Recognizing Languages

- regular expressions provide a way to mechanically generate languages
- how to mechanically recognize languages?

- this would be useful for implementing pattern-matching, identifying and compiling legal programs, …

Recognizing Languages

Definition 3.5. Formally, a ***deterministic finite-state automaton*** M is specified by 5 components: $M = (Q, \Sigma, q_0, \delta, F)$ where

- Q is a finite set of states;
- Σ is an alphabet called the *input alphabet*;
- $q_0 \in Q$ is a state which is designated as the *start state*;
- F is a subset of Q; the states in F are states designated as *final* or *accepting* states;
- δ is a transition function that takes <state, input symbol> pairs and maps each one to a state: $\delta : Q \times \Sigma \to Q$. To say $\delta(q, a) = q'$ means that if the machine is in state q and the input symbol a is consumed, then the machine will move into state q'. The function δ must be a total function, meaning that $\delta(q, a)$ must be defined for every state q and every input symbol a. (Recall also that, according

- the states act as a memory for what has been matched so far in the string
- the transitions capture what can come next
- the accepting states indicate when the match is complete

Recognizing Languages

Definition 3.5. Formally, a ***deterministic finite-state automaton*** M is specified by 5 components: $M = (Q, \Sigma, q_0, \delta, F)$ where

- Q is a finite set of states;
- Σ is an alphabet called the *input alphabet*;
- $q_0 \in Q$ is a state which is designated as the *start state*;
- F is a subset of Q; the states in F are states designated as *final* or *accepting* states;
- δ is a transition function that takes <state, input symbol> pairs and maps each one to a state: $\delta : Q \times \Sigma \to Q$. To say $\delta(q, a) = q'$ means that if the machine is in state q and the input symbol a is consumed, then the machine will move into state q'. The function δ must be a total function, meaning that $\delta(q, a)$ must be defined for every state q and every input symbol a. (Recall also that, according

The ***language accepted by*** M, denoted $L(M)$, is the set of all strings $w \in \Sigma^*$ that are accepted by M: $L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$.

$\delta^*(q,w)$ = state after consuming w, starting from state q

$\delta^*(q,\varepsilon) = q$

Specifying DFAs

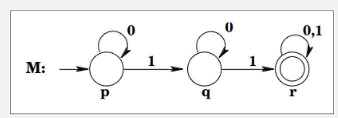
	p	q	r
0	p	q	r
1	q	r	r

- transition table
  - states p, q, r
  - inputs 0, 1

$M = (\{p, q, r\}, \{0, 1\}, p, \delta, \{r\})$, where δ is given by

$$\delta(p,0) = p \qquad \delta(p,1) = q$$
$$\delta(q,0) = q \qquad \delta(q,1) = r$$
$$\delta(r,0) = r \qquad \delta(r,1) = r$$

- formal definition

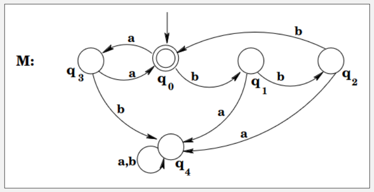


- transition diagram
  - incoming arrow denotes the start state (p)
  - double circles denote accepting states (r)

$L(M) = \{ x \in \{0,1\}^* \mid n_1(x) \geq 2 \}$

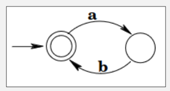
$L(M) = L(0^*10^*1(0|1)^*)$

Shortcuts



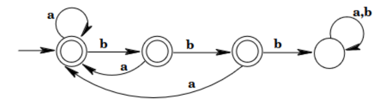
L(*M*) = L((aa|bbb)*)

- q_4 is a garbage or trap state – a non-accepting state which is not possible to escape
  - reflects having encountered something that disqualifies the string from being in the language
- such states are commonly omitted from the transition diagram
  - this is no longer a complete DFA! (but it is understood how to turn it into a complete DFA)

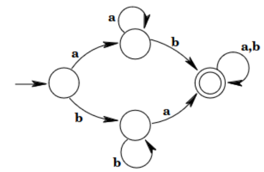


2. What languages do the following DFAs accept?

a)



b)



1. Give DFAs that accept the following languages over $\Sigma = \{a, b\}$.

a) $L_1 = \{x \mid x \text{ contains the substring } aba\}$
b) $L_2 = L(a^*b^*)$
c) $L_3 = \{x \mid n_a(x) + n_b(x) \text{ is even }\}$
d) $L_4 = \{x \mid n_a(x) \text{ is a multiple of 5 }\}$
e) $L_5 = \{x \mid x \text{ does not contain the substring } abb\}$
f) $L_6 = \{x \mid x \text{ has no } a\text{'s in the even positions}\}$
g) $L_7 = L(aa^* \mid aba^*b^*)$

3. Let $\Sigma = \{0, 1\}$. Give a DFA that accepts the language

$$L = \{x \in \Sigma^* \mid x \text{ is the binary representation of an integer divisible by } 3\}.$$