Automata

David Robinson

Finite Automaton

A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that consists of:

1. Q: A finite set of states
2. Σ: An alphabet
3. $\delta = Q \times \Sigma \to P(Q)$: A transition function
4. $q_0 \in Q$: A start state
5. $F \subseteq Q$: A set of accept states

Acceptance

Let $M = (Q, \Sigma, \delta, q_0, F)$ and $w = w_1 w_2 \dots w_n$ be a string of length n over Σ.

M **accepts** w if there exists a sequence of states in Q, $r_0, r_1, \dots, r_n$ so that:

1. $r_0 = q_0$
2. For i from 0 to $n - 1$, $\delta(r_i, w_{i+1}) = r + 1$
3. $r_n \in F$

Recognition

A machine M **recognizes** language A if $A = \{w \mid M \text{ accepts } w\}$.

Regular Language

A language is **regular** if and only if it can be recognized by a DFA.

Nondeterminism

A Nondeterministic Finite Automata (NFA):

1. Can have more than one possible transition per state per input symbol.
2. Doesn't have to have a transition for every state for every input symbol.
3. Can transition on the empty string.

A DFA accepts if *the* path for the input string ends on an accept state, while an NFA accepts if *any* path for the input string ends on an accept state.