AUTOMATA FORMAL LANGUAGES AND LOGIC

Lecture Notes on
Non-deterministic Finite Acceptors
(NFA/λ-NFA)

Prepared by :
Prof. Sangeeta V I
Assistant Professor

Department of Computer Science & Engineering

PES UNIVERSITY

(Established under Karnataka Act No.16 of 2013)
100-ft Ring Road, BSK III Stage, Bangalore – 560 085

Table of Contents

Examples Solved

1.Non-deterministic Finite Accepters/Automata(NFA)

An NFA can be represented by a 5-tuple (Q, ∑, δ, q_0, F) where –

- Q is a finite set of states.
- ∑ is a finite set of symbols called the alphabets.
- δ is the transition function where δ: Q × ∑ → 2^Q
  (Here the power set of Q (2^Q) has been taken because in case of NFA, from a state, transition can occur to any combination of Q states)
- q_0 is the initial state from where any input is processed ($q_0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

In an NFA, a (state, symbol) combination may lead to several states simultaneously.

If a transition is labeled with the empty string as its input symbol, the NFA may change states without consuming input.

An NFA may have undefined transitions.

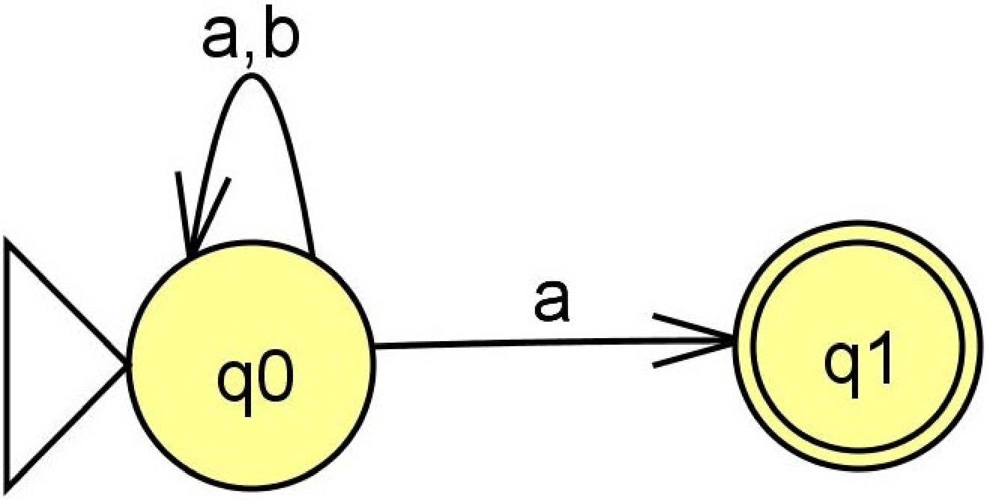
In DFA we have to be determined about the transition of every input symbol on every state.

IN NFA we do not have this restriction.

Example 1:

Assume Σ={a,b}
Consider language L1={set of string that end in ‘a’}
Minimum string : a

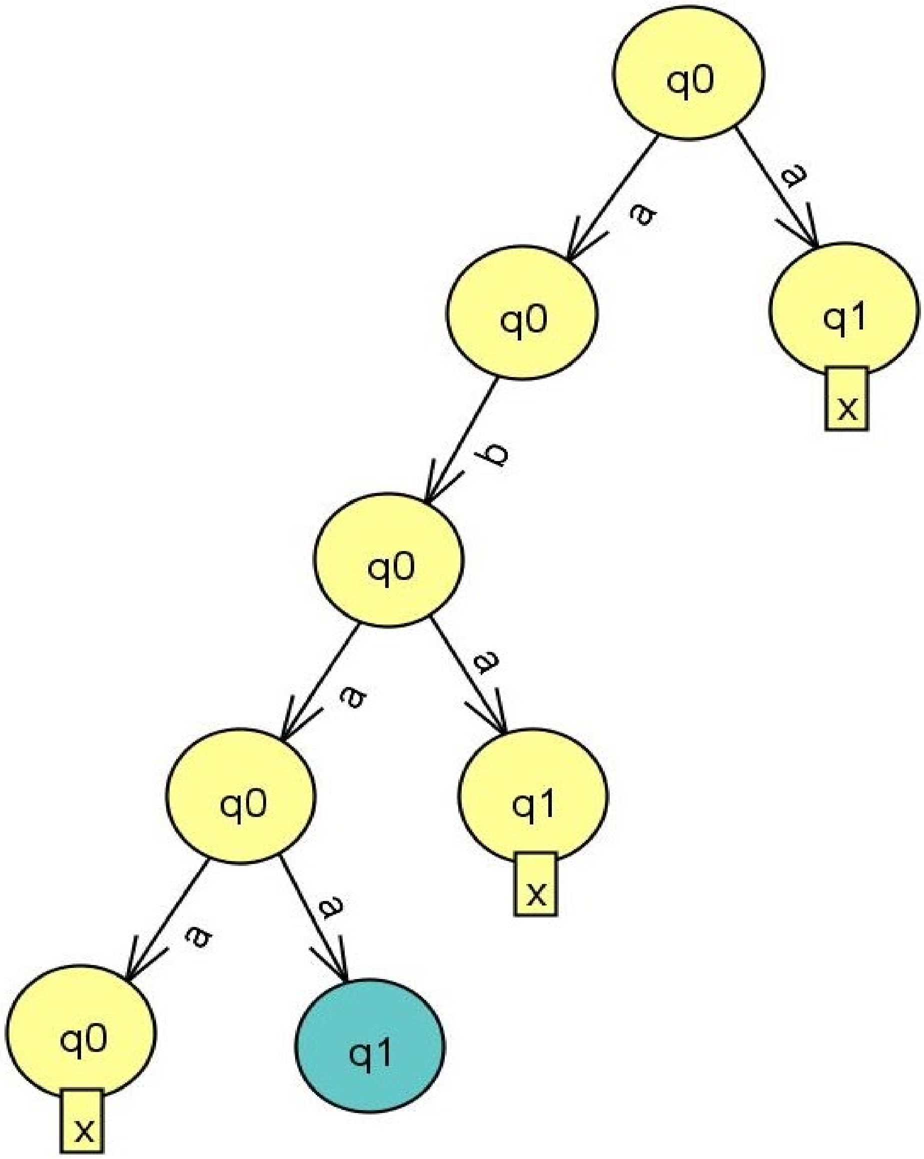
L1={a,(a’s or b’s) a}
We will first construct the NFA for the set of strings ending in a.



- In NFA we can have multiple transitions on the same symbol from a given state or there can be no transition on a symbol from a given state.
- We have two transitions on ‘a’ in q0,one remains in the same state and the other moves to state q1.
- In q1, there is no transition on a and b.This is allowed in NFA but not in DFA.
- It is easier to construct a NFA ,since we don't have to worry about the transition of every symbol in every state.
- We can convert NFA to DFA using an algorithm .

- Computation in NFA for the string abaa.
  We will try out simultaneously all the possible paths the string can take ,even if one of the paths leads us to the final state then we will accept the string.

  abaa



Tracing the string "abaa"

- On q0,if we see **'a'** there are two possibilities,one it remains in q0 ,other it goes to state q1.We will check both the possibilities and see which path takes us to the final state.

- On q0, we can go to state q1 after seeing ‘a’ but in q1 we do not have any transition ,hence we do not choose this path(we haven't reached the end of the string) .

- We will pick the other path where on q0,after seeing ‘a’ we remain in q0.

- Now,we are in q0,the next symbol from the string “abaa” is **‘b’.**

- In q0 ,on ‘b’ we remain in q0 .

- Now,we are still in q0,the next symbol from the string “abaa” is **‘a’.**

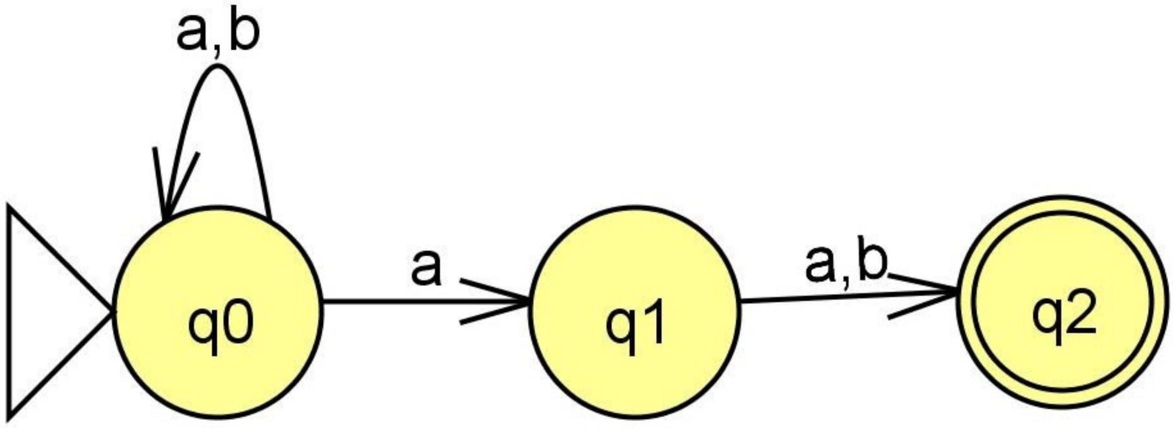
- On q0, we can go to state q1 after seeing ‘a’ but in q1 we do not have any transition ,hence we do not choose this path.

- We will pick the other path where on q0,after seeing ‘a’ we remain in q0.

Example 2:

Assume Σ={a,b}

Consider language L2={Strings where the second symbol from RHS is 'a'}
(a's or b's) a (a or b)



λ-Non-deterministic Finite Acceptors/Automata(λ-NFA)

A nondeterministic finite automata with λ–transitions is a 5-tuple (Q, Σ, q0, A, δ), where Q and Σ are finite sets, q0 ∈ QQ, A ⊆ Q, and δ : Q × (Σ ∪ {λ}) → 2 Q .
When we have a λ–transition we jump to states without consuming any input.

Example 1:

Assume Σ={a,b}
Consider language L1={Strings that start and end with the same symbol}
L1={awa or bwb |w∈{a,b}*}
We construct two NFA one each for awa and bwb and combine the NFA with a single new start state .