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

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

- observation
  - let *v* be the value of the bits read so far (starting from the left)
  - let *b* be the next bit read
  - then the new value read so far is 2*v*+*b*
    - adding one more place to the right = left shift, which multiplies by 2
- idea
  - states track the remainder
    - use state q_i to correspond to remainder *i*
  - if *v* mod 3 = *r*, then (2*v*+*b*) mod 3 = (2*r*+*b*) mod 3
    - v mod 3 = r means v = 3k+r for integer k
    - 2v+b = 2(3k+r)+b = 6k+2r+b
    - 6k mod 3 = 0, so (2v+b) mod 3 = (2r+b) mod 3

$\rightarrow \delta(q_i, b) = q_{(2i+b) \bmod 3}$

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 – ∂(*q*,*a*) is a set
- ε-transitions – can change state without consuming input

Recognizing Languages

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 $q_F \in F$.

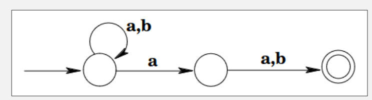
The ***language accepted by*** M, denoted $L(M)$, is the set of all strings $w \in \Sigma^*$ that are accepted by M: $L(M) = \{w \in \Sigma^* \mid \partial^*(q_0, w) \cap F \neq \emptyset\}$.

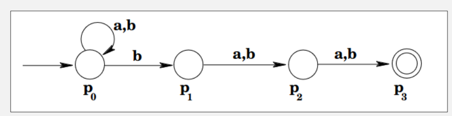
- an NDFA accepts *w* if it is *possible* to end up in an accepting state

for a DFA, δ*(*q*,*w*) = state reached after consuming *w*, starting from state *q*

for an NDFA, ∂*(*q*,*w*) = set of states that are possible to reach after consuming *w*, starting from state *q*

- 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)

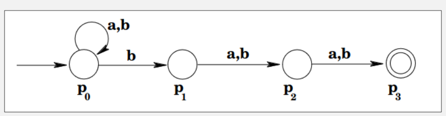
Equivalence of DFAs and NDFAs

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

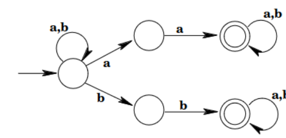
- let NFA $N = (P,\Sigma,p_0,\partial,F)$ and DFA $D = (Q,\Sigma,q_0,\delta,F)$
- idea – the states of the DFA D correspond to sets of states in the NDFA N
- q_0 corresponds to $\partial^*(p_0,\varepsilon)$
- repeatedly
  - find a state q that has been added to D but whose out-transitions have not yet been added
  - for each input symbol a, look at all of N's states that can be reached from any one of the $p_1, p_2, \ldots, p_n$ corresponding to q by consuming a (include ε-transitions)
    - add state $q' = \partial^*(p_1,a) \cup \ldots \cup \partial^*(p_n,a)$ if not already present
    - add transition $\delta(q,a) = q'$ to D
- accepting states of D are those corresponding to at least one of N's

NFA to DFA

- q_0 corresponds to $\partial^*(p_0,\varepsilon)$
- repeatedly
  - find a state q that has been added to D but whose out-transitions have not yet been added
  - for each input symbol a, look at all of N's states that can be reached from any one of the $p_1, p_2, \ldots, p_n$ corresponding to q by consuming a (include ε-transitions)
    - add state $q' = \partial^*(p_1,a) \cup \ldots \cup \partial^*(p_n,a)$ if not already present
    - add transition $\delta(q,a) = q'$ to D
- accepting states of D are those corresponding to at least one of N's



2. Give a DFA that accepts the language accepted by the following NFA.



3. Give a DFA that accepts the language accepted by the following NFA. (Be sure to note that, for example, it is possible to reach both q_1 and q_3 from q_0 on consumption of an a, because of the ε-transition.)