) Subset construction algorithm

c) Hopcroft's algorithm

d) CYK parsing algorithm

Answer: b) Subset construction algorithm

Terminal questions

1.What does NFA - DFA equivalence mean?
2. How are NFAs and DFAs different in terms of structure and behavior?
3. Can NFAs and DFAs recognize the same set of languages?
4. What is the process of converting an NFA to an equivalent DFA called?
5. What is the subset construction algorithm used for?
6. How does the subset construction algorithm establish the equivalence between NFAs and DFAs?
7. Are there any languages that can be recognized by an NFA but not by an equivalent DFA?
8. Are there any languages that can be recognized by a DFA but not by an equivalent NFA?
9. Why is NFA - DFA equivalence important in automata theory?
10. How does understanding NFA - DFA equivalence help in designing efficient automata-based systems and algorithms?

THANK YOU

Team – TOC