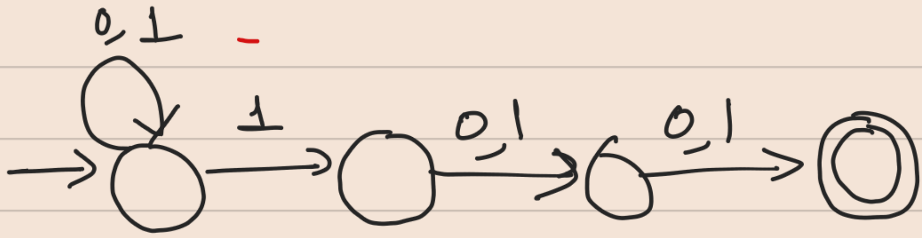
Non-deterministic Finite Automata:
(NDFA or NFA)

<u>Ex</u>: Design a FA for

Third Last 1 = $\{ w \in \{0,1\}^* \mid$ The third from last character is a 1 $\}$



Non-deterministic Finite Automata:
(NFA)

Mathematically NFA is a 5-tuple
$(Q, \Sigma, \delta, q_0, F)$.

Q: finite set of states

Σ: finite set of symbols, input alphabet

δ:

$$Q \times (\Sigma \cup \{\epsilon\}) \longrightarrow P(Q) \rightarrow \text{Power set of } Q$$

OR

$$Q \times (\Sigma \cup \{\epsilon\}) \longrightarrow 2^Q$$

<u>Note</u>: $\Sigma \cup \{\epsilon\}$ means Σ augmented by ϵ. This indicates that machine can move forward without an input symbol.

$q_0 \in Q$: initial state

$F \subseteq Q$: set of accept states

Note about δ:

1. Range of δ is a set of states, so that, given a state and a symbol, the machine can go to several states. We assume that it will always take the correct move

   Note

2. Move can be made without using the input. These are ∈-transitions and they can be made before or after an input symbol is used.

   Note

Note:

NDFA 'M' accepts $w \in \Sigma^*$ iff there exists at least one path (sequence of moves) leading to acceptance.

More on NFA :

NFA is like a DFA but with three extra features:

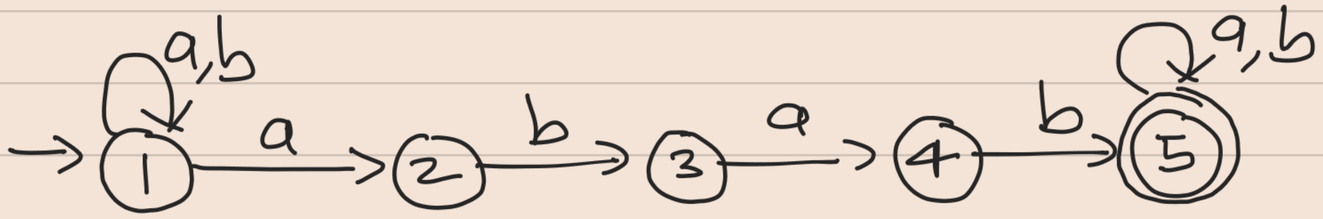
1. Easier to construct because they can be composed in a modular fashion.
2. Easier to read and they tend to be much smaller, therefore easy to describe as well.
3. Computationally, they are equivalent to DFA's, in the sense that they recognizes the same languages.

NFA's important features

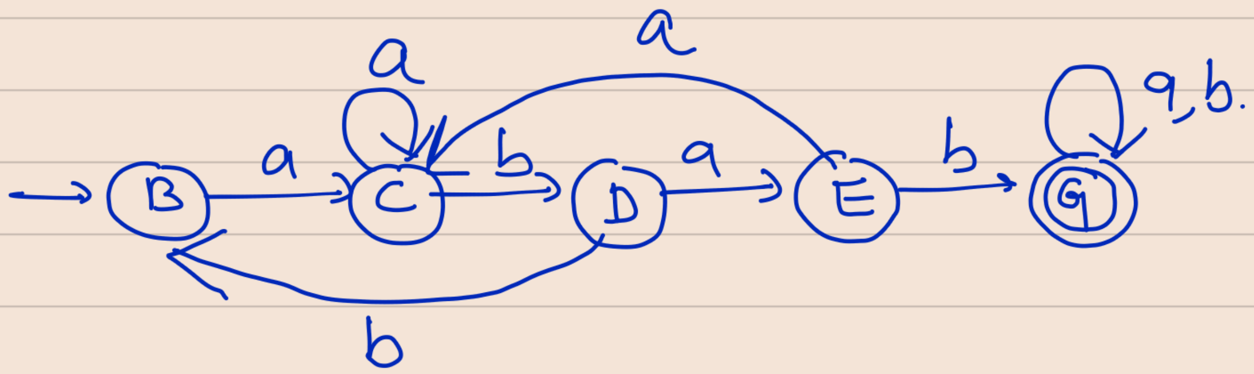
1. Epsilon transition possible.
   → Easy to represent optional things
2. Missing transitions
   → All possibilities need not be explored.
3. Multiple transitions
   → Need to guess which path will accept the string.

Ex:

the automata (NFA) below accepts string containing the substring abab.



the respective DFA, shown below, needs a lot more transitions and is somewhat harder to read.



Let $\Sigma' = \{0, 1, 2, \ldots 9\}$. Let us denote this as $[0, 9]$ in short form. Let

$$L = \{ w\#c \mid c \in \Sigma',\ w \in \Sigma'^{*} \text{ and } c \text{ occurs in } w \}$$

For example, the word 314159#5 is in L, and so is 314159#3. But the word 314159#7 is not in L.