Properties of Regular Languages

DEFINITION 1.16

A language is called a ***regular language*** if some finite automaton recognizes it.

- DFA and NFA are finite automaton
- So, a language recognized by DFA or NFA is a regular language.

Closure Properties

- **THEOREM 1.45**

  The class of regular languages is closed under the union operation.

- Product DFA construction proof, we have seen.
- Now, we attempt using NFAs.

- Let $L(N_1) = A_1$, and $L(N_2) = A_2$

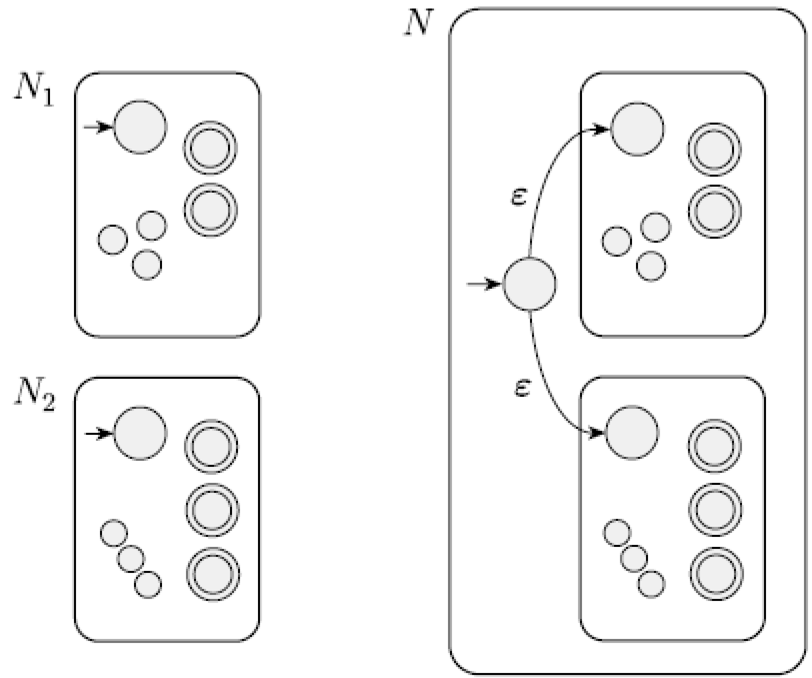


FIGURE 1.46
Construction of an NFA N to recognize $A_1 \cup A_2$

- *Mathematical description of this construction is left as an exercise. {can refer to Sipser book}*

- But, for intersection, still product machine is needed. You cannot do like this for intersection.

THEOREM 1.47

The class of regular languages is closed under the concatenation operation.

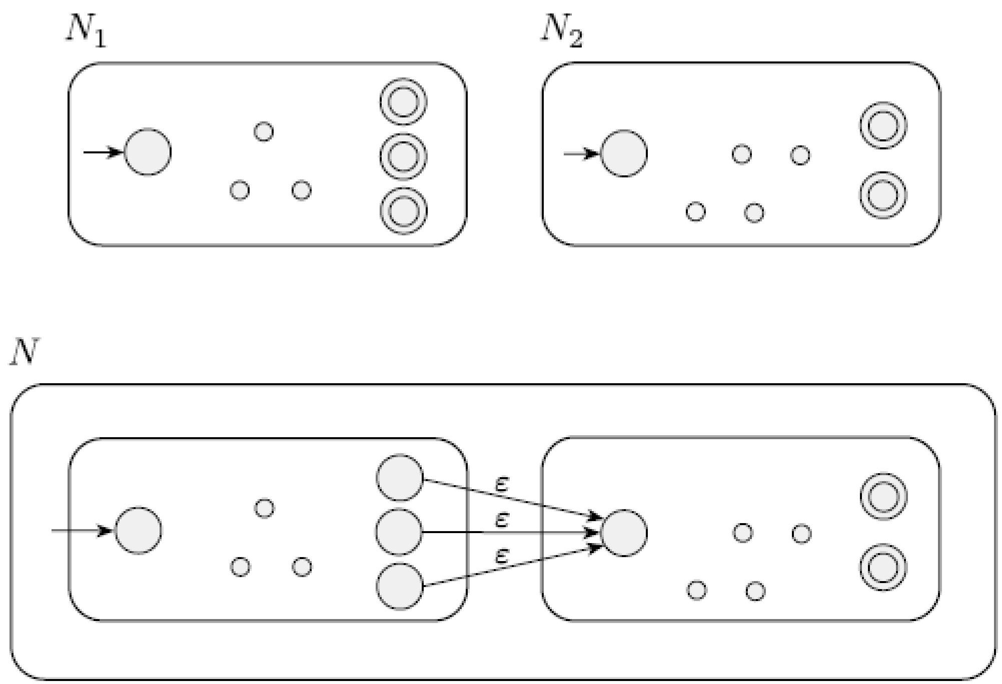
- 

FIGURE 1.48
Construction of N to recognize $A_1 \circ A_2$

Mathematically,

- PROOF

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1, and
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

Construct $N = (Q, \Sigma, \delta, q_1, F_2)$ to recognize $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   The states of N are all the states of N_1 and N_2.
2. The state q_1 is the same as the start state of N_1.
3. The accept states F_2 are the same as the accept states of N_2.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q,a) & q \in Q_2. \end{cases}$$

THEOREM 1.49

The class of regular languages is closed under the star operation.

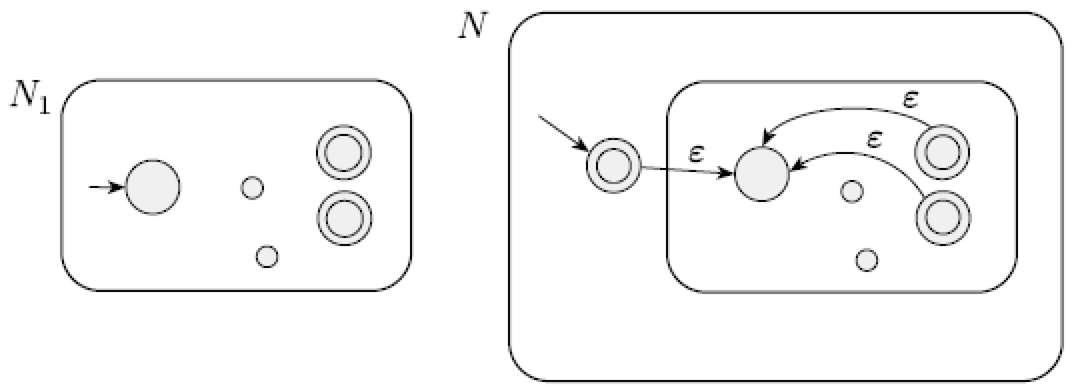
- 

FIGURE 1.50
Construction of N to recognize A^*

PROOF Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1.
Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize A_1^*.

1. $Q = \{q_0\} \cup Q_1$.
   The states of N are the states of N_1 plus a new start state.
2. The state q_0 is the new start state.
3. $F = \{q_0\} \cup F_1$.
   The accept states are the old accept states plus the new start state.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

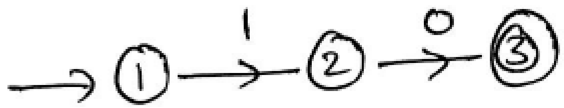
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon. \end{cases}$$

Exercise

- **design a DFA to accept A* where A= {10, 01}.**
  - **Construct NFA**
  - **Then convert this to DFA**

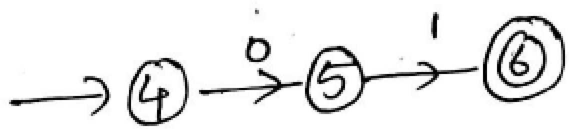
NFA for $\{10\}$

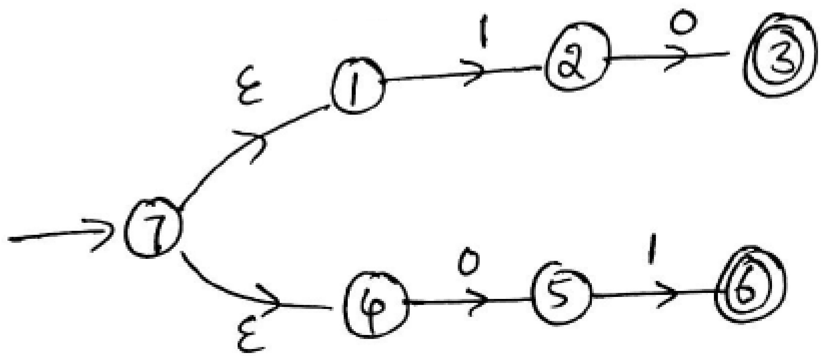
N_1



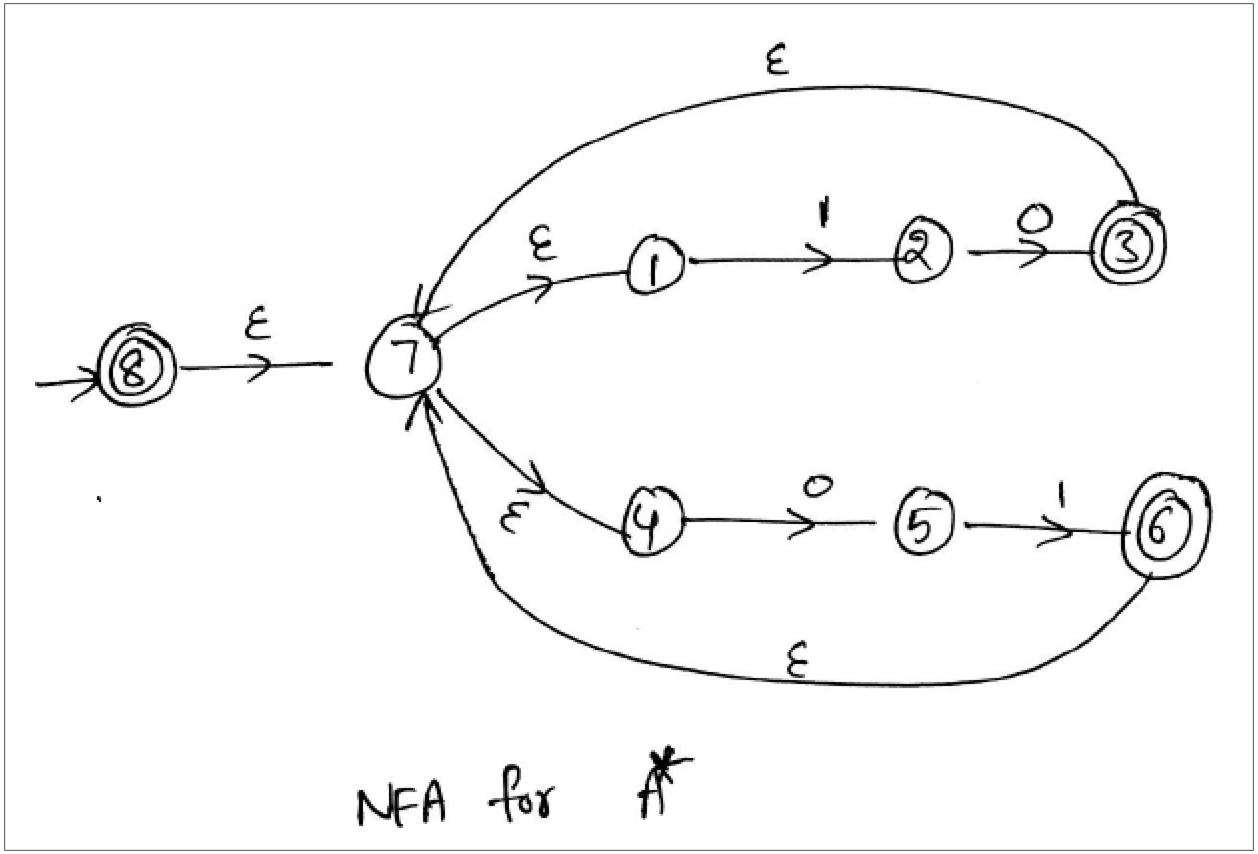
NFA for $\{01\}$

N_2





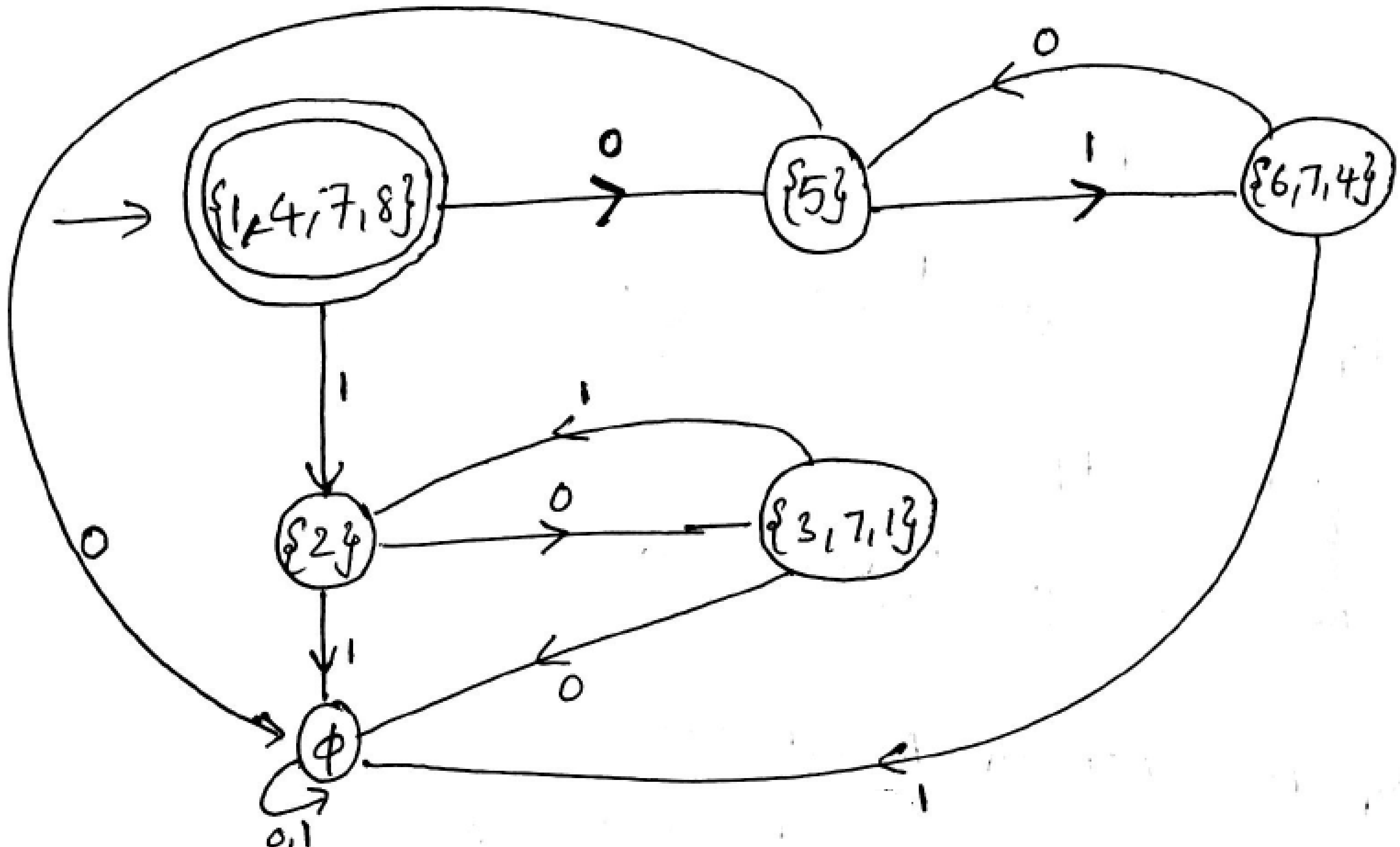
NFA N for $A = \{01, 10\}$



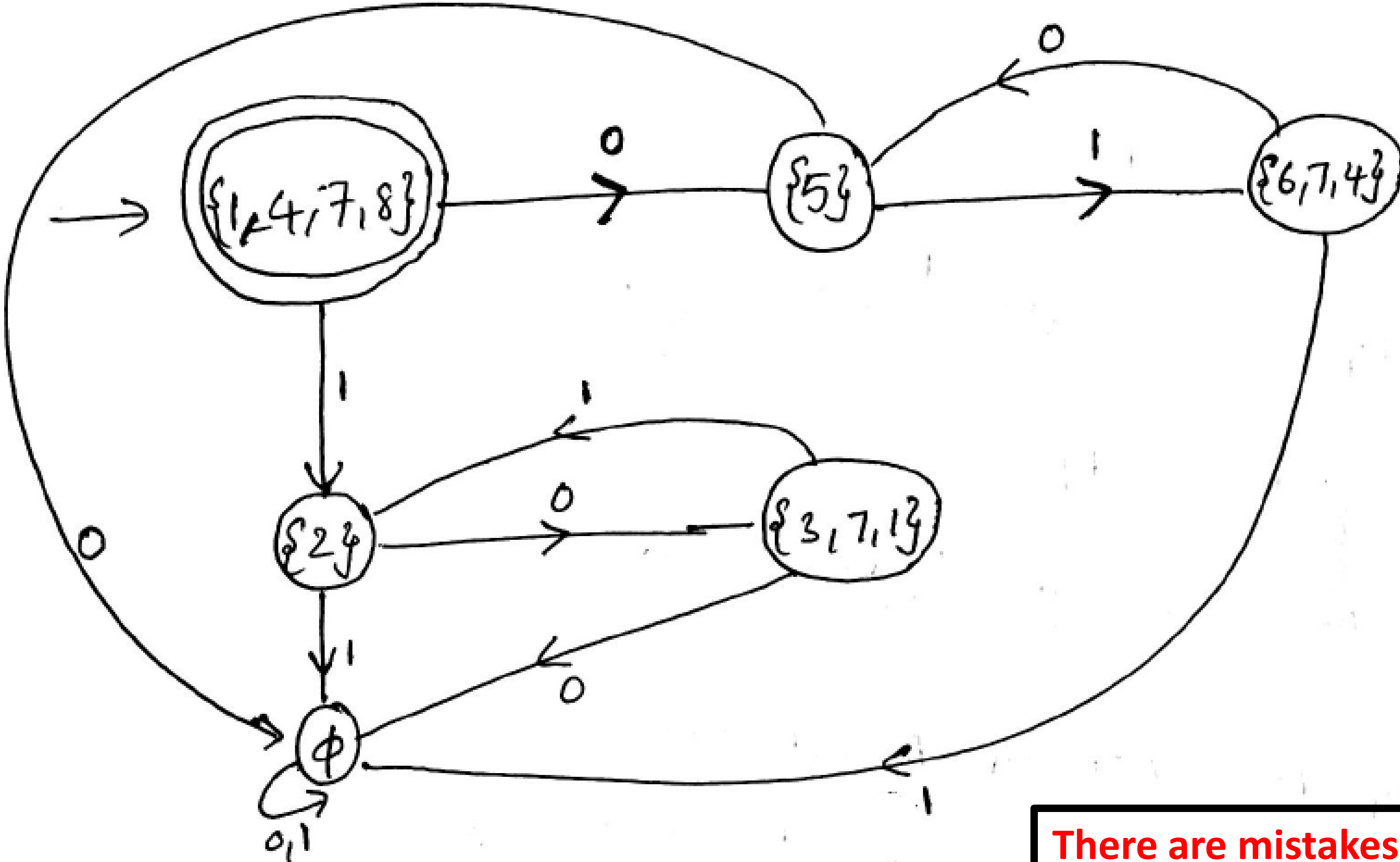
Now we should convert this to DFA.

Note that, E({8}) = {8, 7, 1, 4}.

Now, can you convert this NFA in to an equivalent DFA ?



DFA for A^*



There are mistakes in this slide, try to correct them !!!

DFA for A^*