Lecture 2 1/6/2010

Finite State Automaton (FSA)

5 pieces

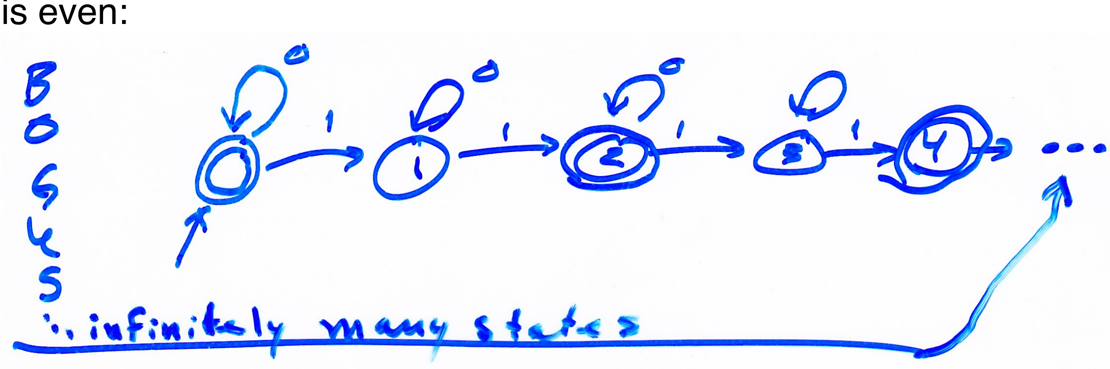
- States
- Alphabet
- Transitions
- Start
- Final or Accept

An Example

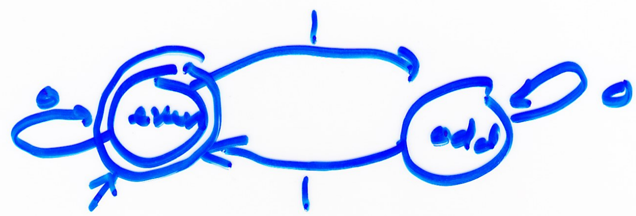
$\Sigma = \{0, 1\}$

$L = \{ w \in \Sigma^* \mid$ # of 1's in w is even $\}$

The "obvious" algorithm: first count the 1's, then decide whether the count is even:



It works, but is not a finite state machine. This is:



Formal definition:

A finite state machine

$M = (Q, \Sigma, \delta, q_0, F)$

where

Q is a finite set (states)

$q_0 \in Q$ start state

Σ is a finite set (alphabet)

$F \subseteq Q$ Final states
Accepting states

$\delta : Q \times \Sigma \to Q$ transition function

function

Formal version of the above example (done 2 different ways):

$M_{parity} = (Q, \Sigma, \delta, q_0, F)$

where

$Q = \{even, odd\}$

$\Sigma = \{0, 1\}$

$q_0 = even$ (one element)

$F = \{even\}$ (a set containing one element)

$\delta(q, a)$:

q \ a	0	1
even	even	odd
odd	odd	even

Even more succinctly if we let $Q = \{0, 1\}$ also

then $\delta(q, a) = (q + a) \bmod 2 \quad \forall q \in Q, \forall a \in \Sigma$

2.5

Another example:

$\Sigma = \{a, b\}$

$L = \{ w \mid$ 3rd letter from the right end of w is "a" $\}$

aab Y
bab N
abb Y
aaa Y

ab
a
ε
) N