Regular Languages

Outline

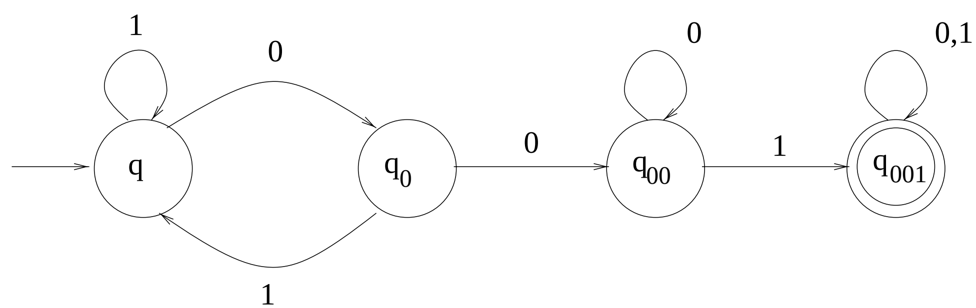
- Last time
  - What are finite automata?
  - Application, definition, graphical representation, language
- This time
  - Analysis of DFA
  - Design of DFA
  - Regular languages

Analysis of FSM

- Describe what language a FSM recognizes.
  Given M, find L(M).
- Examples

Design of FSM

- Given a language L, design M s.t. L(M) = L.
- Method: "reader as automaton"
  - pretending you are the machine
  - figuring out what you need to remember (finite memory)
  - assigning states for crucial information
- Examples
  - Design M accepts
    L = $\{ w|w$ contains 001 as a substring, $w \in \{0,1\}^* \}$



Operations of Languages

- Theory of computation: languages and how to manipulate them
- Regular operations

  Let A, B be languages.

  - union: $A \cup B = \{x | x \in A \text{ or } x \in B\}$
  - concatenation: $A \circ B = \{xy | x \in A \text{ and } y \in B\}$
  - star: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \text{ and each } x_i \in A\}$

Properties of Regular Languages

- A set A is closed under operation $P(x, y)$ if
  $P : A \times A \longrightarrow B \subseteq A$
  - Let $N = \{1, 2, 3, \cdots\}$,
    N is closed under $\times$, but not under $/$.
- The class of regular languages is closed under $\cup$.
  If A_1, A_2 are regular languages, so is $A_1 \cup A_2$.
  Proof: proof by construction.

Proof

Assume A_1: $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$
A_2: $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$
Construct $M = (Q, \Sigma, \delta, q, F)$ s.t. $L(M) = A_1 \cup A_2$

1. $Q = \{(r_1, r_2) | r_1 \in Q_1, r_2 \in Q_2\}$
2. To make things simple, assume the same Σ.
3. δ: For each $(r_1, r_2) \in Q$ and each $a \in \Sigma$, let
   $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
4. $q_0 = (q_1, q_2)$
5. $F = \{(r_1, r_2) | r_1 \in F_1 \text{ or } \ r_2 \in F_2\}$

More Properties of Regular Languages

- The class of regular languages is closed under concatenation.

  If A_1, A_2 are regular languages, so is $A_1 \circ A_2$.

  Proof: not quite easy.

  Let's introduce a new technique: nondeterminism.