CSE 596 Lecture Wed 8/29 Fall 2018

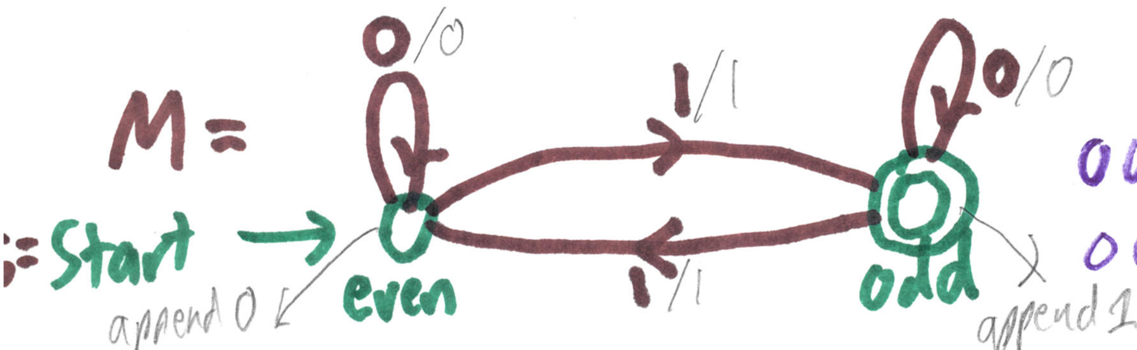
Example of a Deterministic Finite Automaton (DFA): Check whether parity of a binary string is od

00101001|x # of 1s is 3, which is odd.

00101001|1 # is 4, even. Components are:

- States $Q = \{\text{'even'}, \text{'odd'}\}$ Means: Parity of the 1's read thus far.
- Start in even, because zero is an even number
- Goal: state odd $F = \{\text{odd}\}$ desired Final state
- Characters $\Sigma = \{\text{'0'}, \text{'1'}\}$ • Rules δ shown in a picture

M =



00101001 $\in$ L(M)

001010011 $\notin$ L(M)

L(M), the language of M, is the set of binary strings that cause M to end in a goal state when started at S

Formal Defn, in "Tuple Style" pioneered for math in the 1920s. Example: A Field is a 5-tuple (F, +, •, 0, 1) where [laws of + and •, identity, div, etc. hold]

Def^n: A DFA is a 5-tuple $M = (Q, \Sigma, \delta, s, F)$

where: Q is a *finite* set of states

s, a member of Q, is the start state (Most texts write q_0)

F, a *subset* of Q, is the set of desired final states

Σ is a finite alphabet of chars, and

δ is a function from $Q \times \Sigma$ to Q.

Example had: $\delta(\text{even}, 0) = \text{even}$, $\delta(\text{odd}, 0) = \text{odd}$
$\delta(\text{even}, 1) = \text{odd}$ $\delta(\text{odd}, 1) = \text{even}$

$$\delta : Q \times \Sigma \rightarrow Q$$

Typical argument: $\delta(p, c) = q$ — p, q: states, c: char

Graphically: $p \xrightarrow{c} q$ — $q = p$ allowed, then it's a "loop"

View as an instruction $\underline{(p, c, q)}$.

Example: $\delta = \{(\text{even}, 0, \text{even}), (\text{odd}, 0, \text{odd}), (\text{even}, 1, \text{odd}), (\text{odd}, 1, \text{even})\}$.

(Justified on next slide)

Benefit: Same notation is also good for NFAs to come soon.

enum State {"even", "odd"}; (

```
class DFA {
    Set<State> Q;
    State s;   //start state
    Set<State> F;
    set<char> Sigma;
    State delta( State p, char c);
};
```

Flaw: the simple green form would force all DFAs to use the same method!

Better (IMHO) → set<Triple> delta: where Triple is a type for $((Q \times \Sigma) \times Q)$

If $\delta(p, c) = q$, write the <u>instruction</u> as $((p, c), q)$

Every function $f: A \to B$ is formally a relation $R_f \subseteq A \times B$, $R_f = \{(a, b) : b = f(a)\}$, such that $(\forall a \in A)(\exists! b \in B)(a, b) \in R_f$

Here, $A = Q \times \Sigma$, $B = Q$.

M will be a DFA if its <u>delta</u> set of instructions has this functional property.

Clearly set<Triple> delta is a class member—can be different for different DFAs.

[The Friday 8/31 lecture was a demo of the "Turing Kit"]