Finite State Machines

Read K & S 2.1
Do Homeworks 4 & 5.

Finite State Machines 1

A DFSM to accept odd integers:

Definition of a Deterministic Finite State Machine (DFSM)

M = (K, Σ, δ, s, F), where
- K is a finite set of states
- Σ is an alphabet
- $s \in K$ is the initial state
- $F \subseteq K$ is the set of final states, and
- δ is the transition function. It is function from $(K \times \Sigma)$ to K

i.e., each element of δ maps from: a state, input symbol pair to a new state.

Informally, M **accepts** a string w if M winds up in some state that is an element of F when it has finished reading w (if not, it **rejects** w).

The **language accepted by M**, denoted L(M), is the set of all strings accepted by M.

Deterministic finite state machines (DFSMs) are also called deterministic finite state automata (DFSAs or DFAs).

Computations Using FSMs

A **computation** of A FSM is a sequence of configurations, where a configuration is any element of $K \times \Sigma^*$.
The **yields** relation $|-_M$:
(q, w) $|-_M$ (q', w') iff
- w = a w' for some symbol $a \in \Sigma$, and
- δ (q, a) = q'

(The yields relation effectively runs M one step.)

$|-_M^*$ is the reflexive, transitive closure of $|-_M$.
(The $|-_M^*$ relation runs M any number of steps.)

Formally, a FSM M **accepts** a string w iff
(s, w) $|-_M^*$ (q, ε), for some $q \in F$.

An Example Computation

A DFSM to accept odd integers:
On input 235, the configurations are:
(q0, 235) $|-_M$ (q0, 35)
$|-_M$
$|-_M$

Thus (q0, 235) $|-_{M^*}$ (q1, ε). (What does this mean?)

Finite State Machines 2

A DFSM to accept $.50 in change:

More Examples

((aa) ∪ (ab) ∪ (ba) ∪ (bb))*

(b ∪ ε)(ab)*(a ∪ ε)

More Examples

L1 = {w ∈ {a, b}* : every a is immediately followed a b}

A regular expression for L1:

A DFSM for L1:

L2 = {w ∈ {a, b}* : every a has a matching b somewhere before it}

A regular expression for L2:

A DFSM for L2:

Another Example: Socket-based Network Communication

Client	Server	Σ = {Open, Req, Reply, Close}
open socket		
send request		
	send reply	L = Open (Req Reply)* (Req $\cup$ ε) Close
send request		
	send reply	
…		M =
close socket		

Definition of a Deterministic Finite State Transducer (DFST)

M = (K, Σ, O, δ, s, F), where

K is a finite set of states
Σ is an input alphabet
O is an output alphabet
$s \in K$ is the initial state
$F \subseteq K$ is the set of final states, and
δ is the transition function. It is function from

$(K \times \Sigma)$ to $(K \times O^*)$

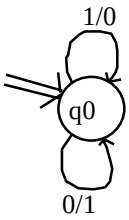
i.e., each element of δ maps from: a state, input symbol pair
to : a new state and zero or more output symbols (an output string)

M **computes a function M(w)** if, when it reads w, it outputs M(w).

Theorem: The output language of a deterministic finite state transducer (on final state) is regular.

A Simple Finite State Transducer

Convert 1's to 0's and 0's to 1's (this isn't just a finite state task -- it's a one state task)



An Odd Parity Generator

After every three bits, output a fourth bit such that each group of four bits has odd parity.