NFA (Non-Deterministic finite automata)

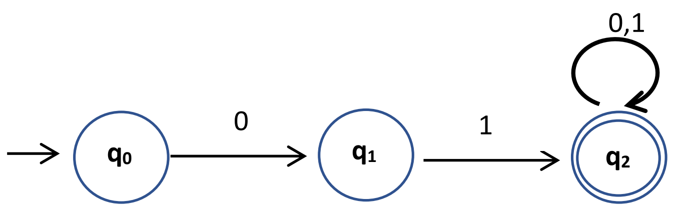
- NFA stands for non-deterministic finite automata. It is easy to construct an NFA than DFA for a given regular language.
- The finite automata are called NFA when there exist many paths for specific input from the current state to the next state.
- Every NFA is not DFA, but each NFA can be translated into DFA.
- NFA is defined in the same way as DFA but with the following two exceptions, it contains multiple next states, and it contains ε transition.

NFA also has five tuples same as DFA, but with different transition functions, as shown as follows:

1. Q: finite set of states
2. ∑: finite set of the input symbol
3. q0: initial state
4. F: final state
5. δ: Transition function

note: δ in NFA $\delta: Q \times \sum \rightarrow 2^Q$, but in DFA $\delta: Q \times \sum \rightarrow Q$

Ex: NFA with ∑ = {0, 1} and accepts all strings with 01.



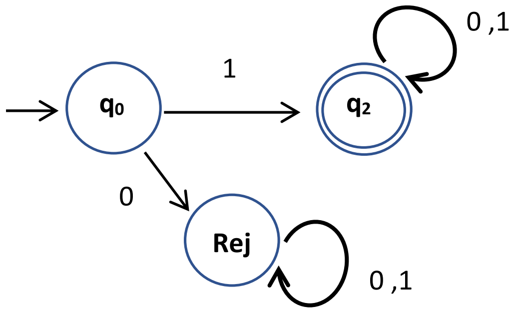
δ-table:

state	0	1
→q0	q1	ε
q1	ε	q2
*q2	q2	q2

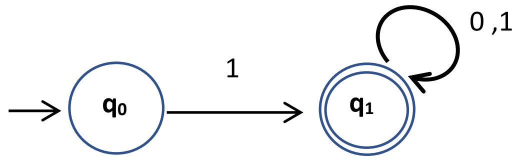
Ex:

L= {1x|x ϵ {0,1}*} in DFA, NFA machines.

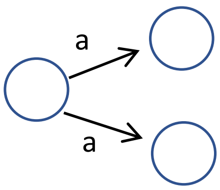
DFA:



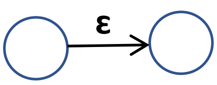
NFA:



- A nondeterministic finite automata (NFA) allows transitions on a symbol from one state to possibly more than one other state.

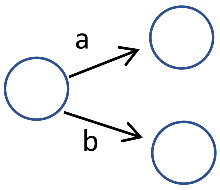


- Allow ε-transitions from one state to another whereby we can move from the first state to the second without inputting the next character.



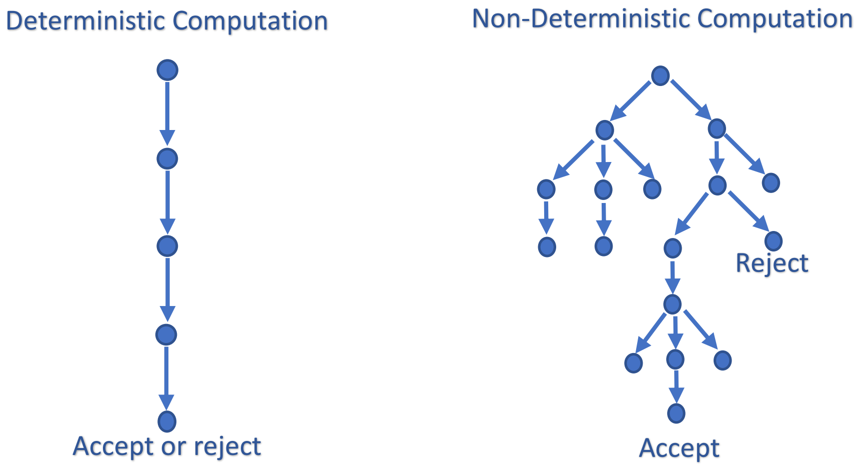
- In an NFA a state may have zero, one, or more exiting arrows for each symbol of alphabet.

$\sum=\{a,b,c\}$

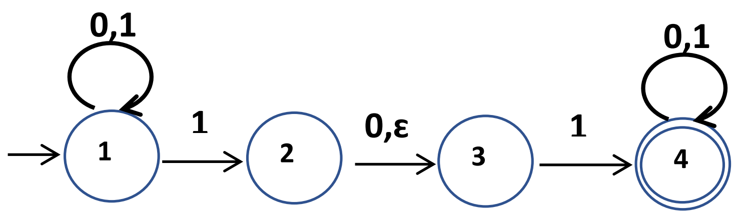


Basic NFA ideas

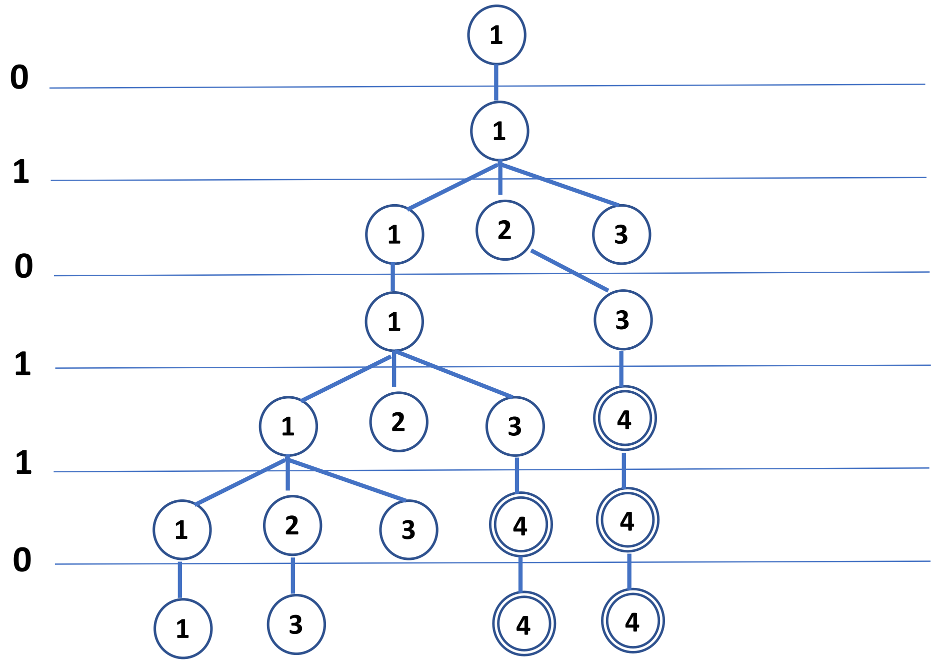
- ✓ Start in the start state.
- ✓ If any ε transition, clone a machine for each.
- ✓ Read a symbol, and clone a machine for each matching transition.
- ✓ If a symbol is read and there is no way to exit from a state, that machine dies.
- ✓ At the end of input if any machine accepts then accept.



Ex: Build NFA machine that accepts strings containing either 101 or 11 as a substring for ∑= 0, 1}, and read 010110.



Read: 010110



Equivalence of machines

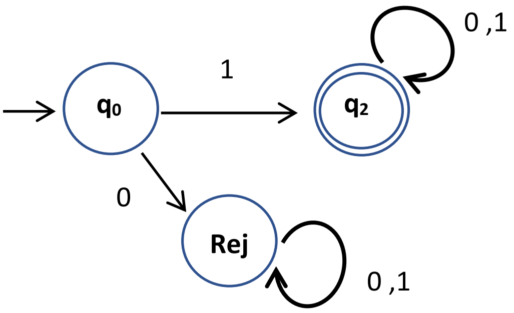
Definition:

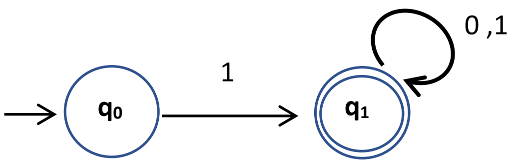
Machine M_1 is equivalent to machine M_2 if $L(M_1) = L(M_2)$

Example of equivalent machines:

L= {1x|x ϵ {0,1}*}

DFA: NFA:



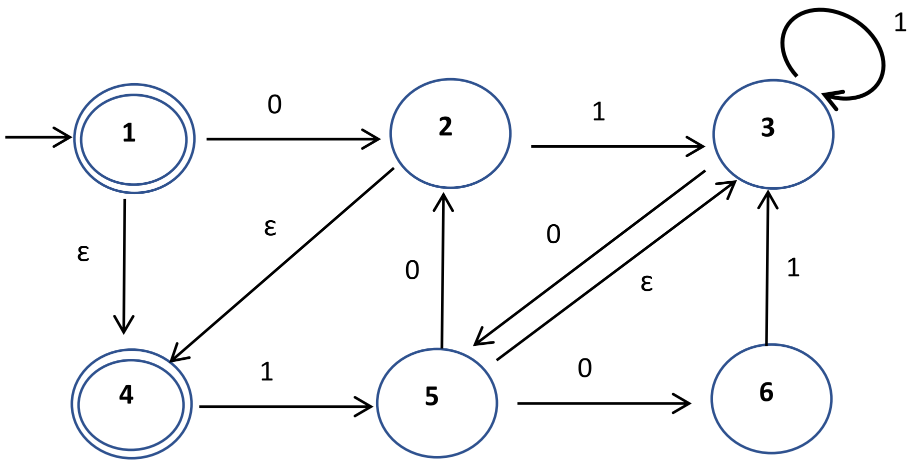


Any language L accepted by a DFA is also accepted by an NFA

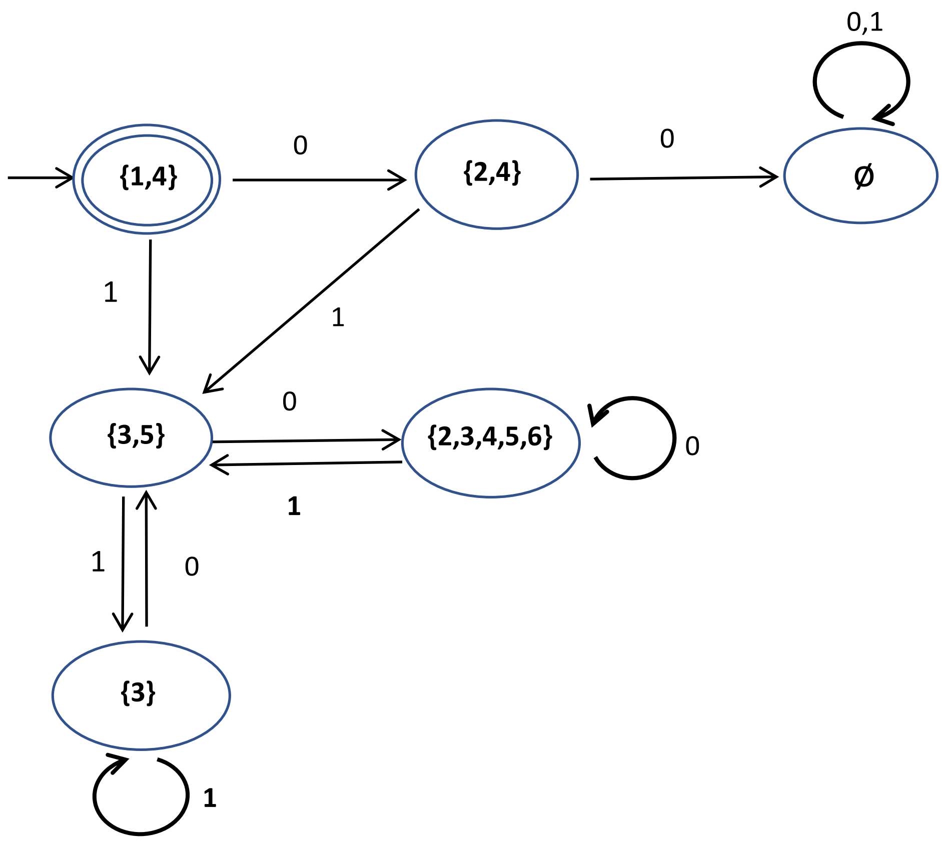
Proof by construction:

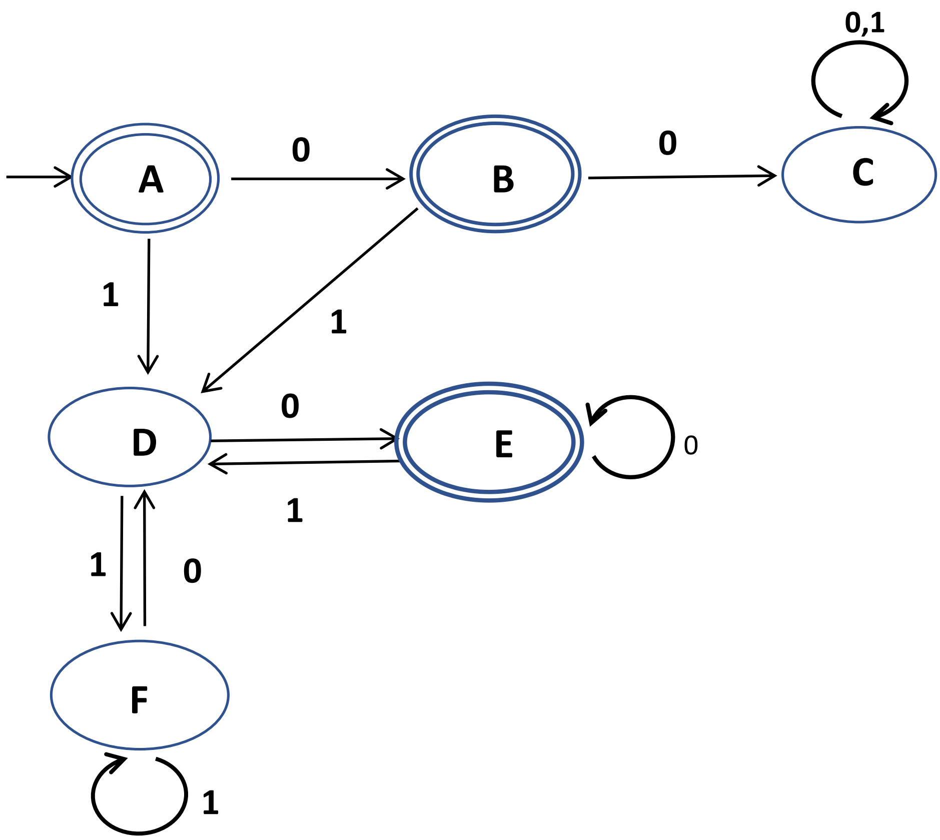
given an arbitrary NFA **N**, construct an equivalent DFA **D**

suppose that N of the $\sum = \{0, 1\}$ is as follows:



Solution:




0,1
A
0
B
0
C
1
1
D
0
E
0
1
1
0
F
1