Math 294 Week 14 - Deterministic Finite Automata

A **binary string** is a finite sequence of 0's and 1's.

We can use binary strings to represent (basically) any sort of data that we want, so they are an important object of study in theoretical computer science.

eg Any natural number can be written in a binary expansion.

eg If we use the encoding scheme

$a = 00000$ $\quad g = 00110$

$b = 00001$ $\quad h = 00111$

$c = 00010$ $\quad i = 01000$

$d = 00011$ $\quad j = 01001$ $\quad$ and so on...

$e = 00100$ $\quad k = 01010$

$f = 00101$ $\quad l = 01011$

we can encode any English word.

eg "bee" becomes "000010010000100".

A **decision problem** is a function mapping binary strings to {yes, no}.

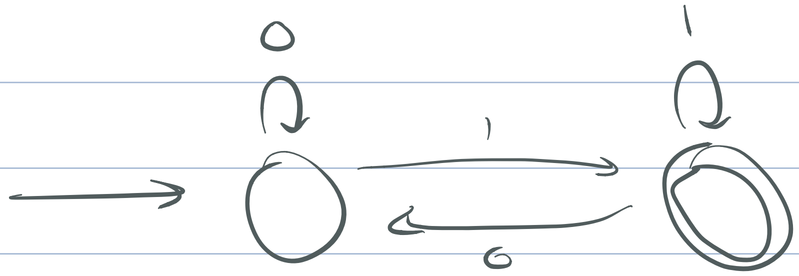
eg

$$f(s) = \begin{cases} \text{yes} & \text{if } s \text{ is the binary representation of a prime number} \\ \text{no} & \text{otherwise} \end{cases}$$

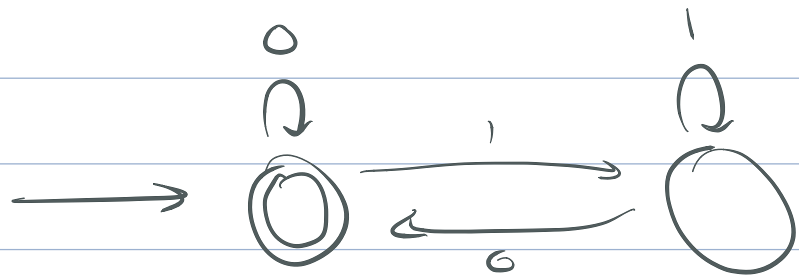
$$g(s) = \begin{cases} \text{Yes} & \text{if } s \text{ is the representation of an English word} \\ \text{no} & \text{otherwise} \end{cases}$$

DFAs

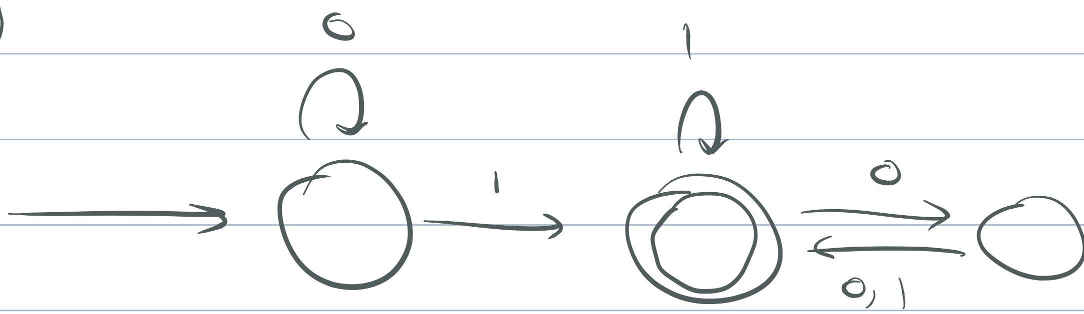
(1a)



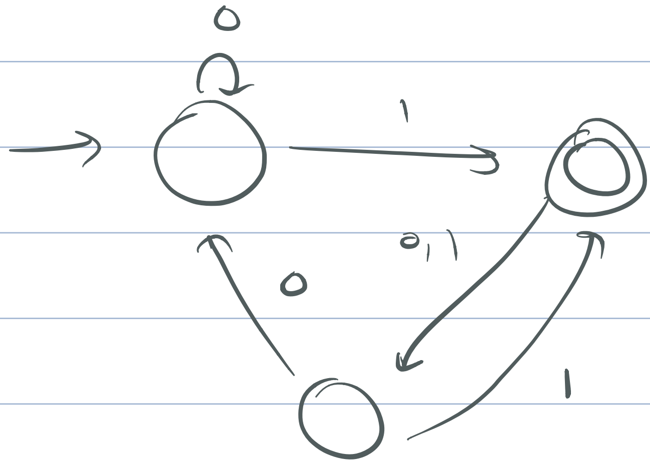
(1b)



(2)



(3)



Q Suppose we are given a DFA M.
Can we construct a DFA M^{opp} that does the exact opposite of M?
(i.e. M^{opp} accepts everything M rejects
and rejects everything M accepts)

Q Suppose we are given two DFAs M and N.
Can we construct a DFA Z that accepts exactly the strings accepted by either M or N?