CSE 396 — Lecture Tue 2/7/17 — Spring 2017

Formal Defn of a DFA — and an "Obj Oriented" version

deterministic finite automaton

A (DFA) is a 5-tuple $M = (Q, \Sigma, \delta, s, F)$ where

- Q is a finite set of <u>states</u>
- Σ is the <u>input alphabet</u>
- s, a member of Q, is the <u>start state</u>
- F, a subset of Q, is the set of (desired) <u>final states</u> also called accepting states, and
- $\delta : Q \times \Sigma \to Q$ is the transition function.

Insofar as a function $f : A \to B$ can be listed out as the set of ordered pairs (a, b) where $b = f(a)$, we can also list out δ as a set of ordered triples called <u>instructions</u>.

(works for NFA too)

```
class DFA {

   set<State> Q;

   set<char> Sigma;

   State s;

   set<State> F;
```

now a subtlety: You might expect "delta" to be a method:

```
   State delta(State q, char c);
```

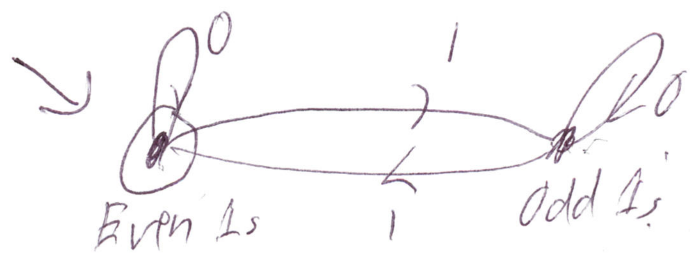
<u>But</u>, then every DFA would have to have the same delta. What you really want is for the class to have a <u>member</u> function, aka <u>delegate</u> (in C#) aka pointer-to-function in C/C++:

```
   State (*delta)(State q, char c);
};
```

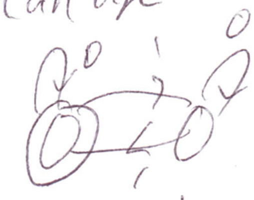
```
set<triple<State, char, State>> delta;
```

Example

Parity Check DFA:



Can use



state two.

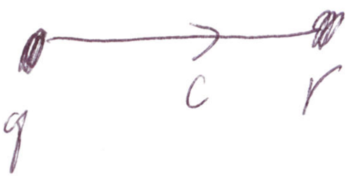
Abstract picture:

$\delta(q,c) = r$

instruction

(q, c, r)

usually



$\#c(x)$ = the number of times the char c appears in the string x.

$Q = \{Even, Odd\}$

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

$s = Even$

$F = \{Even\}$

> note these have different type.

$\delta(Even, 0) = Even \quad \delta(Odd, 0) = Odd$

$\delta(Even, 1) = Odd \quad \delta(Odd, 1) = Even$

OR $\delta = \{(Even, 0, Even), (Even, 1, Odd), (Odd, 0, Odd), (Odd, 1, Even)\}$

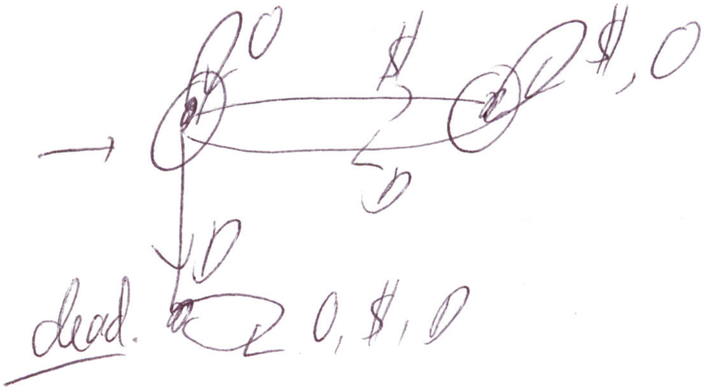
as a set of instructions

OK not to write out whole tables and just give the diagram so long as every state has a comment saying its purpose or meaning.

Given a DFA M, $L(M) = \{x \in \Sigma^*: M$ when run on input x ends up in a state in $F\}$.

Language of M. Here, $L(M) = \{x : \#1(x)$ is even$\}$

DragonSlayer DFA for This: $\Sigma = \{0, \$, D\}$

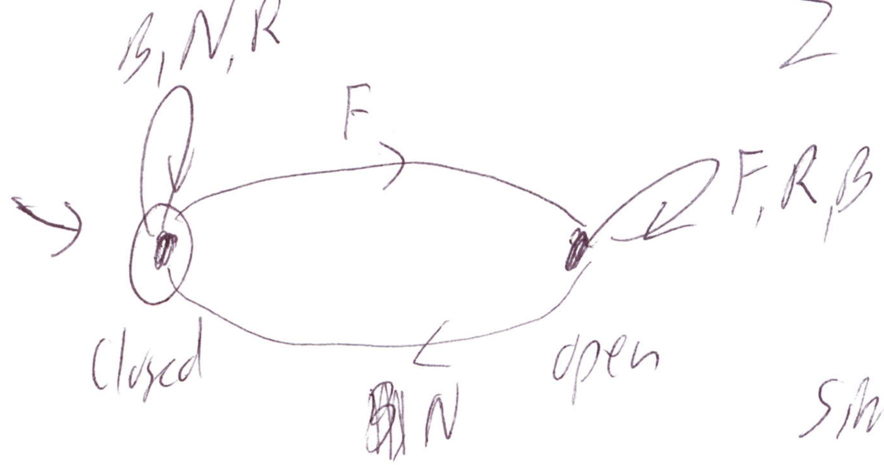


A DFA must define an option for every pair (q, c) even if $q =$ "dead". (An NFA will be able to leave some cases out.)

Text Automated Door Example: $Q = \{Open, Closed\}$

$\Sigma = \{N, F, R, B\}$

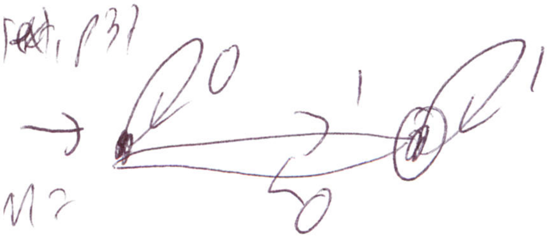
nobody, front only, rear only, both.



Similar but different to parity check example

door begins closed and we desire that it ends closed.

text, p37



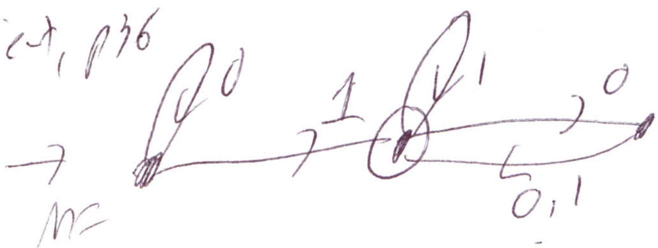
M=

$L(M) = \{x \in \{0,1\}^* : x \text{ ends in a } 1\}$

Regular expression: $(0 \cup 1)^* 1$

any random 1s & 0s followed by a last 1.

text, p36



M=

$L(M) = \{x : x$ has at least one 1 and ~~does not end with 10~~$\}$.

$0^*1(1 + 00 + 01)^*$

Not so easy to define in words!

Lecture page 4, if time allows one more example.

$L(M) = \{x : \#1(x)$ is congruent to $1 \bmod 3\}$