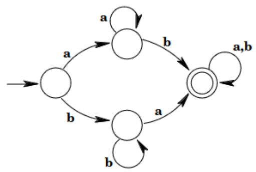
2. What languages do the following DFAs accept?

b)



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

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) $L_6 = \{x \mid x \text{ has no } a\text{'s in the even positions}\}$

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

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

Recognizing Languages

Definition 3.7. Formally, a nondeterministic finite-state automaton M is specified by 5 components: $M = (Q, \Sigma, q_0, \partial, F)$ where

- Q, Σ, q_0 and F are as in the definition of DFAs;
- ∂ is a transition function that takes <state, input symbol> pairs and maps each one to a set of states. To say $\partial(q, a) = \{q_1, q_2, \ldots, q_n\}$ means that if the machine is in state q and the input symbol a is consumed, then the machine may move directly into any one of states $q_1, q_2, \ldots, q_n$. The function ∂ must also be defined for every <state,ε> pair. To say $\partial(q, \varepsilon) = \{q_1, q_2, \ldots, q_n\}$ means that there are direct ε-transitions from state q to each of $q_1, q_2, \ldots, q_n$.

  The formal description of the function ∂ is $\partial : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$.

NDFAs allow –

- more than one transition involving the same state and symbol – $\partial(q,a)$ is a set
- ε-transitions – can change state without consuming input

Recognizing Languages

Definition 3.7. Formally, a nondeterministic finite-state automaton M is specified by 5 components: $M = (Q, \Sigma, q_0, \partial, F)$ where

- Q, Σ, q_0 and F are as in the definition of DFAs;
- ∂ is a transition function that takes <state, input symbol> pairs and maps each one to a set of states. To say $\partial(q, a) = \{q_1, q_2, \ldots, q_n\}$ means that if the machine is in state q and the input symbol a is consumed, then the machine may move directly into any one of states $q_1, q_2, \ldots, q_n$. The function ∂ must also be defined for every <state,ε> pair. To say $\partial(q, \varepsilon) = \{q_1, q_2, \ldots, q_n\}$ means that there are direct ε-transitions from state q to each of $q_1, q_2, \ldots, q_n$.

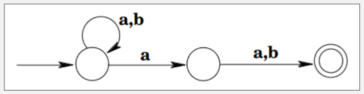
  The formal description of the function ∂ is $\partial : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$.

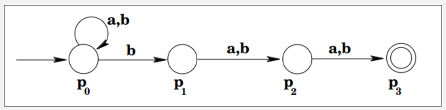
Definition 3.8. Let $M = (Q, \Sigma, q_0, \partial, F)$ be a nondeterministic finite-state automaton. The string $w \in \Sigma^*$ is ***accepted*** by M iff $\partial^*(q_0, w)$ contains at least one state $q_F \in F$.

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

accept *w* if it is possible to end up in an accepting state

- what language is accepted?





Equivalence of DFAs and NDFAs

- every language accepted by a DFA is accepted by an NDFA
  - a DFA is (essentially) an NDFA – NDFA does not require multiple or ε-transitions, and for $\delta(q,a) = q'$, $\partial(q,a) = \{q'\}$

- every language accepted by an NDFA is accepted by a DFA

Theorem 3.2. *Every language that is accepted by an NFA is accepted by a DFA.*

  - proof idea: give an algorithm for constructing an equivalent DFA from an NDFA (then prove the algorithm correct)