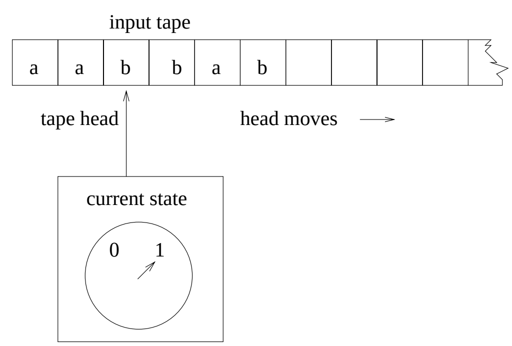
CPS 140 - Mathematical Foundations of CS
Dr. Susan Rodger
Section: Finite Automata (Ch. 2) (handout)

Deterministic Finite Accepter (or Automata) (Read Ch. 2.1-2.2)

A DFA=(K,Σ,δ,q_0,F)

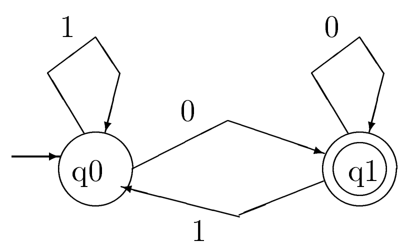


where

K is finite set of states
Σ is tape (input) alphabet
q_0 is initial state
F $\subseteq$ K is set of final states.
δ:K$\times\Sigma\rightarrow$K

Example: Create a DFA that accepts even binary numbers.

Transition Diagram:



M=(K,Σ,δ,q_0,F) =

Tabular Format

	0	1
q0	q1	q0
q1	q1	q0

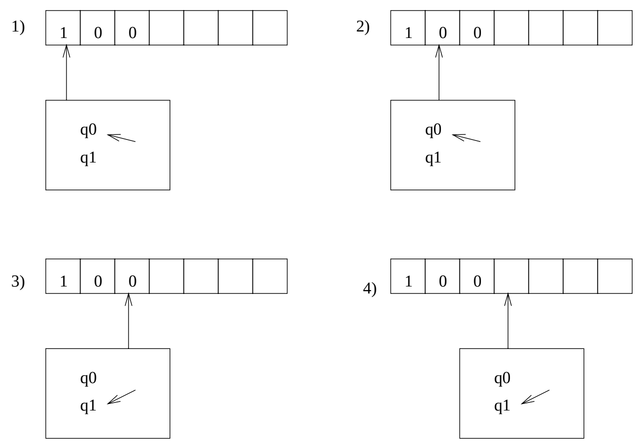
Example of a move: δ(q0,1)=

Algorithm for DFA:

Start in start state with input on tape
q = current state
s = current symbol on tape
while (s != blank) do
    q = δ(q,s)
    s = next symbol to the right on tape
if q$\in$F then accept

Example of a trace: 11010

Pictorial Example of a trace:



Definition:

Configuration: element of $K \times \Sigma^*$

Move between configurations: $\vdash$

Move between several configurations: $\vdash *$

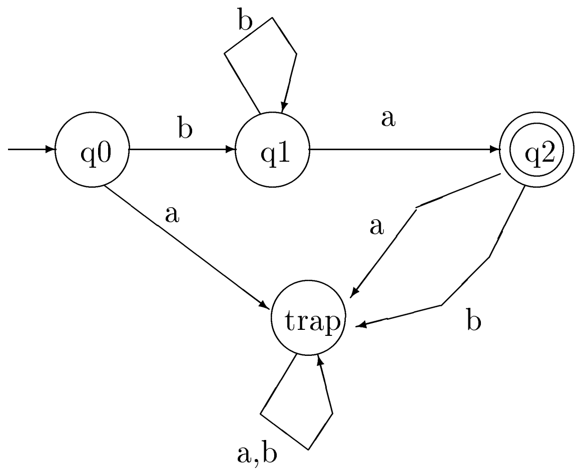
Examples (from prev FA):

Definition The language accepted by a DFA M=(K,Σ,δ,q_0,F) is set of all strings on Σ accepted by M. Formally,

L(M)=$\{w \in \Sigma^* \mid (q_0, w) \vdash *(p, \epsilon), p \in F\}$

Trap State

Example: L(M) = $\{b^n a \mid n > 0\}$



You don't need to show trap states! Any arc not shown will by default go to a trap state.

Example: Create a DFA that accepts even binary numbers that have an even number of 1's.

Definition A language is regular iff there exists DFA M s.t. L=L(M).

Chapter 2.2

Nondeterministic Finite Automata (or Accepter)

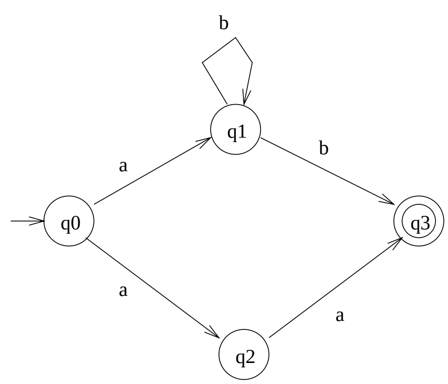
Definition

An NFA=(K,Σ,Δ,q_0,F)

where

K is finite set of states
Σ is tape (input) alphabet
q_0 is initial state
F $\subseteq$ K is set of final states.
Δ: subset of K$\times(\Sigma \cup \{\epsilon\})\times$K

Example



Note: In this example with state q_0 and input a,

Notation: $\Delta(q, a)$ = set of states reachable from q on a

$\Delta(q_0, a) =$

Example

L=$\{(ab)^n \mid n > 0\} \cup \{a^n b \mid n > 0\}$

Definition $(q_i, w) \vdash *(q_j, \epsilon)$ if and only if there is a walk from q_i to q_j labeled w.

Example From previous example:

What is q_j in $(q_0, ab) \vdash *(q_j, \epsilon)$?

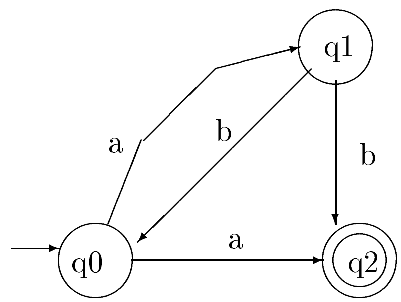
What is q_j in $(q_1, aba) \vdash *(q_j, \epsilon)$?

Definition: For an NFA M, L(M)=$\{w \in \Sigma^* \mid \exists p \in F s.t. (q_0, w) \vdash *(p, \epsilon)\}$

The language accepted by nfa M is all strings w such that there exists a walk labeled w from the start state to final state.

NFA vs. DFA: Which is more powerful?

Example:



Theorem Given an NFA M_N=$(K_N, \Sigma, \Delta_N, q_0, F_N)$, then there exists a DFA M_D=$(K_D, \Sigma, \delta_D, q_0, F_D)$ such that $L(M_N) = L(M_D)$.

Proof:

We need to define M_D based on M_N.

$K_D =$

$F_D =$

δ_D :

Definition: $E(q)$ is the closure of the set $\{q\}$

$E(q) = \{p \in K \mid (q, \epsilon) \vdash *(p, \epsilon)\}$

Algorithm to construct M_D

1. start state is $E(q_0)$
2. While can add an edge
   (a) Choose a state A=$\{q_i, q_j, ...q_k\}$ with missing edge for $a \in \Sigma$
   (b) Compute B = $\Delta(q_i, a) \cup \Delta^(q_j, a) \cup \ldots \cup \Delta(q_k, a)$
   (c) apply closure to B, B = E(B)
   (d) Add state B if it doesn't exist
   (e) add edge from A to B with label a
3. Identify final states

Note this proof is different than the proof in the book. In the book instead of starting with the start state, it takes the closure of the start state, including all states reachable on ϵ

Example: