CSE396 Lecture Tue Feb 6 Spr 2018 ①

| 23314S |
| Text 7718 |

Ofc Hrs: KWR Wed 1-3pm Thu 1-2 Fri 1-2.
First Assgt → Thu, Some TAs will have hrs Fri aft.

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

- Q is a finite set of states
- Σ is an alphabet — that is, a finite set of chars.
- s, a member of Q, is the start state (q_0 in text)
- F, a subset of Q, is the set of (desired) final states. also called the set of accepting states.
- Greek delta δ is the transition function: $\boxed{\delta : Q \times \Sigma \to Q.}$

This definition means:

```
class DFA {
    set<State> Q;
    set<char> Sigma;
    State s;
    set<State> F;
    State delta(State p, char c);      //REQ: c is in Sigma
In C++ — State (*delta)(State p, char c);
KWR prefers: — set<Triple<State, char, State>> delta;
};
```

State is an enumeration type, or any finite set.

defines the (sub-)set of chars that the DFA uses.

This is a class method. We need a member method

ISSUE: this would define the same delta method for all DFA objects!

→ pointer to code that can be tailored to a particular M!

This makes delta a member rather than a class-wide method so clearly it depends on an instance M. Triples are instructions

Visualization: $\delta \subseteq (Q \times \Sigma) \times Q$ $p, q \in Q$, $c \in \Sigma$

Q is a set of nodes. Element: $((p, c), q)$

δ is a set of edges with labels in Sigma. Visualize: $p \xrightarrow{c} q$

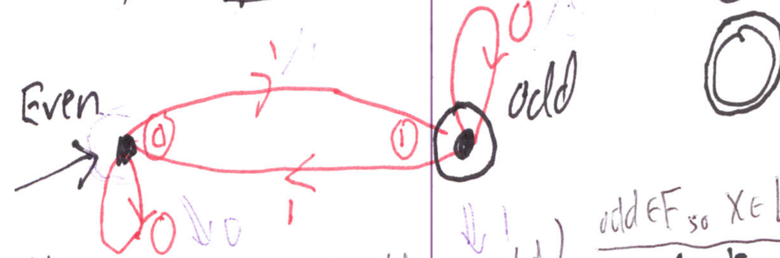
RED ≡ Unnecessary

Example: Tell whether a given string x over $\boxed{\Sigma = \{0,1\}}$ has an odd number of 1s.

$\boxed{Q = \{even, odd\}}$

A DFA M "for" parity: ← parity

$x = 11010$ Computation $n = 5$



(s, 1, odd, 1, s, 0, s, 1, odd, 0, odd)

odd ∈ F so $x \in L(M)$

Interpretation / INVariant: Current state reflects the number of 1s seen so far.

∴ Start state $\boxed{s = even}$, since initially we have seen zero 1s and 0 is an even number.

$$\boxed{\delta = \{(s,0,s), (odd,0,odd), (s,1,odd), (odd,1,s)\}}$$

We desire that x has an odd number of 1s

$\boxed{F = \{odd\}}$

$\delta(p,c)$ = switch states if $c = $ '1', leave p alone if $c = $ '0'.

In a DFA, the set δ has exactly one member $(p, c, \cdot)$ for all $p \in Q, c \in \Sigma$.

$\delta(s,0) = s$, $\delta(s,1) = odd$, $\delta(odd,0) = odd$, $\delta(odd,1) = s$.

The language $L(M)$ of this DFA M equals $\{x \in \{0,1\}^* : \#1(x) \text{ is odd}\}$.

Formally, we can define $L(M)$ for any DFA M via: "zero or more" "the number of 1's in x".

Defn: A computation by a DFA $M = (Q, \Sigma, \delta, s, F)$ is a sequence

$$\vec{C} = (q_0, x_1, q_1, x_2, \ldots, x_{n-1}, q_{n-1}, x_n, q_n) \text{ where:}$$

$n = |x|$ (the length of x)

$x = x_1 \cdots x_n$ where x_i is ith bit.

$q_0 = s$, each $q_i \in Q$, and: $\boxed{\text{For all } j, 1 \le j \le n, (q_{j-1}, x_j, q_j) \in \delta}$

$\boxed{\vec{C} \text{ is accepting if also } q_n \in F.}$

Defn: $L(M) = \{x \in \Sigma^* : M \text{ has an accepting computation on input } x\}$.

OK to define DFAs by their diagrams. Some simple cases.

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

$M_0 =$ 

$Q = \{s\}$, $F = \emptyset$

$L(M_0) = \emptyset$

$M_{all} =$ 

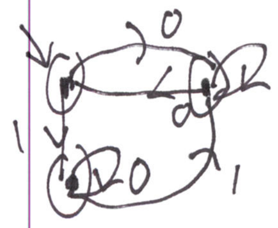
$Q = \{s\}$ $F = \{s\}$

$F = \{s\}$

$L(M_{all}) = \Sigma^* = \{0,1\}^* = \{$all strings formed by zero or more chars in sequence$\}$

Other DFAs M s.t. $L(M) = \Sigma^*$: 

$F = \{s, odd\}$



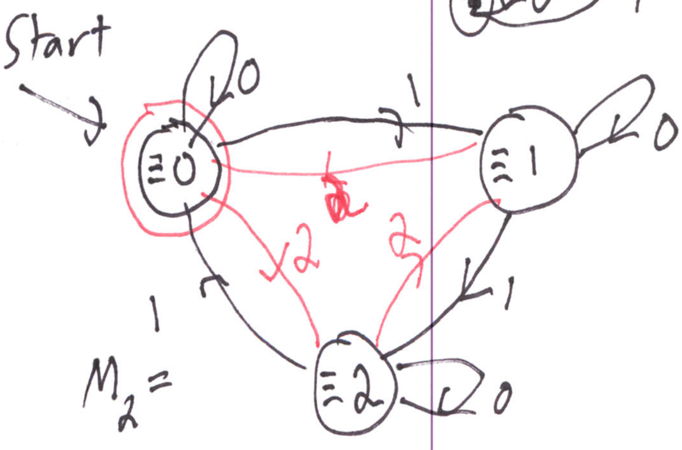
Hence a DFA need not be "in lowest terms".

One More Example:

How about $\#1(x)$ mod 3?

$\#1(\epsilon) = 0 \equiv 0 \bmod 3$

Start $\in F \iff \epsilon$ should be in L.



$M_2 =$

Read $c=0$: congruence stays same.

Read $c=1$: congruence up by 1 mod 3

Cycle clockwise.

$\Sigma = \{0,1,\underline{2}\}$

Read $c=2$: cycle counterclockwise

$F = \{s\}$ makes $L(M) = \{x \in \{0,1,2\}^*$: sum of digits is a multiple of 3$\}$.

(Lecture ended here.)

cise: DFA to

ll whether a decimal

umber is a multiple

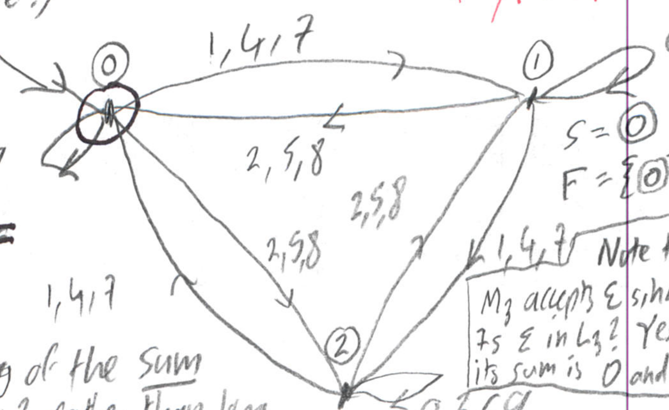
3. The language is

$3 = \{x \in DIG^*$: the

m of the digits in

belongs to $L_3\}$

$IG = \{0,1,2,3,4,5,6,7,8,9\}$

$M_3 =$ 

$s = ⓪$

$F = \{⓪\}$

Note that M_3 accepts ϵ since $s \in F$. Is ϵ in L_3? Yes since its sum is 0 and $0 \in L_3$.

The way I defined L_3 looks circular, but that's a problem only when x is a single digit, so we declare that $0,3,6,9 \in L_3$ as the basis. For other x, it's well-founded.

Indeed, M_3 is kind-of the same as M_2 but with a bigger alphabet.

Thinking of the sum modulo 3 rather than long division makes it easier to see that M_3 is correct: $L(M_3) = L_3$