Difference between DFA and NFA

In automata theory four types of finite automata are used to recognize the regular language among these two are DFA and NFA. Both have the same function but there are some differences in their structure and working. It is very essential for one to have a clear understanding of the difference between DFA and NFA for anyone to understand the computational models or to know how to design an algorithm. This article will try to discuss the differences as well as give examples that can help in the understanding of the topics.

What is DFA?

DFA refers to Deterministic Finite Automaton. A Finite Automata(FA) is said to be deterministic if corresponding to an input symbol, there is a single resultant state i.e. there is only one transition.

A deterministic finite automaton is set of five tuples represented as,

Where,

- Q: A non-empty finite set of states in the finite control(qo, q1, q2, ...).
- Σ: A non-empty finite set of input symbols.
- δ: It is a transition function that takes two arguments, a state, and an input symbol, it returns a single state.
- qo: It is starting state, one of the states in Q.
- F: It is a non-empty set of final states/ accepting states from the set belonging to Q.

What is NFA?

NFA refers to Nondeterministic Finite Automaton. A Finite Automata(FA) is said to be non-deterministic if there is more than one possible transition from one state on the same input symbol.

A non-deterministic finite automata is also a set of five tuples and represented as,

Where,

- Q: A set of non empty finite states.
- Σ: A set of non empty finite input symbols.
- δ: It is a transition function that takes a state from Q and an input symbol from and returns a subset of Q.
- qo: Initial state of NFA and member of Q.
- F: A non-empty set of final states and member of Q.

Difference Between DFA and NFA

DFA	NFA
DFA stands for Deterministic Finite Automata.	NFA stands for Nondeterministic Finite Automata.
For each symbolic representation of the alphabet, there is only one state transition in DFA.	No need to specify how does the NFA react according to some symbol.
DFA cannot use Empty String transition.	NFA can use Empty String transition.
DFA can be understood as one machine.	NFA can be understood as multiple little machines computing at the same time.
In DFA, the next possible state is distinctly set.	In NFA, each pair of state and input symbol can have many possible next states.
DFA is more difficult to construct.	NFA is easier to construct.
DFA rejects the string in case it terminates in a state that is different from the accepting state.	NFA rejects the string in the event of all branches dying or refusing the string.
Time needed for executing an input string is less.	Time needed for executing an input string is more.
All DFA are NFA.	Not all NFA are DFA.
DFA requires more space.	NFA requires less space then DFA.

Dead configuration is not allowed.

eg: if we give input as 0 on q0 state so we must give 1 as input to q0 as self loop. Dead configuration is allowed.

eg: if we give input as 0 on q0 state so we can give next input 1 on q1 which will go to next state.

δ: QxΣ -> Q i.e. next possible state belongs to Q.   δ: Qx(Σ U ε) -> 2^Q i.e. next possible state belongs to power set of Q.

Conversion of Regular expression to DFA is difficult. Conversion of Regular expression to NFA is simpler compared to DFA.

Epsilon move is not allowed in DFA   Epsilon move is allowed in NFA

In a DFA, there is only one possible transition for each input symbol from any given state.In an NFA, there can be multiple transitions for a single input symbol from a given state.

Conclusion

In conclusion, DFAs are deterministic, uncomplicated to implement but they create greater automata while NFAs are flexible and can be less bulky but difficult to implement. In fact, despite these differences, NFAs and DFAs are as powerful as each other, in that they are able to recognize the same classes of languages. It is possible to convert any NFA to a DFA even though the amount of states in a resultant DFA can be more than in the initial NFA.