# Determinism and Deterministic Finite Automata

Formal Definition: A Definitive finite automaton is a 5-tuple (Q, Σ, δ, q0, F), where.

1. Q is a finite set called the states

2. Σ is a finite set called the alphabet

3. δ : Q × Σ−→Q is the transition function

4. is the start state

5. F ⊆ Q is the set of accept states

Deterministic Finite Automata – Input is given and next state is known

1. First, every state of a DFA always has exactly one exiting transition arrow for each symbol in the alphabet
2. Second, in a DFA, labels on the transition arrows are symbols from the alphabet.
3. In a DFA, the transition function takes a state and an input symbol and produces the next state.
4. DFA cannot have an empty string as an input

Let M = (Q, Σ, δ, q0, F) be a finite automaton and let be a string where each wi is a member of the alphabet Σ. Then M accepts w if a sequence of states in Q exists with three conditions:

Condition 1 says that the machine starts in the start state.

Condition 2 says that the machine goes from state to state according to the transition function.

Condition 3 says that the machine accepts its input if it ends up in an accept state.

We say that M recognizes language A if A = {w| M accepts w}

A language is called a regular language if some finite automaton recognizes it.

Union operation: It simply takes all the strings in both A and B and lumps them together into one language.

Concatenation: The concatenation operation is a little trickier. It attaches a string from A in front of a string from B in all possible ways to get the strings in the new language.

Star: It works by attaching any number of strings in A together to get a string in the new language. Any number could possibly include 0

The class of regular languages is closed under the union operation. In other words, if and are regular languages, so is

The class of regular languages is closed under the concatenation operation. In other words, if A1 and A2 are regular languages then so is A1 ◦ A2.

# Nondeterminism and Nondeterministic Finite Automata

Nondeterminism is a generalization of determinism, so every deterministic finite automaton is automatically a nondeterministic finite automaton

Nondeterministic Finite Automata – At any point several choices may exist for the next state

1. In an NFA, a state may have zero, one, or many exiting arrows for each alphabet symbol.
2. In general, an NFA may have arrows labeled with members of the alphabet or ε. Zero, one, or many arrows may exit from each state with the label ε
3. ε represents an empty string input
4. In an NFA, the transition function takes a state and an input symbol or the empty string and produces the set of possible next states

Formal definition: A nondeterministic finite automaton is a 5-tuple (Q, Σ, δ, q0, F), where

1. Q is a finite set of states

2. Σ is a finite alphabet

3. δ : Q × Σε−→P(Q) is the transition function

4. q0 ∈ Q is the start state

5. F ⊆ Q is the set of accept states.

Let N = (Q, Σ, δ, q0, F) be an NFA and w a string over the alphabet Σ. Then we say that N accepts w if we can write w as , where each is a member of and a sequence of states exists in Q with three conditions:

1.

2 for , and

3.

Condition 1 says that the machine starts out in the start state

Condition 2 says that state is one of the allowable next states when N is in state and reading . Observe that is the set of allowable next states and so we say that is a member of that set

Condition 3 says that the machine accepts its input if the last state is an accept state

How does an NFA compute?

Suppose that we are running an NFA on an input string and come to a state with multiple ways to proceed. For example, say that we are in state q1 in NFA N1 and that the next input symbol is a 1. After reading that symbol, the machine splits into multiple copies of itself and follows all the possibilities in parallel. Each copy of the machine takes one of the possible ways to proceed and continues as before. If there are subsequent choices, the machine splits again. If the next input symbol doesn’t appear on any of the arrows exiting the state occupied by a copy of the machine, that copy of the machine dies, along with the branch of the computation associated with it. Finally, if any one of these copies of the machine is in an accept state at the end of the input, the NFA accepts the input string

# Equivalence of NFAs and DFAs

Say that two machines are equivalent if they recognize the same language. Every nondeterministic finite automaton has an equivalent deterministic finite automaton.