Lecture 6

Homework #5: 2.1.1, 2.1.2 a-d*, 2.1.3 b & c & d,
2.1.4 a (i)&(ii), b (i),(ii),(iii),(v)

From last time:

Recall that:
an **alphabet** is a finite set of symbols
a **string** is a finite sequence of symbols from an alphabet
a **language** is a set of strings over an alphabet

NOTE:
in genl, will use a, b, c to denote symbols;
x, y, z, u, v, w to denote words;
Σ will often be used to denote an alphabet;
L will often be used to denote a language

Regular languages – class of languages that can be expressed by regular expressions

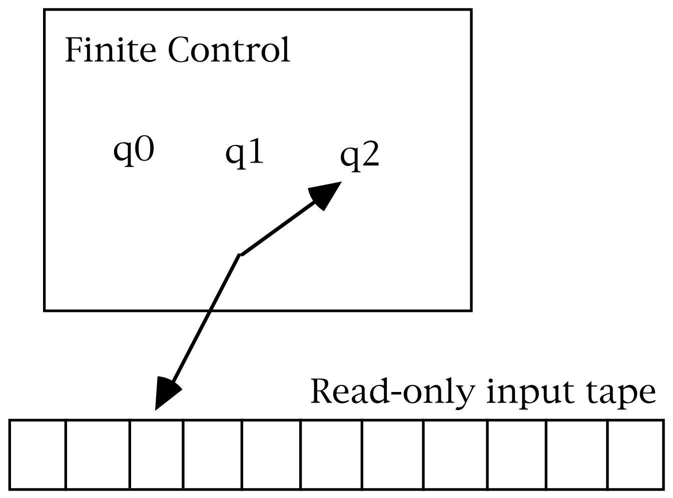
Now let's move to the flip-side: the computational models that work on these strings of data.

Computational models are based on our general notion of computers:
CPU, memory, I/O

We will start with simple (restricted) models and work our way up.

Finite Automaton
has: a very limited processor
no memory (except for state)
input on a tape
no output other than accept/reject

Deterministic Finite Automaton - operation completely determined by input:



Starts with:

1) tape head on leftmost square
   (tape head will move right after reading each input)
2) finite control in some designated state (the start state)

Operation:

1) read input square
2) depending on symbol and current state, go to new state
   (completely determined)
3) advance read head to next square
4) when read head reaches end, check state - certain states
are designated as "accept" states

Formal description of a DFA:

A DFA is a quintuple $M = (K, \Sigma, \delta, s, F)$, where

K is a finite set of states;
Σ is an alphabet;
$s \in K$ is the start state
$F \subseteq K$ is the set of final states

$\delta: K \times \Sigma \to K$
is the transition function

δ encodes rules for moving from 1 state to another.
$\delta(q,\sigma) = q2$ means
in state q, when reading σ, go to state q2.

Example.

$K = \{q0, q1\}$
$\Sigma = \{a, b\}$
$s = q0$
$F = \{q0\}$

q	σ	$\delta(q,\sigma)$
q0	a	q0
q0	b	q1
q1	a	q1
q1	b	q0

Say we have the string abba $\Rightarrow$ accept

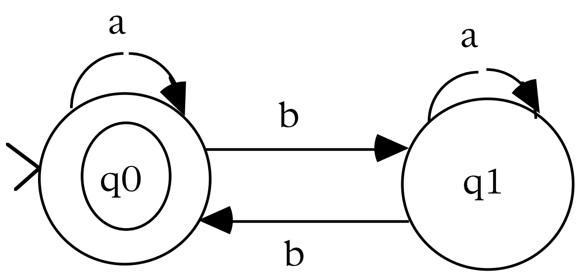
What kind of strings get accepted? Those with an even number of b's (including e).

Def. the **language accepted** by a machine is the set of strings it accepts.

L(M) is the set of all **strings accepted by M.**

Another representation makes certain properties of the automaton clearer: **state diagram**

states = nodes
transitions = directed arcs

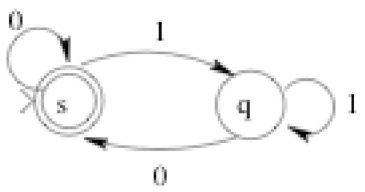


Note the way that initial and final (start and accept) states are labeled.

Now consider another example.

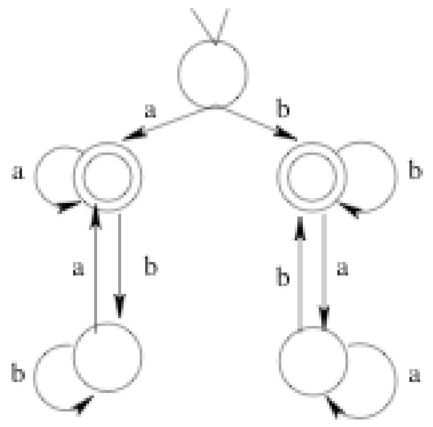
K = {s, q}
Σ = {0, 1}
s is the start state
F = {s}

q	σ	δ(q,σ)
s	1	q
s	0	s
q	1	q
q	0	s



What kinds of strings does this DFA accept?

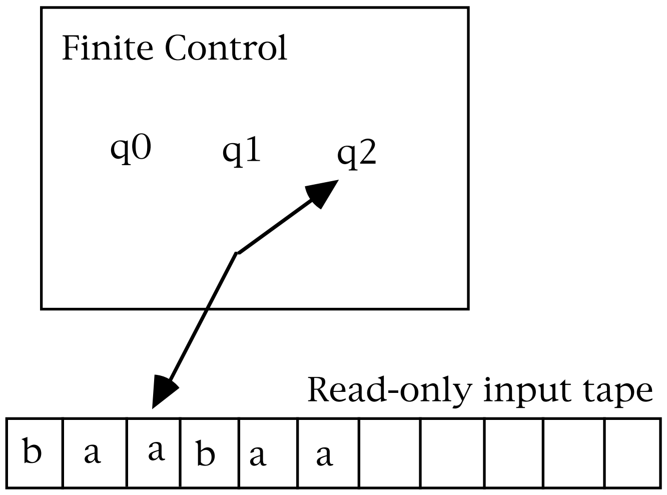
Now consider one more example. What does the following DFA accept?



Exercise.
Give deterministic finite automata to recognize integers and reals.

Notation (definitions) for the transitions (computations) in these machines:

Let (q, w) represent the **current configuration** of a DFA, where w is the string from the tape head to the end of the tape (all symbols upon which the DFA has not yet acted).



Current config is: (q2, abaa)

Define a **relation** $\vdash_M$ between 2 configurations that holds iff M can pass from 1 config to the other in 1 step.

$(q,w) \vdash_M (q^1,w^1)$ iff $\delta(q,\sigma) = q^1$ and $w = \sigma w^1$, $\sigma \in \Sigma$.

$\vdash_M^*$ is the relexive transitive closure of M. ("**yields**")

$(q,w) \vdash_M^* (q^1,w^1)$ after some number of steps (possibly 0)

Note that the state diagram represents the $\vdash_M$ relation.