

MIT 6.840 - Theory of Computation Lecture 1 - Introduction, Finite Automata, Regular Expressions

Lecture 1 - Introduction, Finite Automata, Regular Expressions

Finite Automata

A finite automaton is a model of computation. It consists of a set of “states”, and transitions between those states based on inputs. Finite automata are used to recognize patterns in strings, and they are the basis for regular expressions.

An automaton can be described formally as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where: 1. Q is a finite set of states. 2. Σ is a finite set of input symbols (the alphabet). 3. $\delta : Q \times \Sigma \rightarrow Q$ is the transition function, which takes a state and an input symbol and returns the next state. 4. $q_0 \in Q$ is the start state. 5. $F \subseteq Q$ is the set of accept states.

A string is accepted by the automaton if, starting from the start state q_0 and processing each symbol of the string in order, the automaton ends in an accept state after reading the entire string. Therefore, an automaton can either accept or reject a string. We call the set of all strings accepted by an automaton A its language:

$$L(A) = \{w \in \Sigma^* \mid A \text{ accepts } w\}$$

We say that A is the language of an automaton M and that M recognizes A , or $A = L(M)$. We say that a language is a regular language if there is a finite automaton that recognizes it. That is, a language is regular if we can build a finite automaton such that the automaton accepts all the strings in the language and only the strings in the language.