Aim of today's lecture

1) To show equivalence between NFA & DFA

2) To show equivalence between DFA & regular expressions.

Equivalence between NFA & DFA

Theorem A language is acceptable by an NFA if and only if it acceptably by a DFA

Proof Idea

DFA $\Rightarrow$ NFA

This is trivial as by definition, every DFA is also an NFA.

NFA $\Rightarrow$ DFA

We will show this in two steps -

a. First we show that for every NFA with ϵ transitions (we will call this as ϵ-NFA), there exist an

NFA without ε - transitions, that accepts the same language.

b. Then we show that for every NFA without ε - transitions, there is a a DFA that accepts the same language.

Step - a ε-NFA to NFA

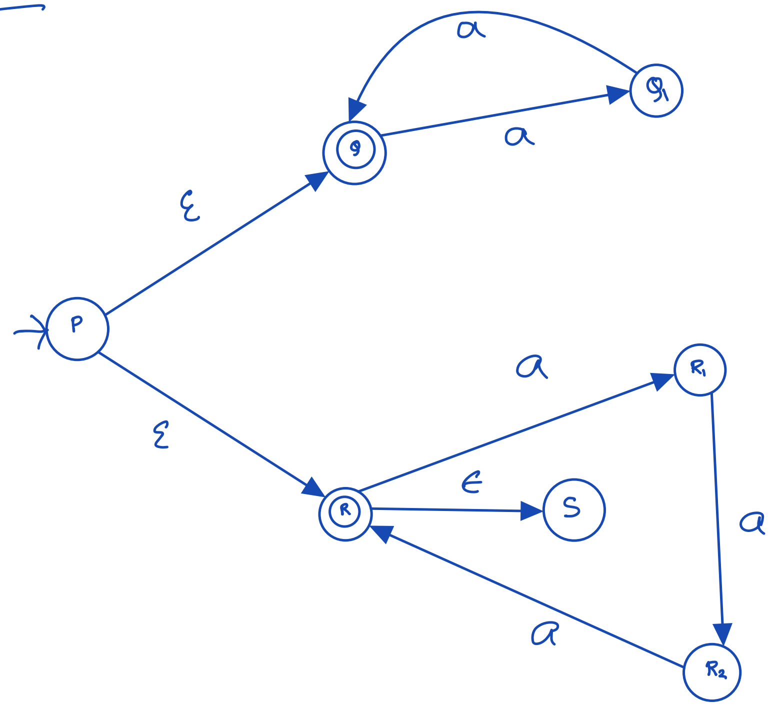
general procedure

1. First we define the term ε- closure as follows.

The ε-closure of a state p in the ε-NFA, denoted by E(p), ~~as~~ and defined as

$$E(p) = \{ q \mid q \text{ can be reached from } p \text{ through zero or multiple } \varepsilon\text{-transitions} \}$$

Example



$E(P) = \{P, Q, R, S\}$

$E(R) = \{R, S\}$

$E(Q) = \{Q\}$

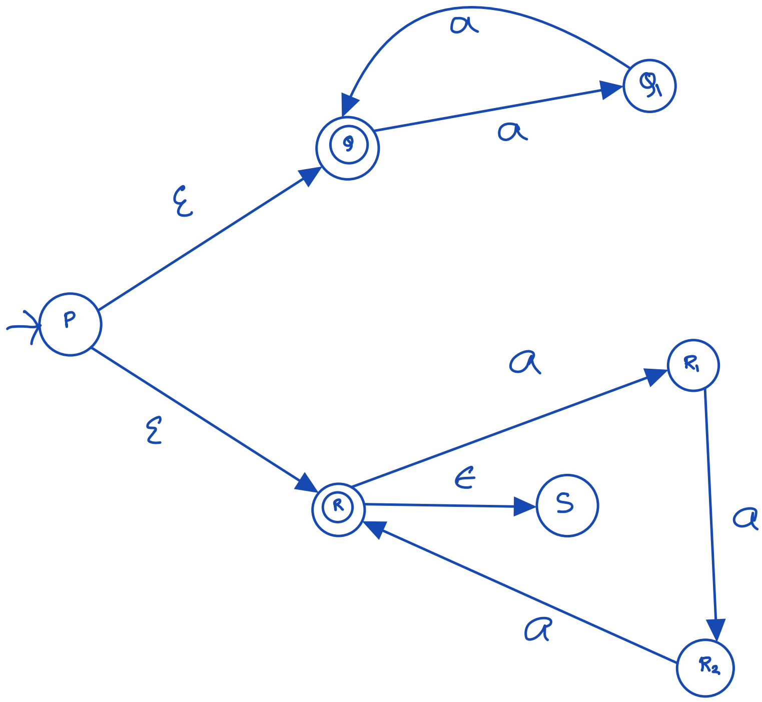
$E(Q_1) = \{Q_1\}$

$E(R_1) = \{R_1\}$

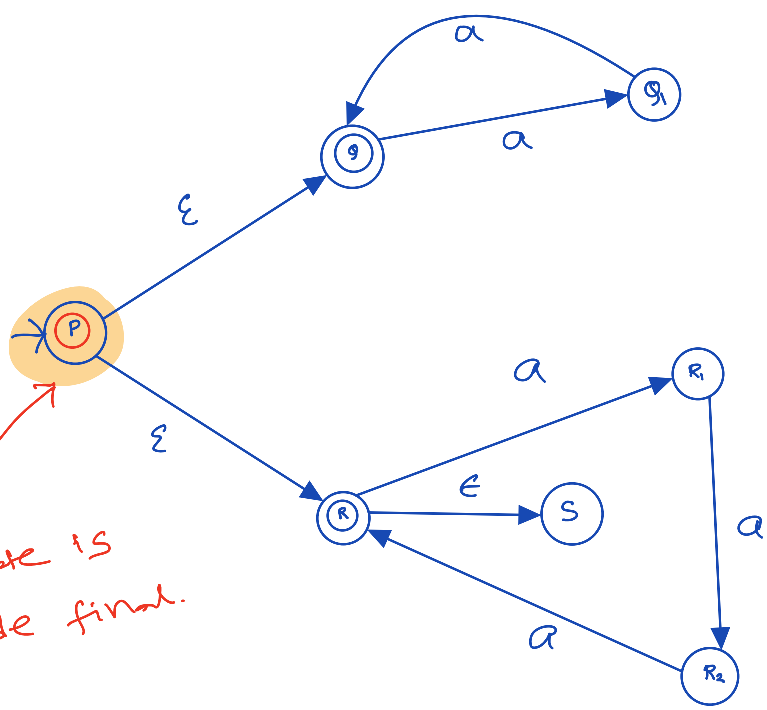
$E(R_2) = \{R_2\}$

$E(S) = \{S\}$.

2. Make a state p as final state iff $E(p)$ contains an accepting state in the ε-NFA



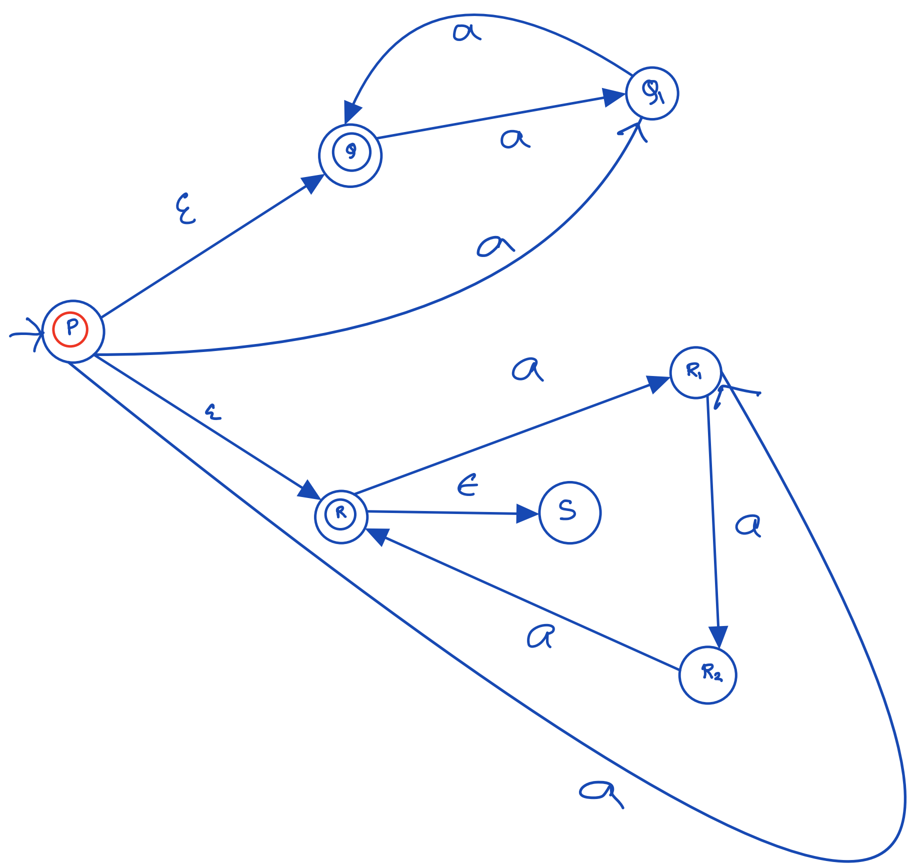
After executing step 2



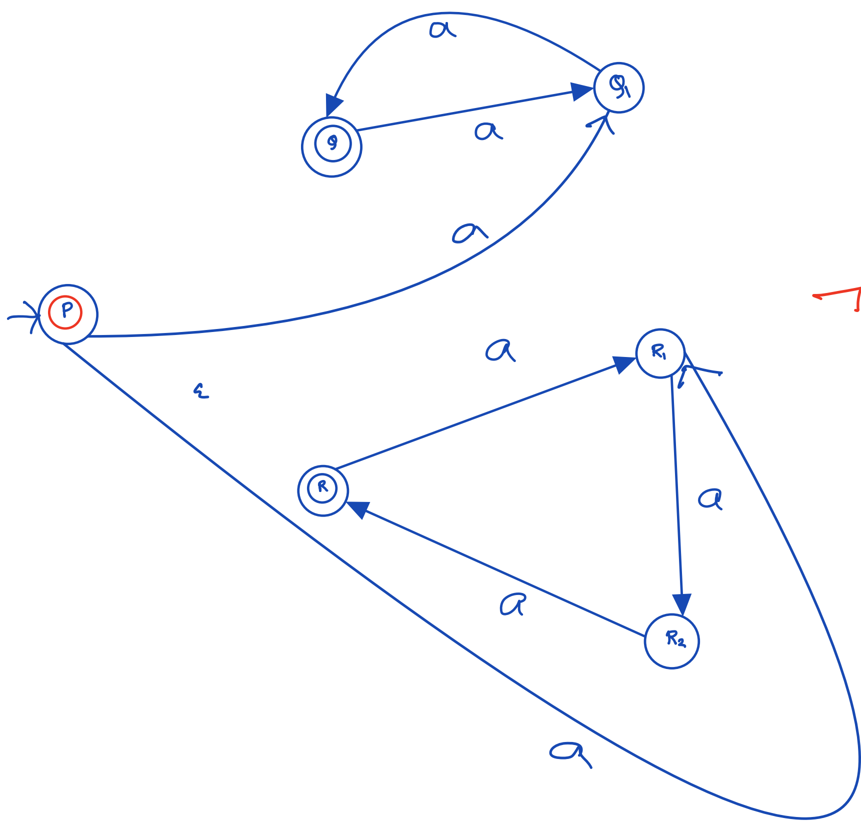
This state is made final.

3. Add an edge from a state x to y on symbol a iff there is a transition from some state of $E(x)$ to y on symbol a.

after step 3



4. Delete all the ϵ transitions



The state S becomes useless.

Step-6 . NFA → DFA

Strategy

Write down all possible subsets of the set of states.

Each subset will be a possible state of the new machine.

A transition from one subset S to another T is added on character c, iff, trans s c = t, and t∈T and s∈S

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA

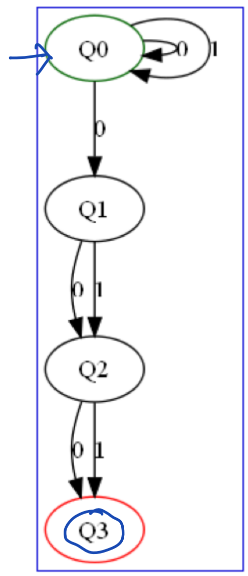
We construct a DFA $M = (P(Q), \Sigma, \hat{\delta}, \langle q_0 \rangle, \hat{F})$

Where

$$\hat{\delta}(S, a) = \bigcup_{q \in S} \delta(q, a) \quad \forall a \in \Sigma$$

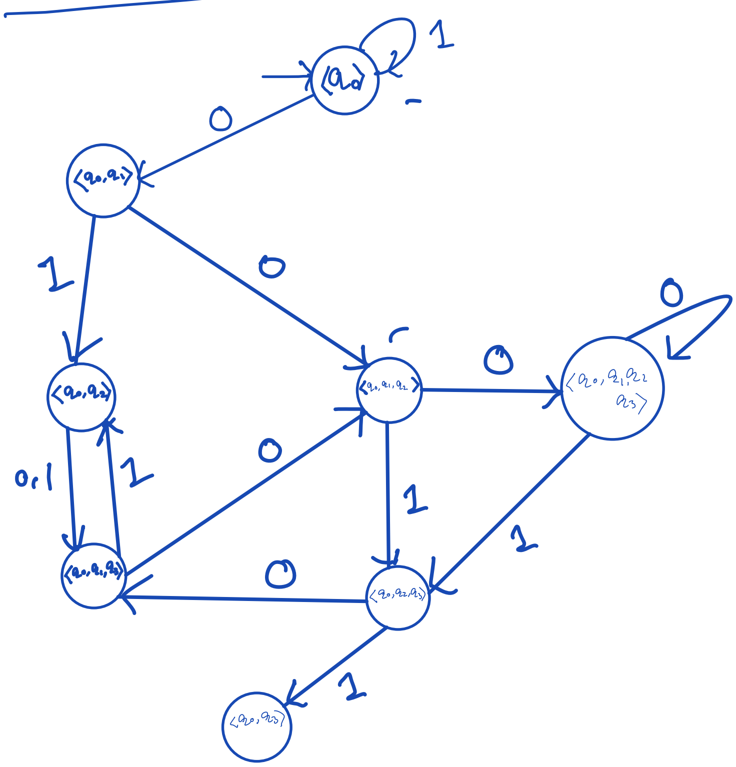
$$\hat{F} = \{ S \in Q \mid S \cap F \neq \phi \}$$

Example



Note

Equivalent DFA



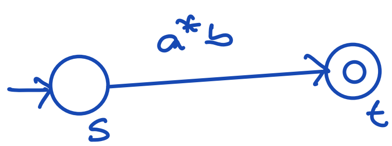
Equivalence between DFA & regular expression

Here, we discuss ~~how~~ from a DFA for a language L, how to construct a regular expression for L.

General strategy

We introduce the term generalized NFA, which is a machine where the transitions are defined on a regular expression instead of a symbol.

Example



Here, for every string w in a^*b,

$\delta(S, w) = t$.

Steps for converting DFA to regular expression

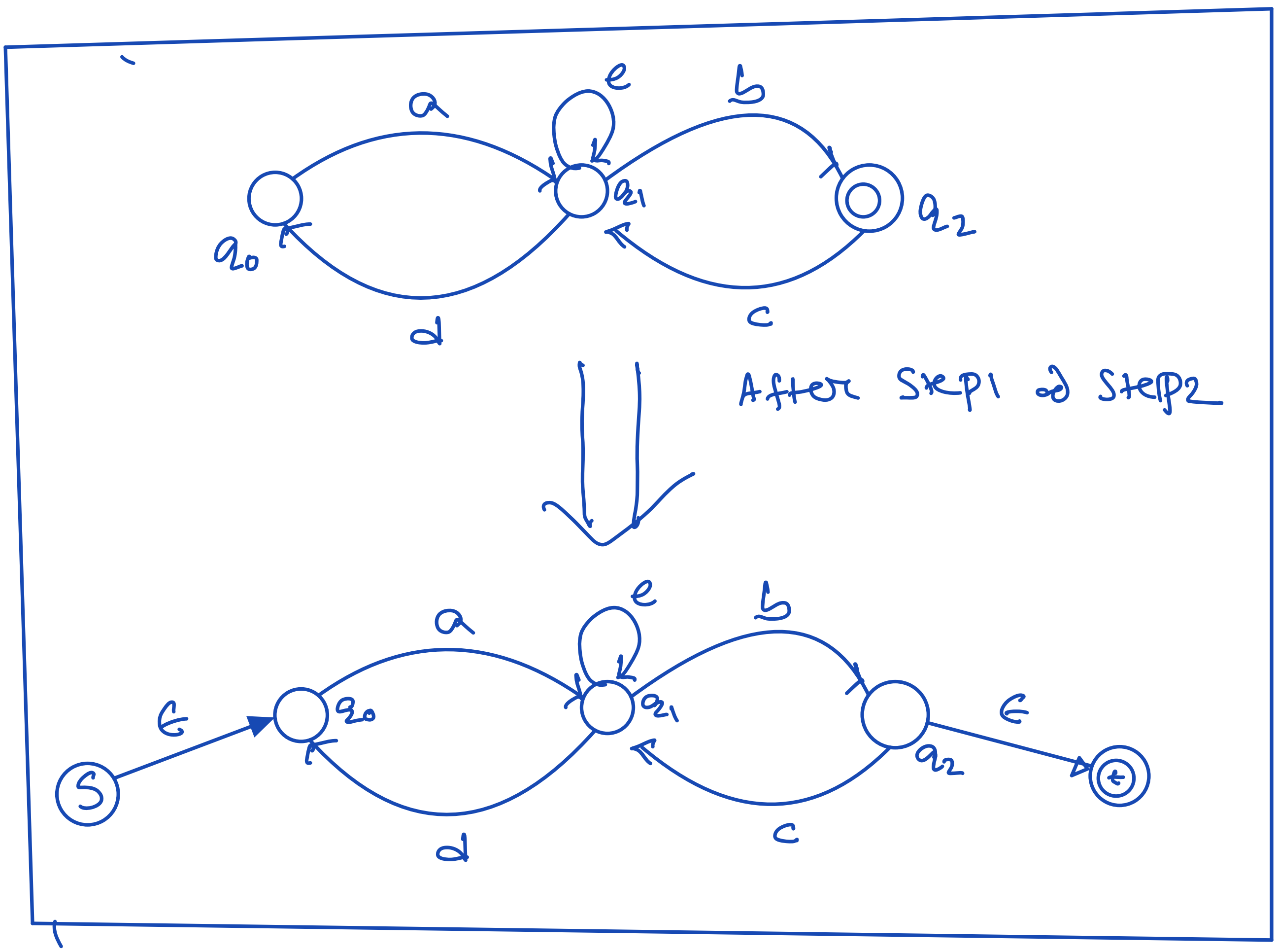
Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA

1. Add a new start state s & a new final state t.

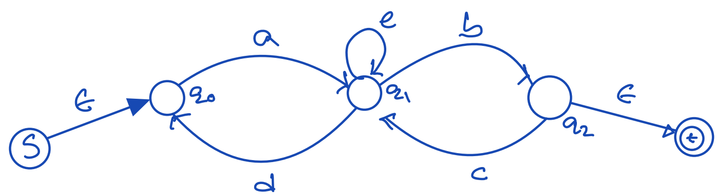
2. Add transitions

$$\delta(s, \epsilon) = q_0$$

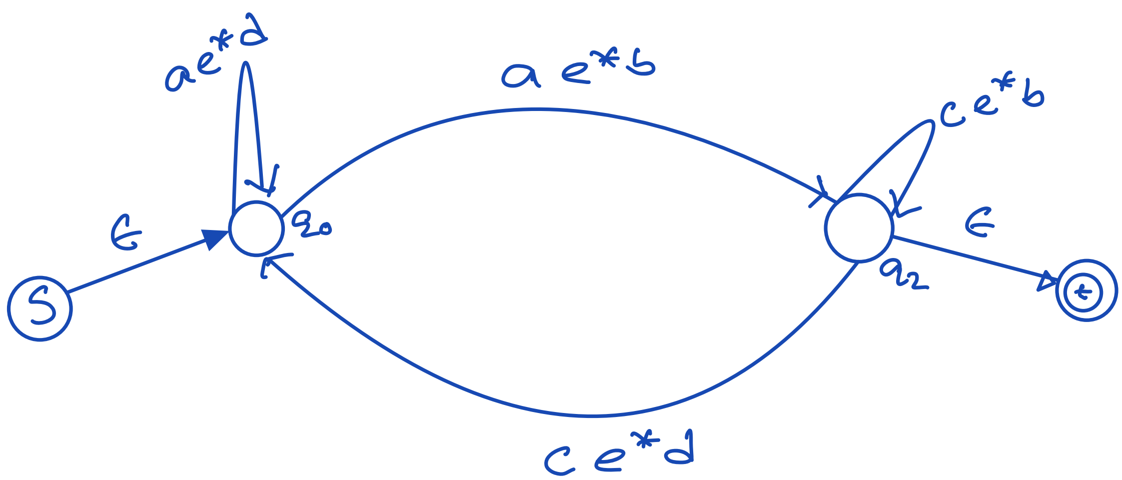
$$\delta(f, \epsilon) = t, \quad \forall f \in F.$$



3. Remove the states one by one & convert the DFA into a generalized NFA.

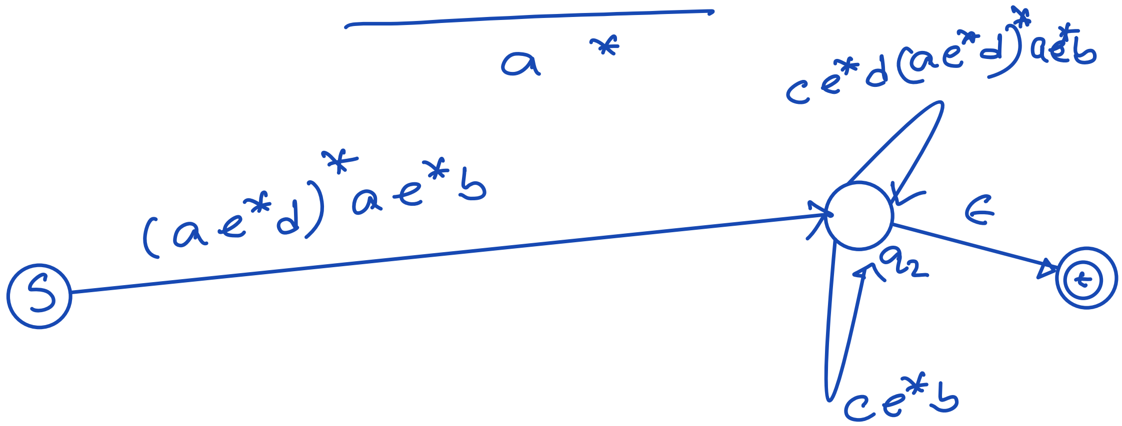


<u>Delete q_1</u>



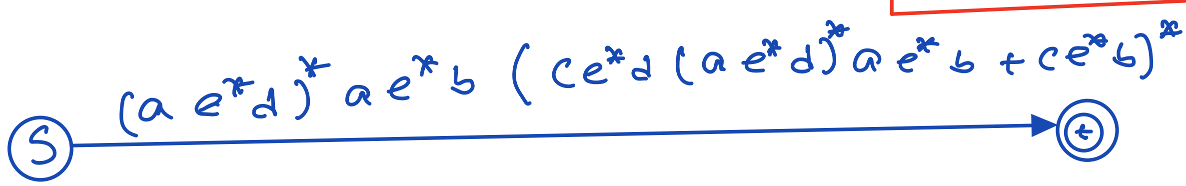
Delete q_0

$a \quad *$



Delete q_2

The regular expression



Another example is discussed in the class.