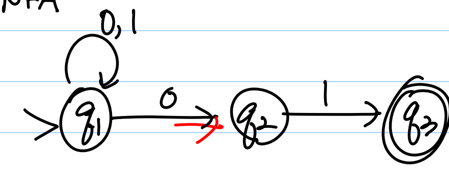
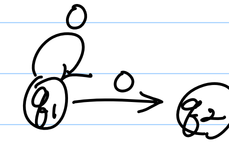
< Lecture 6 >

- NFA and DFA have the same expressive power.
- Any DFA is trivially a NFA. (Why?)
- Any NFA can be turn to an equivalent DFA

(We say that two machines are equivalent if they recognize the same languages

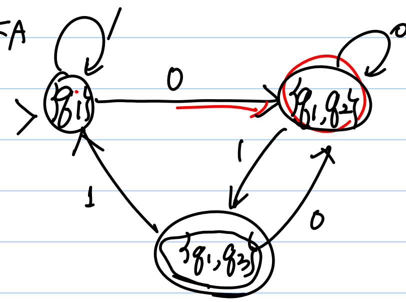
NFA

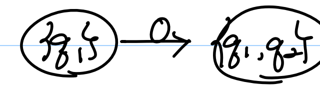




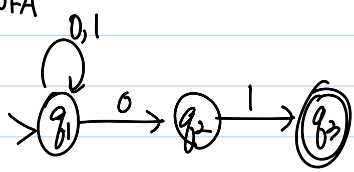
vs

DFA

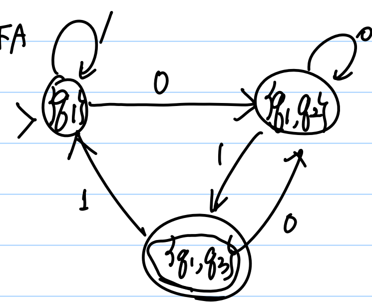




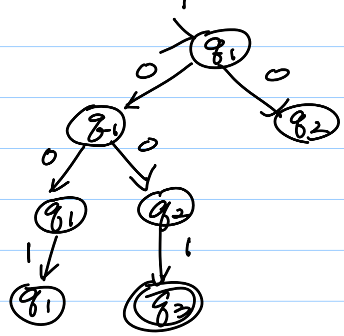
NFA

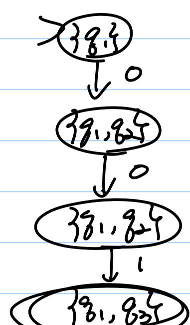


DFA

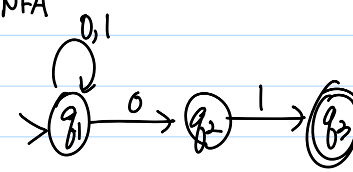


On input 001 :

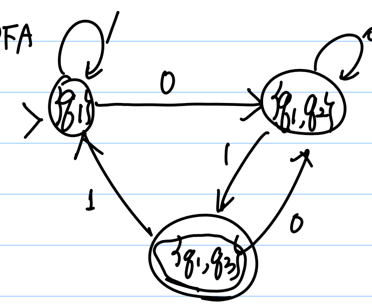




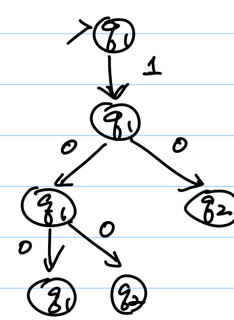
NFA

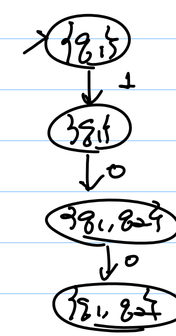


DFA



On input 100 :





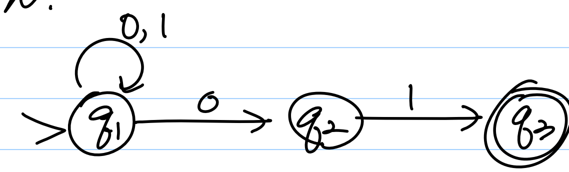
Thm 1.39 Every NFA is equivalent to a DFA.
That is, Given a NFA N that recognizes A, there is a DFA M that recognizes A

Proof idea: Given any NFA, we build a DFA that "simulates" the NFA.

- Each state of the DFA is a set of "possible" states in NFA.
- Each transition of the DFA simulates how the set of possible states will change according to the transition of the NFA.

[I] Simple case first: Assume no ε edges

Given NFA $N=(Q, \Sigma, \delta, q_0, F)$, we build DFA $M=(Q', \Sigma, \delta', q_0', F')$ that is equivalent to N.



1. $Q' = \mathcal{P}(Q)$

$Q' =$

2. $q_0' = \{q_0\}$

3. $F' = \{R \in Q' \mid R \text{ contains an accept state of } N\}$

4. $\delta'(R,a) = \bigcup_{r\in R} \delta(r,a)$ for $R \in Q'$

$\delta'(\emptyset, 0) = \emptyset$
$\delta'(\emptyset, 1) = \emptyset$

$\delta'(\{q_1\}, 0) = \{q_1, q_2\}$
$\delta'(\{q_1\}, 1) = \{q_1\}$

$\delta'(\{q_2\}, 0) = \emptyset$
$\delta'(\{q_2\}, 1) = \{q_3\}$

$\delta'(\{q_3\}, 0) = \emptyset$
$\delta'(\{q_3\}, 1) = \emptyset$

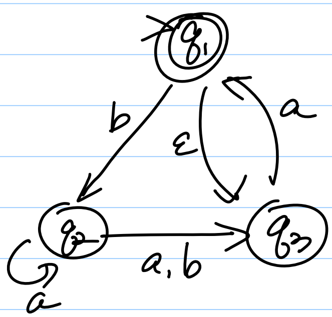
$\delta'(\{q_1, q_2\}, 0) = \{q_1, q_2\}$
$\delta'(\{q_1, q_2\}, 1) = \{q_1, q_3\}$

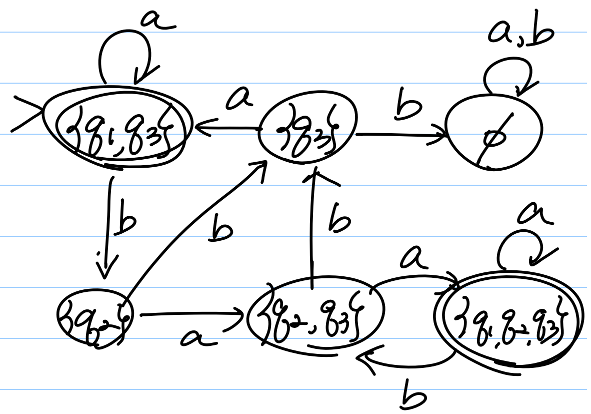
$\delta'(\{q_2, q_3\}, 0) = \emptyset$
$\delta'(\{q_2, q_3\}, 1) = \{q_3\}$

$\delta'(\{q_1, q_3\}, 0) = \{q_1, q_2\}$
$\delta'(\{q_1, q_3\}, 1) = \{q_1\}$

$\delta'(\{q_1, q_2, q_3\}, 0) = \{q_1, q_2\}$
$\delta'(\{q_1, q_2, q_3\}, 1) = \{q_1\}$

In the presence of ε-transitions





ε, a, baba, baa

b, bb, babba

[2] General case: N has ε transitions

Notation:

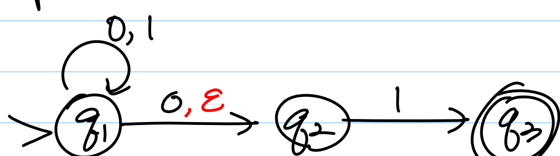
$E(R) = \{q \mid q$ can be reached from R without reading input$\}$

1. $Q' = \mathcal{P}(Q)$
2. $q_0' = \cancel{\{q_0\}}\ E(\{q_0\})$
3. $F' = \{R \in Q' \mid R$ contains an accept state of $N\}$
4. $\delta'(R,a) = \cancel{\bigcup_{r\in R} \delta(r,a)}$ for $R \in Q'$

   $\bigcup_{r\in R} E(\delta(r,a))$

Example

NFA



DFA : States?

start state?

accept states?

transition function?

Cor 1.40. A language is regular iff there is a NFA that recognizes it.

- Gives an alternative characterization of a regular language.
- We could have defined a regular language as

A language is <u>regular</u> if there is a ~~DFA~~ NFA that recognizes it.

Recap:

- A language is a set of strings
- First classification : regular vs non-regular languages
- DFA : 1. state diagram
  [NFA] 2. Formal definition

  "M accepts w" : 1. informal definition (in terms of state diagram)
  2. formal definition
- DFA and NFA are equivalent in terms of reco nizing languages
- Constructing a new DFA given some DFAs.
  - DFA for recognizing $A \cup B$, $A \cap B$, $\overline{A}$ (complement)
- Constructing an equivalent NFA given a DFA.