Finite Automata

Finite Automata

- A finite-state machine (finite automaton) is a simple, limited model of computation
  - states
  - transition from one state to another

Applications of FSM

- Controllers: elevators, ovens, stereo systems
- Logic circuit design: arithmetic and logic units, buffers
- Programming utilities: lex, grep, awk, perl
- Text editors: pattern-matching
- Computers

Characteristics of FSM

- Operate on strings.
- A finit set of states.

  Some states (“accepting” states) cause machine to get excited.
- The current state entirely dictates the next state.

  Rules: “If in state q_i, scanning 0, go to state q_j and move right one cell.”
- If light is on after reading the last character on the input, then M accepts the input.

Graphical Representation of FSM

- State diagram: a directed graph
  - nodes: states
  - arcs: transitions between states
  - start state: an arrow points at it from nowhere
  - accept state: with double circle.
  - For each state q and symbol $a \in \Sigma$,
    $\exists$ exactly one edge leaving q labelled with a.
- M accept w iff
  the path starting from q_0 labelled with w ends at a final state.

Formal Definition

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where

  Q: finite set of states

  Σ: finite input alphabet

  δ: $Q \times \Sigma \Leftrightarrow\!\rightarrow Q$, transition function

  $q_0 \in Q$: initial state

  $F \subseteq Q$: accept/final states

- Examples

Language of a Machine

- M recognizes language A: if A contains all strings that M accepts
- Def: $L(M) =$ " Language accepted by M"
  $\forall w$, $w \in L(M) \Leftrightarrow$ M accepts w
- A formal definition of " M accepts w"
  if there exists a sequence of states $r_0, r_1, \cdots, r_n$
  1. $r_0 = q_0$
  2. $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, 1, \cdots, n \Leftrightarrow 1$
  3. $r_n \in F$
- A language is a regular lauguage if some DFA recognizes it.
  $L \subseteq \Sigma^*$ is regular iff $\exists$ DFA M s.t. $L = L(M)$.