https://www.tutorialspoint.com/what-is-the-difference-between-dfa-and-nfa

DFA

A Deterministic Finite automata is a five-tuple automata. Following is the definition of DFA −

M =(Q, Σ, δ,q0,F)

δ : Q X Σ → Q

Where,

- Q : Finite set called states.
- Σ : Finite set called alphabets.
- δ : Q × Σ → Q is the transition function.
- q0 ϵ Q is the start or initial state.
- F : Final or accept state.

NFA

NFA also has five states same as DFA, but with different transition function, as shown follows −

δ: Q X Σ→ 2^Q

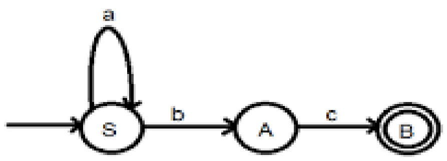
Where,

- Q : Finite set of states
- Σ : Finite set of the input symbol
- q0 : Initial state
- F : Final state
- δ : Transition function

S.No.	DFA	NFA
1	The full form of DFA is Deterministic Finite Automata.	The full form of NFA is Nondeterministic Finite Automata (NFA).
2	It is not competent in handling an Empty String transition.	It is competent to handle an empty string transition.

3	In DFA, only a sole state transition can be accomplished for each symbolic representation of the characters.	In NFA, no particulars are required from the users.
4	It can be defined as one machine.	Multiple machines execute computational tasks at the same time.
5	In DFA, backtracking is allowed.	In NFA, backtracking is not allowed.
6	In DFA, empty string transitions are not noticed.	It allows empty string transition.
7	It is tough to construct a DFA.	It is easy to construct a NFA.
8	It needs more space.	It needs less space.
9	The complete time needed for managing any input string in DFA is shorter than NFA.	The complete time needed for managing any input string in NFA is larger than DFA.
10	All DFA are considered as NFA.	All NFA are not considered as DFA.

DFA:



NFA: