1-1/ effective math $\sum_{i=0}^{20} 2^i$ vs $\sum_{i=0}^{\infty} 2^i$

$$4x^3y^2z + 8xyz - 99xy^2z^4 = 0$$

which statements are true?

"All birds have wings"

"1 + 1 = 2" vs "1 + 1 = 3"

Defining the set of true stmts

Making a decision procedure

Generating a list

A statement is a string of characters from an alphabet

$\underbrace{\text{from}}_{\text{some finite set}}$

A finite set is one where you can write down all the elements! Σ

$S = \{$ Pikachu, Charmander, Squirtle, Bulbasaur$\} = \{C, P, B, S\}$

A string of Σ is a finite sequence of Σs

P P P P CSCSCSB $\underbrace{\quad} = \epsilon$

1-3/ A language is a set of strings

$\{\varepsilon, P, PP, PPP, PPPP\}$ – finite

$\{\varepsilon, P, PP, \ldots, P^{17}, \ldots P^{256}, \ldots\}$ – infinite

$x \in S$ – x is inside S $\quad P \in \{\varepsilon, P, PP\}$

$x \in X \cup Y$ iff $x \in X$ or $x \in Y$

$x \in X \cap Y$ iff $x \in X$ and $x \in Y$

$x \in \overline{Y}$ iff $x \notin Y$ (but $x \in \mathcal{U}$ – universe)

→ complement or negation of Y

$x \circ y$ = the sequence of x, then y

$PP \circ BC = PPBC$

$x \circ y \in X \circ Y$ iff $x \in X$ and $y \in Y$

$PB \in \{P, PP\} \circ \{B, S, C\}$

$PPC \in$

---

lexicographic ordering of Σ is a sequence

$lo(\{0,1\}) = \underbrace{\varepsilon}, \underbrace{0, 1}, \underbrace{00, 10, 01, 11}, \underbrace{000, 001, 010, 011, 100, 101, 110, 111}, \ldots$

$8 - 1 = 7 - 2 = 5 - 4 = 1$

1-3/ lexi : num → string of Σ ($\Sigma = \{0,1\}$ $|\Sigma| = 2$)

lexi 0 = ε lexi 1 = 0 lexi 2 = 1

lexi 3 = 00

lexi n = sz = size of Σ

if $n < sz^0$ then ret ε of len 1

$(n - sz^0) < sz^1$ then convert $(n-sz^0)$ into a string

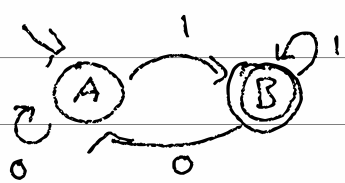
$(n-sz^0) - sz^1 < sz^2$ convert of len 2

The set of strings in the lexicographic ordering of Σ = Σ^*

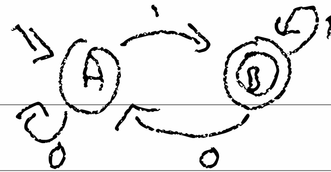
$\varepsilon \in \Sigma^*$ PPCBSPPPP $\in \Sigma^*$

$010111 \in \Sigma^*$

Deterministic Finite Automata (DFA)



2-1) DFA



◎ means Yess

○ means No

0110 = No	1 = Yes
0111 = Yes	11 = Yes
0010 = No	00 = No
1101 = Yes	
ε = No	

transistion function (edges)

$Q \times \Sigma \to Q$

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

	A	B
0	A	A
1	B	B

always finite

Q: are the states $\{A, B\}$

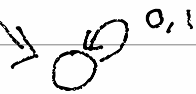
Σ: $\{0,1\}$

q_0: startastate $= A$

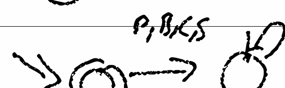
F: accepting states $\{B\}$

"n % 2 == 1" ie "odd? n"

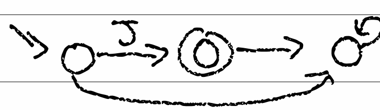
No string DFA :



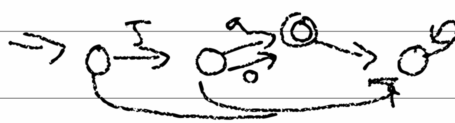
only empty string :



only the string 'J' :



'Ja' and 'Jo'



$: Q \times \Sigma \to Q$

2-2/ DFA $d = (Q, \Sigma, q_0, \delta, F)$

accepts? d s : DFA $\times$ $\Sigma^* \Rightarrow$ Bool

accepts? d ε = is q_0 in F ?

d.F.member (d.q_0)

accepts d (c :: s)

: DFA : Q : Σ^*

accepts d s = helper d d.q_0 s

helper d q_i ε = $q_i \in$ d.F

helpre d q_i (c::s) = helper d q_j s

q_j = d.δ(q_i , c)

```
DFA :: Accepts ( 304 String s) {
    Q qi = this.q0;
    while ( s != empty ) {
        qi = this.delta (qi , s.first);
        s = s.rest }
    return this.F.member (qi) }
```

✓ trace

2-3) 0110 ⟹ Even, Odd, Odd, Even

Transducers are DFAs where trace matters

Moore machines

L(d) = the language of DFA d

= { s | accepts d s = true }

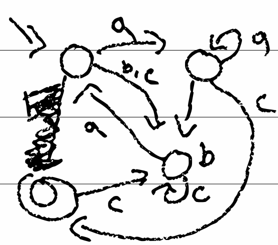
may be infinite

Given a DFA, return a string that would be accepted

example : DFA ⇒ Σ^* or false

s.t. if example d returns s then

accepts? d s = true



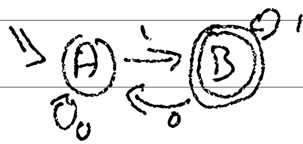
'b'

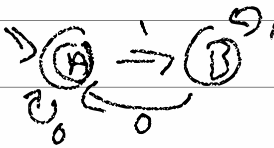
2-4/ Suppose that d is a DFA, construct d' where

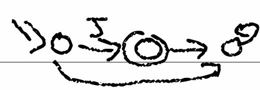
$L(d') = \overline{L(d)}$ (i.e. d' says yes when d says no & vice versa)

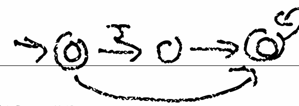
negate : DFA → DFA

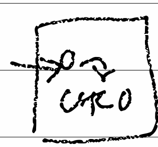
negate (Odds) = Evens

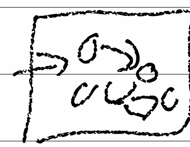
in: 

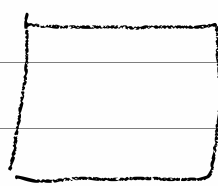
out: 

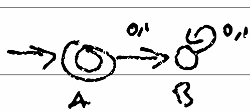
in: 

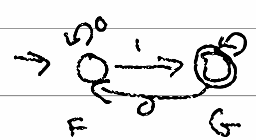
out: 

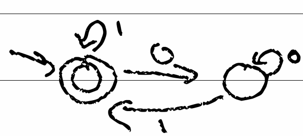
X: 

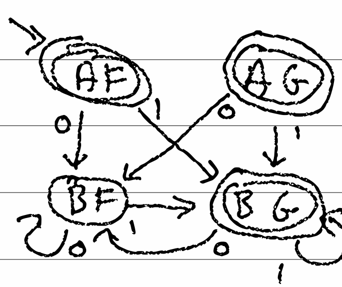
Y: 

X∪Y: 

X: 

Y: 

X∪Y: 

Z: 

2-5/ union $(X : DFA)\ (y : DFA) = (z : DFA)$

$z.Q = (x.Q \times y.Q)$

$z.\Sigma = x.\Sigma = y.\Sigma$

$z.q_0 = (x.q_0, \ y.q_0)$

$z.F = \{ (q_x, q_y) \mid q_x \in x.F$ or $q_y \in y.F \}$

and to make intersect

$z.\delta((q_x, q_y), c)$

$= (x.\delta(q_x, c),$
$\quad y.\delta(q_y, c))$

$X \subseteq Y$ (subset) iff

$\forall q \in X.\ q \in Y.$

$X = Y$ iff

$X \subseteq Y$

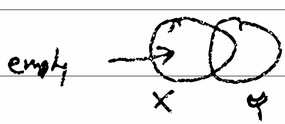
and $Y \subseteq X$

Subset? : DFA $\times$ DFA $\to$ bool

Subset? $(\emptyset)$ X = Yes

(epsilon) (Evens) = Yes

(epsilon) (odd) = No



X − Y must be empty

$X \cap \overline{Y}$ if empty