Finite Automata and Regular Expressions - I

A finite automaton has a finite set Q of states. Its input is a string of letters from a finite alphabet Σ. It starts at time 0 in a designated start state $q_0 \in Q$ at the left end of the input string. At time i it reads the i th letter of the input string and changes state according to a "transition function" - $\delta : Q \times \Sigma \to Q$. The set of states Q is partitioned into two parts - $F, Q \setminus F$, where F are the "final" or accepting states. If at the end, it is in one of the states in F, it "accepts" the string, otherwise it "rejects" the string. The set of all strings (over Σ) accepted by the automaton is called the "language" accepted by it.

Σ^* denotes the set of all finite length strings over Σ and a subset of Σ^* is said to be "regular" if and only if it is accepted by some finite auotmaton.

The above is called a deterministic finite automaton - DFA. A non-deterministic finite automaton - NFA - has the property that it can non-deterministically choose one of several possible next states - i.e., δ now is a function from $Q \times \Sigma$ to 2^Q. We say that an NFA accepts a string iff there is **some** valid computation of it on the string which leads to an accepting state.

Theorem (Equivalence of NFA and DFA) : For any language L accpeted by a NFA, there is a DFA which accepts L.

The proof gives a DFA which "simulates" the NFA accepting L. The DFA just "remembers" all possible states the NFA could have been in at each time; for this the set of states of the DFA just needs to be 2^Q, still finite.

DFA as well as NFA can be represented by graphs.

ϵ - moves : NFA with ϵ moves can make transitions labelled ϵ without using up any letter of the input.

Theorem A language accepted by an NFA with ϵ-moves can be accpeted by an NFA without ϵ moves.

Regular Expressions represent languages. The regular exprerssions over Σ and the sets they represent are defined recursively as follows :

1. ϕ is a regular expression and denotes the empty set.

2. ϵ is a regular expression and denotes the set $\{\epsilon\}$.

3. For each $a \in \Sigma$, a is a regular expression denoting the set $\{a\}$.

4. If r, s are regular expressions, denoting sets R, S respectively, then $(r + s)$, (rs) and (r^*) are regular expressions denoting sets $R \cup S$, RS (concatenation) and R^* respectively.

For a regular expression r, the set represented by it is denoted $L(r)$.

Theorem Let r be a regular expression. Then there exists an NFA with $\epsilon-$ transitions accepting $L(r)$

Theorem If L is accepted by a DFA, then L can be represented by a regular expression.