

1-1/

How do we know if a math formula  
is true?

How do we know if an algorithm  
(like Euclid's GCD) "works"?

↖ ↘

correct effective

Does an algorithm exist?

What is an algorithm?

Does a program exist? ← problems

What is a program? ← models

I-2] A set is "a bunch of stuff"

$\emptyset$  - nothin' in it  
 $\forall x, x \notin \emptyset$

$\{ \text{pen}, \text{phone} \}$      $\{ \text{phone}, \text{pen} \}$   
 $\{\checkmark, \square\}$

$\nexists \text{ pen} \in \{ \text{pen}, \text{phone} \}$

$\forall x, x \in \{y\}$  iff  $x = y$

union -  $\cup$

$\forall x, x \in A \cup B$  iff  $x \in A$  or  $x \in B$

$\{ \text{pen}, \text{phone} \} = \{ \text{pen} \} \cup \{ \text{phone} \}$

[=3] "The set of all true math formulas"

A set IS its membership

" $1+1=2$ "  $\in TS \uparrow ?$

"Is there a god?"

"Will Buffy be remade?"

All sets "constructed" via  $\emptyset$ ,  $\{\}$ ,  $\cup$  are finite.

$$x \in \{\emptyset\} \cup \{\{\emptyset\}\}$$

The Universe ( $U$ )

$A \subseteq B$  iff  $\forall x, x \in A \rightarrow x \in B$

↳ Our universe is made of strings  
and strings are sequences of characters  
and chars are elements of an alphabet  
an alphabet is a finite set



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

↑  
chars

$$\{0, 1, \cup, \$, +\}$$

↑  
chars

"0100001" = a string = s

length = 7

$$s(0) = 0$$

$$s(1) = 1 \quad s(2) = 0$$

U =  $\Sigma^*$  ← special notation

$$A^* = \{\epsilon\} \cup A \circ A^*$$

epsilon = " " = the string w/ no characters

$x \in A \circ B$  iff  $x(0) \in A$  and  
 $x(1..) \in B$

$$\{0, 1\} \circ \{0, 1\} = \{00, 01, 10, 11\}$$

$$\{1\} \circ \{0\} = \{10\}$$

LS / #1. Decide a data type to represent alphabets and characters.

Alphabet = List < Character >

Character = Object / void\*  
we need equality

#2. Decide a data type for strings

interface String { }

class MtString implements String { .. }

class OneString impl String { }

OneString ( char c, String s ) { ... }

Zero = new BasicChar('0'); One = new BC('1');

010 = new OneS(Zero, new OneS(One, new  
OneS(Zero, new MtS())));

13

1-6 Every alphabet has a lexicographical ordering of the strings in  $\Sigma^*$

$\Sigma = \{0, 1\}$        $\Sigma^0 = \{0, 1\}$   
 $\Sigma^1 = \{00, 1\}$   
 $\Sigma^2 = \{000, 10, 11, 100\}$   
 $\Sigma^3 = \{0000, 100, 110, 101, 111, 1000, 1100, 1010, 1110\}$   
 new list elements  
 $(\text{Zero}, \text{One});$

$|A|^i$  where  $i \in \text{layer } \mathbb{T}^m$  looking at  $\underbrace{\_ \_ \_}_{2^3}$

$$\text{lexi} : \Sigma \times N \rightarrow \Sigma^*$$

$$\text{lex}_i \circ \beta = e$$

$$\det(\mathbf{B} - \lambda\mathbf{I}) = 0$$

$$\log B_z = 1$$

$$\text{lex: } B \quad 6 = 191$$

2-1 "1+1"  $\rightarrow$  "2"

"1+1 = 2"  $\in$  Truth

"1+1 = 3"  $\notin$  Truth

$\emptyset \quad \Sigma^3 \quad A \cup B$

Alphabet  $\Sigma$  Universe  $\Sigma^*$

{0, 1}

{ε, 0110, 000001,



3

$P(A) \quad 2^A$

$x \in P(A)$  iff  $x \subseteq A$  ( $x \subseteq A$ , iff  
 $\forall y \in x, y \in A$ )

$A = \{0, 1, 2, 3\}$

$\emptyset \in P(A) \quad \emptyset \subseteq A$

0110 {1, 2}

{0}  $\in$   $\emptyset \subseteq \{2, 3\} \in \{0, 1, 2, 3\} \quad \underline{\textcircled{1}} \quad 123$

$P(\Sigma^*) \quad \Sigma^* = \{ \epsilon, 0, 1, 00, 111111 \}$

$\emptyset \in P(\Sigma^*)$

...

{ε}  $\in P(\Sigma^*)$

0011, ...

all even length strings  $\in P(\Sigma^*) = \{ \epsilon, 00, 11, 01, \dots \}$

GIFS  $\in P(\Sigma^*)$

{GIFS of me}  $\in P(\Sigma^*)$

JPGs w/ a cat in them  $\in P(\Sigma^*)$

2-2  $\text{ALL} = \mathcal{P}(\Sigma^*)$

$\text{FIN} =$  the set of  
finite sets

ALL	- True math
	- G-ATs
	- Even strings
$\overline{\text{FIN}}$	

$\emptyset \in \text{FIN}$

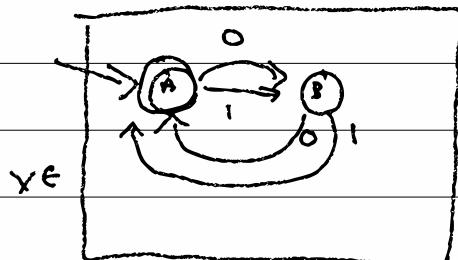
$\forall x \in \Sigma^*, \{x\} \in \text{FIN}$

$A \in \text{FIN} \wedge B \in \text{FIN}$

$\Rightarrow A \cup B \in \text{FIN}$

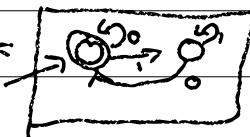
All even strings =

DFA - a deterministic finite automata



$\Sigma$  or  $\Sigma^*$

even numbers =



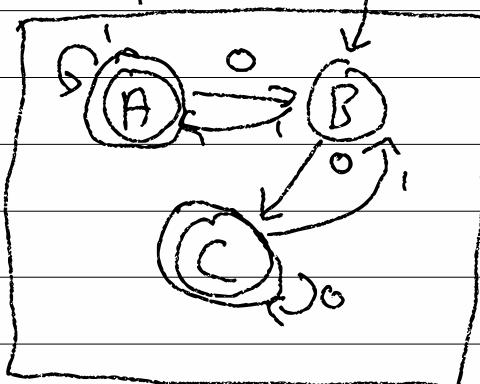
○ - states  $\Sigma A, B \}$

→ ○ - start state  $A$

○○ - accepting states  $\Sigma A \}$

$\overset{x}{\rightarrow} \circ$  - transition  $A \xrightarrow{x} B$

$x$  - labels are  $\Sigma$



$\Sigma A, B, C \}$

$\Sigma A, C \}$

$\Sigma = \Sigma_0, 1 \}$

A	0	1
B	C	A
C	C	B

2-3)  $x \in \text{DFA} (\underbrace{\mathbb{Q}, \Sigma, g_0 \in \mathbb{Q}, F \subseteq \mathbb{Q}}_{\text{states, alphabet, start, accepting}}, \delta: \mathbb{Q} \times \Sigma \rightarrow \mathbb{Q} - \text{transitions})$

DFA configuration =  $\mathbb{Q} \times \Sigma^*$   
 $\stackrel{\uparrow}{[g]} w^{\uparrow}$

config update function : config  $\times$  DFA  $\rightarrow$  config  
 $[g]w \rightarrow [g']w'$

$[g_i]x \rightarrow [g_j]y$  iff  $\delta(g_i, x) = g_j$   
 $x \in \text{DFA}$  iff  $[g_0]x \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow [g_f] \in$   
 and  $g_f \in F$

0110  $\in$  EvenLen ; iff  $[A]0110 \rightarrow [B]110 \rightarrow [A]10$   
 $\rightarrow [B]0 \rightarrow [A] \in A \cap F$  ✓

class DFA  $\Sigma$

..  $\mathbb{Q}, \Sigma, F, g_0, \delta \dots$

public bool accepts (String x) {

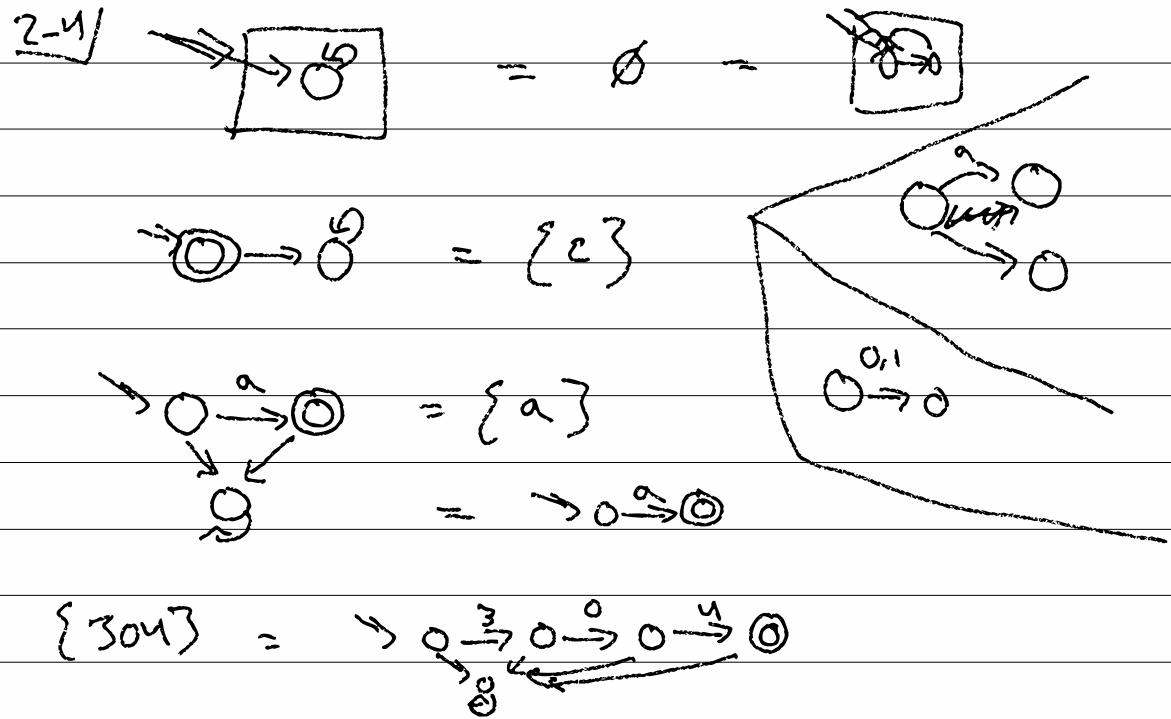
State  $g_i = g_0;$

while ( $(x, \cancel{\neq} \text{empty}) \Sigma$

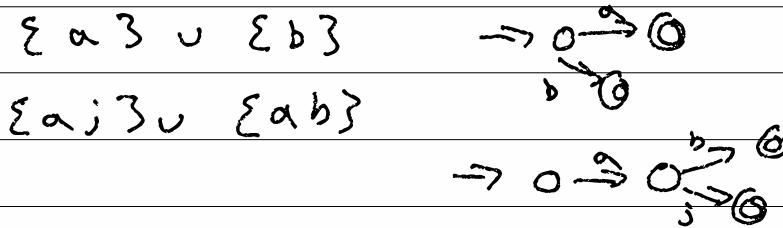
$g_i = \delta(g_i, x, \text{first}());$

$x = x, \text{rest}();$  } }

return  $f, \text{in}(g_i);$  } }



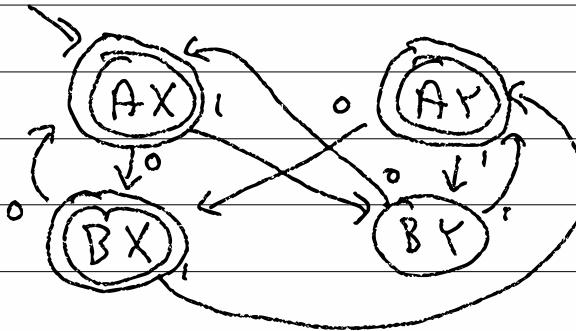
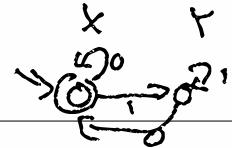
$A \cup B \leftarrow \text{DFA}$  (if  $A \in \text{DFA}$  and  $B \in \text{DFA}$ )



2-5) Even Len



Is Even



ε	✓
00	✓
11	✓
0	✓
110	✓

n

$(x, y) \in A \times B$

; if  $x \in A \wedge y \in B$

$$A = (Q_A, \Sigma, g_{0A}, \delta_A, F_A)$$

$$B = (Q_B, \Sigma, g_{0B}, \delta_B, F_B)$$

$$X = A \cup B$$

$$Q_X = Q_A \times Q_B \quad \delta_X = ((g_A, g_B), c) =$$

$$g_{0X} = (g_{0A}, g_{0B}) \quad ( \delta_A(g_A, c),$$

$$F_X = F_A \times F_B - n \quad \delta_B(g_B, c) )$$

$$F_A \times Q_B \cup Q_A \times F_B - V$$

$x \in A \cap B$  ; if  $x \in A \wedge x \in B$

2-6)  $x + A^c$  iff  $x \notin A \quad (x \in u)$

Even Len      odd Len  
 $\rightarrow \textcircled{0} \rightarrow \textcircled{0}$        $\Rightarrow \rightarrow \textcircled{0} \rightarrow \textcircled{0}$

$$F = \{A\}$$

complement

$$F' = Q - F$$

or  $F^c$  (wrt Q)

Algorithm for  $X \subseteq Y$  if  $X, Y$  are DFAs

### 3-11 DFA $\Rightarrow$ example or false

DFA:

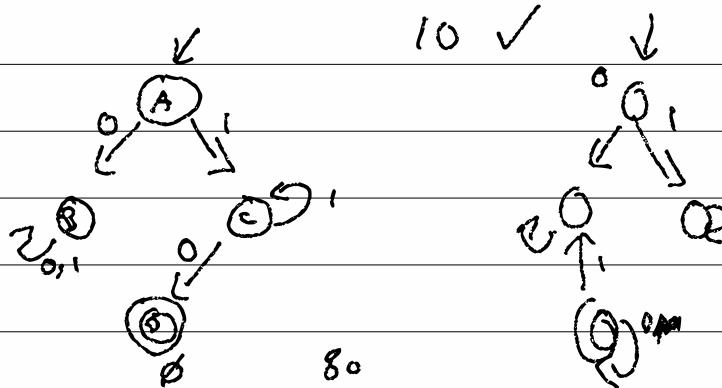
$Q$ : Sstate  $\Rightarrow$  Bool

$\Sigma$ : list of characters

$s_0$ : state

$S$ : (state  $\times$  char)  $\rightarrow$  Sstate

$F$ : State  $\Rightarrow$  Bool



$\Sigma = \{A, B, C, D\}$

$\Sigma = \{A\}$

$[A]$

$A \Rightarrow \text{True}$

$\Sigma = \{B, C, D\}$

$\Sigma = \{A\}$

$[B, C]$

$B \Rightarrow A, 0$   
 $C \Rightarrow A, 1$

$\Sigma = \{C, D\}$

$\Sigma = \{A, B\}$

$[C]$

'Yes, it is possible.'

$\Sigma = \{D\}$

$\Sigma = \{A, B, C\}$

$[D]$

$D \Rightarrow C, 0$

$\Sigma = \{E\}$

$\Sigma = \{A, B, C\}$

$[E]$

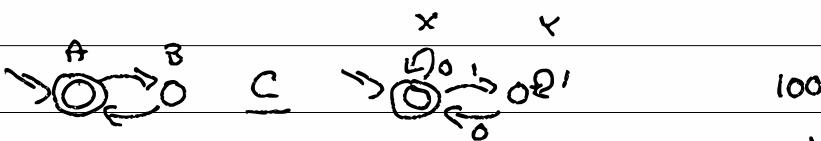
or, if not, No!

3-2/ subset

$A \subseteq B \iff \forall x \in A. x \in B \rightarrow x \in B$

$$\{\alpha, \beta\} \subseteq \{\alpha, \beta, \gamma\} \quad U = \{\alpha, \beta, \gamma\}$$

finite means naive works!



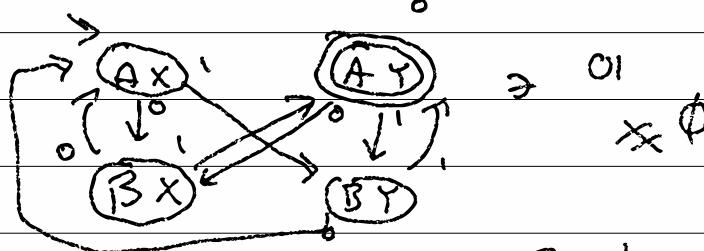
$$\boxed{\bar{A}} \subseteq \boxed{\bar{B}} \cap \boxed{\bar{A}} = \boxed{\bar{A}}$$

$$\boxed{B} \cap \boxed{\emptyset} = \boxed{\text{shaded}} \emptyset$$

Diagram illustrating set intersection and empty set:

- Set  $\bar{A}$  is shown intersecting with set  $\bar{B}$ , resulting in set  $\bar{A}$ .
- Set  $B$  is shown intersecting with the empty set, resulting in the empty set.

$$\mathbb{B} \text{ EvenNum} = \rightarrow \xrightarrow{x^0} \xrightarrow{x^1} \xrightarrow{y^0} \xrightarrow{y^1} \text{01}$$



soundness: model  $\subseteq$  theory  
 completeness: theory  $\subseteq$  model  
 $\text{model} = \text{theory}$

$$3-3) \quad 0, 1, 2, -1, 5 \quad \mathbb{Z}, \mathbb{P}, \mathbb{N}$$

$$\{\mathbb{P}\} + \{\mathbb{N}\} = \{\mathbb{P}, \mathbb{Z}, \mathbb{N}\}$$

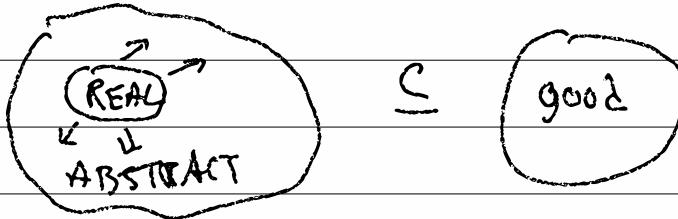
if  $x > 0$  then

$$A \quad y = 5 \Rightarrow \{\mathbb{P}\}$$

o.w.

$$B \quad y = 0 \Rightarrow \{\mathbb{Z}\}$$

$\Rightarrow$  assume  $y = \{\mathbb{P}, \mathbb{Z}\}$



<u>3-w)</u>	Finite	=	$\emptyset$	$\Sigma^3$	$A \cup B$	EDFA
			$A^c$	$A \cap B$	$A \circ B$	

Infinite =  $A^*$

\*  $x \in \Sigma^* \wedge y \in \Sigma^*$  then  $xoy \in A \circ B$  iff  
 $x \in A \wedge y \in B$

$$\varepsilon \circ y = y \quad \text{if } a \in \Sigma, (a \circ x) \circ y = a \circ (xoy)$$

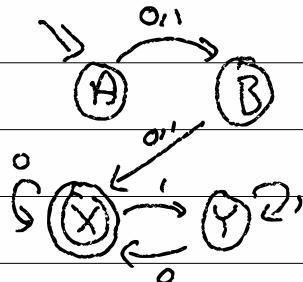
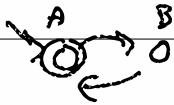
$$abcd = ab \circ cd$$

$$\{\text{jm}\} \circ \{\text{mj, nj}\} = \{\text{jim, jn}\}$$

$x \in A^*$  iff  $x = x_0 \circ x_1 \circ \dots \circ x_n$  for  $n \in N$   
and  $x_i \in A$

$$\{\text{jm}\}^* \ni \varepsilon, \text{ jm, jmjmjmjmjm}$$

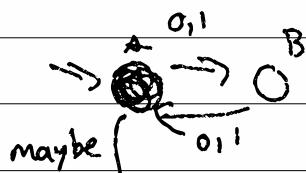
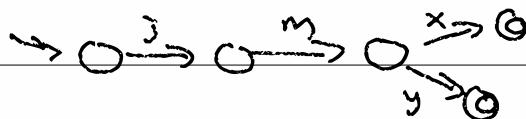
3-5/ Even Len    o Even Num



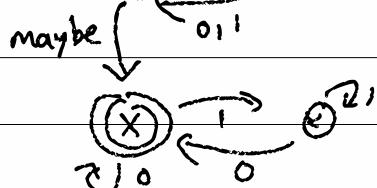
00110 ✓  
0011 x

~~00110011~~

$\{ \text{im } 3 \} \circ \{ x, y \}$

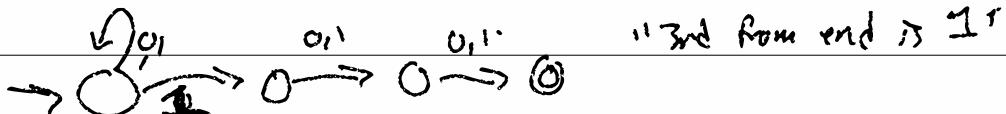


$\Sigma = \{ 0, 1 \}$



$x \in \text{DFA}$  iff

There is some path  
from  $q_0$  to  $q_f \in F$   
labelled w/  $x$



"3rd from end is 1"

3.6) NFA = non-deterministic  
finite automata

old world: the next step was obvious

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

new world: crazy options

- do you even read achar?
- which path do you take?

$$\delta': Q \times \{\text{maybe}\} \cup \Sigma \rightarrow P(Q)$$

$$\delta'(A, r) = \{A, B\}$$

$$\delta'(A, \text{maybe}) = \{C\}$$

epsilon

$$\epsilon \in \Sigma$$

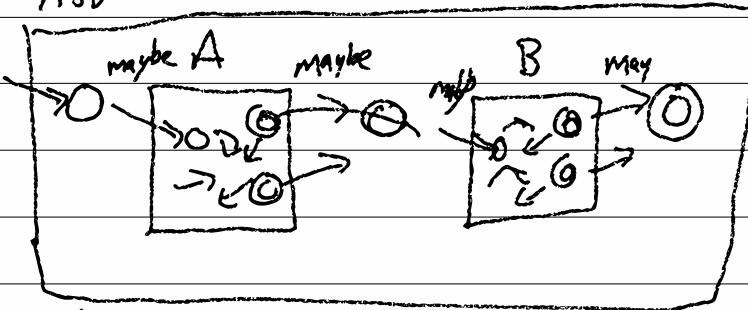
4-11  $A \circ B \in \text{DFA}$  iff  $A \in \text{DFA}$   
 $A^*$   $\wedge B \in \text{DFA}$

NFA ( $N - \underline{\text{non}} \text{ D-deterministic}$ )

$$S: Q \times (\Sigma \cup \{\text{maybe}\}) \rightarrow P(Q)$$

$$S: Q \times \Sigma \rightarrow Q$$

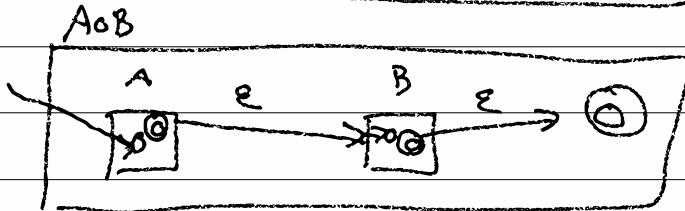
$A \circ B$



maybe written  
as "ε"

what does (NFA)

this mean?



$\text{NFA} \leftrightarrow \text{DFA}$

## 4-2] what do NFAs mean?

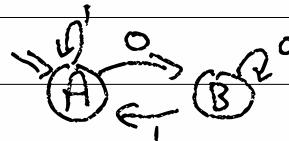
A DFA represents a set and  
a set "is" a membership function

$$U \rightarrow \{0,1\}$$

$$\subseteq \Sigma^* \rightarrow \{\text{Y}, N\}$$

$$\text{config} = \Sigma^* \times Q$$

$$\Sigma^* \rightarrow Q^*$$



$$0110 \rightarrow \underline{ABAAB} \rightarrow \text{a trace}$$

$$\Sigma^* \rightarrow (\underline{Q}, \delta)^*$$

$$0110 \rightarrow \underbrace{(0, B)(1, A)(1, A)(0, B)}_{\text{a trace}} = \Sigma^* \cup \Sigma \epsilon^3$$

$$0A1A1A0B \rightarrow N$$

$$\text{valid? } : \delta(\boxed{\Sigma}, Q)^* \rightarrow \{\text{Y}, N\}$$

$$\text{valid } g; \epsilon = Y$$

$$\text{valid } g; (c, g_j) : \text{more} = \text{if } \delta(g_j, c) = \boxed{g_j}$$

$$\text{Nvalid? } : Q \times (\boxed{\Sigma} \times Q)^* \rightarrow B$$

$$\text{valid } g; \text{ more}$$

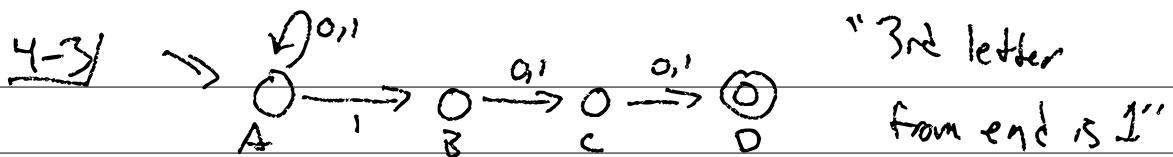
$$\text{Nvalid } g; \epsilon = Y$$

$$\text{o.w. } N$$

$$\text{Nvalid } g; (c, g_j) : \text{more} =$$

$$\text{if } \boxed{g_j} \boxed{e} \boxed{\delta(g_j, c)} \text{ then Ag Oracle}$$

$$\frac{\text{Nvalid } g; \text{ more}}{O.W. N}$$



0 1 00

1 1 1

1 1 0 1 0 0

- Y

0 0 0

1 0 0 0

1 0 1 1

- N

(0, A)(1, A)(0, A)(0, A) ✓

str( $\Sigma \times Q)^* = \Sigma^*$

(0, A)(1, B)(0, C)(0, D) ✓

str  $\epsilon = \epsilon$

(0, B)(1, C)(1, D)(0, D) X

str ( $C, \perp$ ): move  $\in$

$\delta(A, 0) = \Sigma A \}$

$\delta(D, 0) = \emptyset$

$C \circ$  str more

accepts :  $\Sigma^* \rightarrow Y/N$

accepts  $w = Y$  iff  $\exists t \in \text{traces. } \text{str}(t) = w$ .

valid  $\Leftrightarrow t \in F$

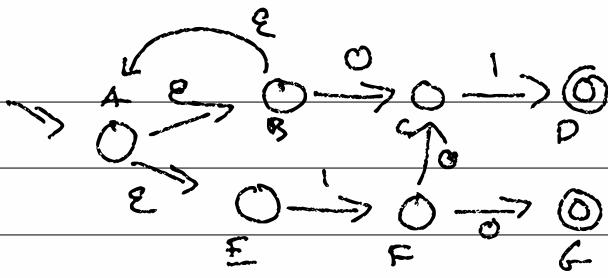
and last-state( $t$ )  $\in F$

NFA-accepts :  $\Sigma^* \rightarrow Y/N$

figure all possible traces

check if valid and if strings match

check if past is in  $\epsilon F$



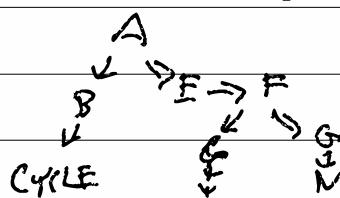
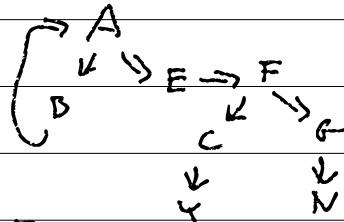
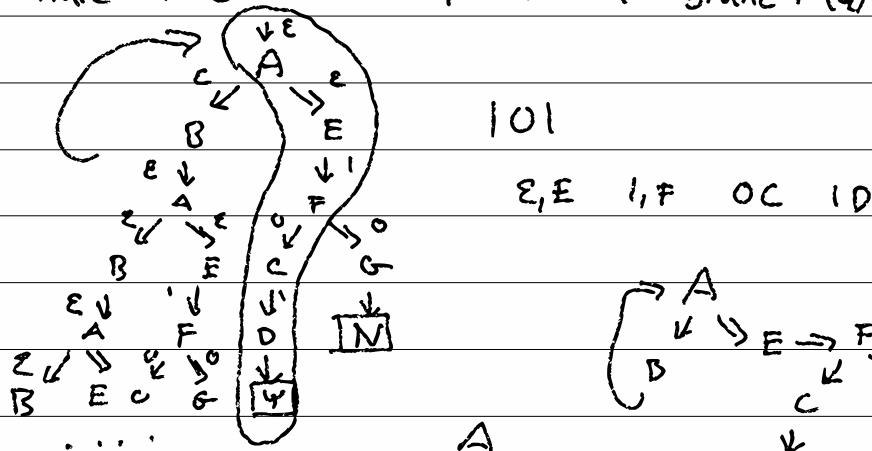
$(\epsilon, B)(0, C)(1, D) \quad 01 = 0001$

$(\epsilon, E)(1, F)(0, G) \quad 101 = 01001$

$(\epsilon, E)(1, F)(0, G) \quad 10 = 0100$

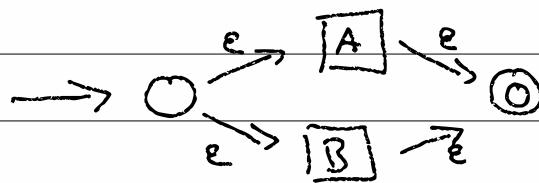
$(\epsilon, B)(\epsilon, A) \times \text{ where } \times \text{ is valid}$   
 $\rightarrow \text{valid}$

Trace Tree = T | N | Branch ( $\epsilon, Q$ ) (List TTI)



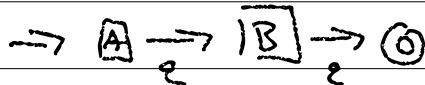
Forking model of NFAs (make TT)  
 Back-tracing model (explores TT)

4-5)  $A \cup B$



$x \in A$   
 $\square \rightarrow \square$   
State  $X$  transitions

$A \circ B$



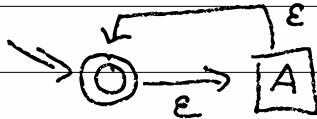
state of  $A$

$\square \rightarrow \square$

All accepting states

of  $A$  transition to  $\square$

$A^*$

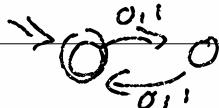


4-6)   $\forall A$ ,  $A \in \text{DFA} \Leftrightarrow A \in \text{NFA}$

$\Rightarrow$

$\Leftarrow$

DFA  $\Rightarrow$  NFA



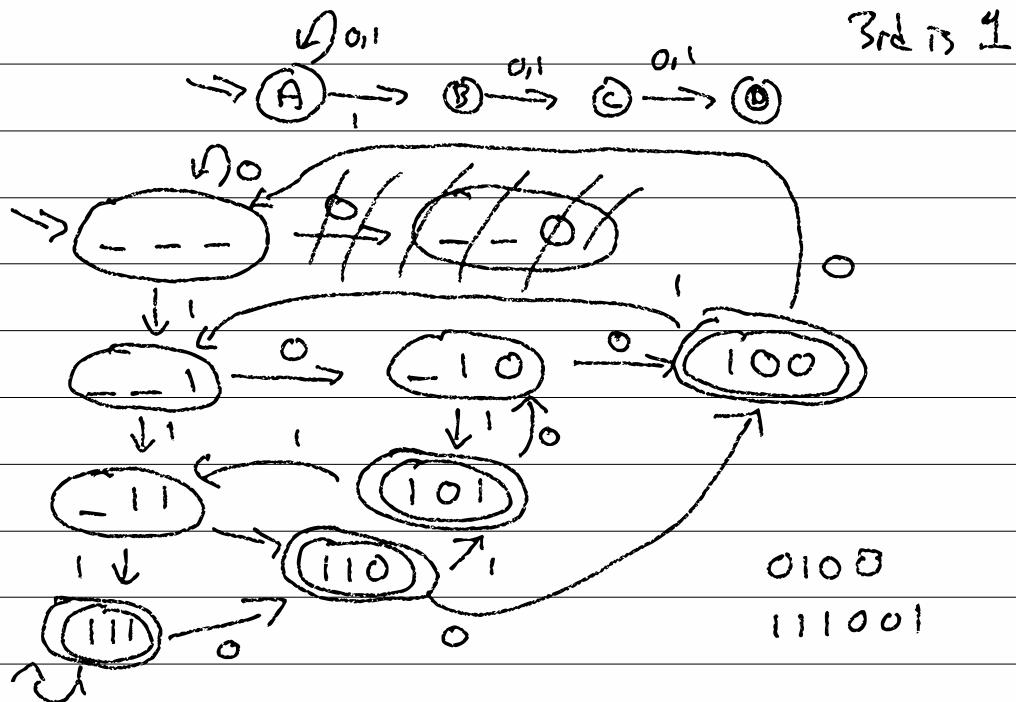
DFA  $\delta: Q \times \Sigma \rightarrow Q$

NFA  $\delta': Q \times \Sigma_c \rightarrow P(Q)$

$$\delta'(q_i, \epsilon) = \emptyset$$

$$\delta'(q_i, c \in \Sigma) = \{\delta(q_i, c)\}$$

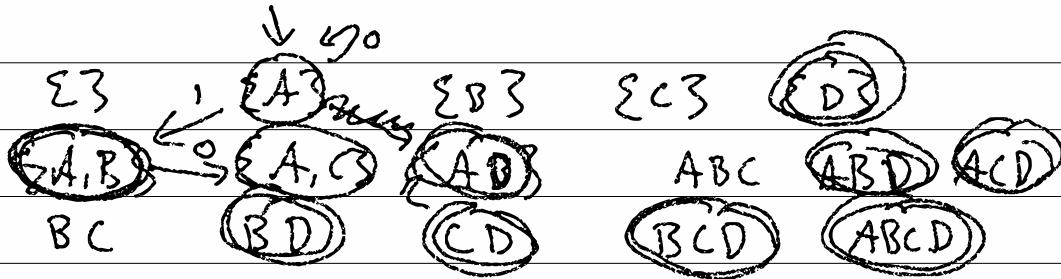
NFA  $\Rightarrow$  DFA



4-7)  $\text{NFA} = (\mathbb{Q}, \Sigma, q_0, \delta; \mathbb{P}(\mathbb{Q}) \xrightarrow{\Sigma} \mathbb{P}(\mathbb{Q}))$

$\text{DFA}^{\text{out}} = (\mathbb{Q}', \Sigma, q'_0, \delta': \mathbb{Q}' \times \Sigma \rightarrow \mathbb{Q}', F' \subseteq \mathbb{Q}')$

$$\mathbb{Q}' = \mathbb{P}(\mathbb{Q})$$



$$q'_0 = \Sigma^{q_0}$$

$F'$  = any state where  $NF \neq \emptyset$

$$\begin{aligned} \delta'(\Sigma^{q_1}, \dots, \Sigma^{q_n}, c) &= \\ \cup \quad \delta(q_i, c) \end{aligned}$$

$$\underline{5-1} / A \cup B \quad \delta_A : Q_A \times \Sigma \rightarrow Q_A$$
$$\delta_B : Q_B \times \Sigma \rightarrow Q_B$$

$$\delta' : \overbrace{Q_A \times Q_B}^{\text{(Q}_A \times \text{Q}_B)} \times \Sigma \rightarrow Q$$

$$\delta'((q_a, q_b), c) = (\delta_A(q_a, c), \delta_B(q_b, c))$$

char

$$(p, c) \Rightarrow \text{new Pair } \left( \begin{array}{l} \downarrow \\ \text{pair} < \text{State}, \text{State} \end{array} \right) \left( \begin{array}{l} \nearrow \\ \text{fst} \end{array} \right) \left( \begin{array}{l} \nearrow \\ \text{snd} \end{array} \right) \left( \begin{array}{l} \text{delta a}(p, \text{fst}, c), \\ \text{delta b}(p, \text{snd}, c) \end{array} \right);$$

## S-2/ NFA $\rightarrow$ DFA

$(Q, \Sigma, q_0 \in Q,$

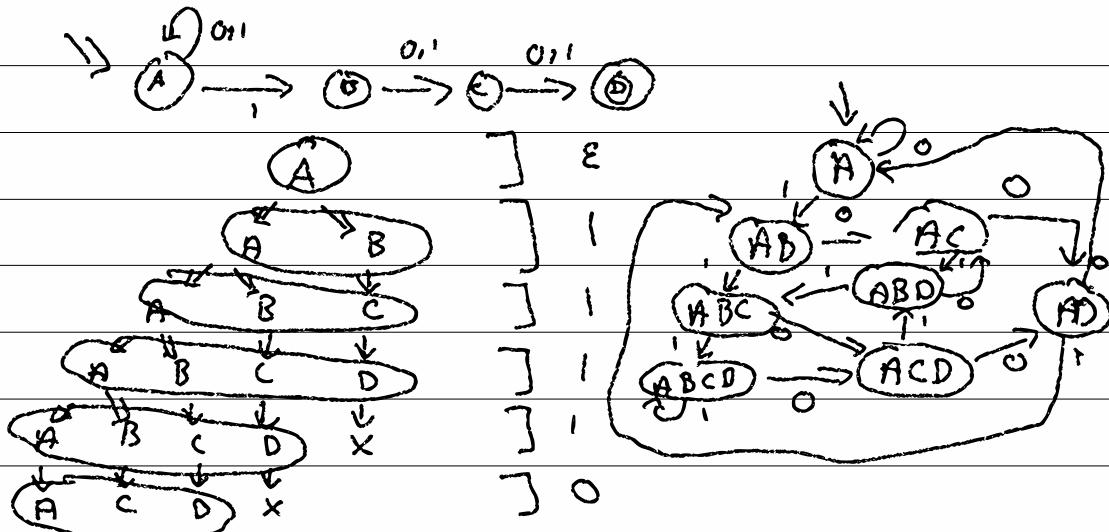
$\delta: Q \times \Sigma \rightarrow P(Q),$

$F \subseteq Q)$

$(Q', \Sigma, q'_0 \in Q'$

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

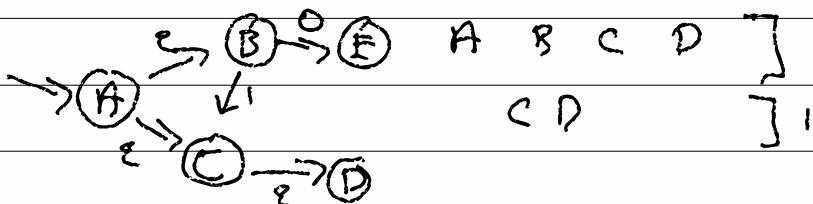
$F' \subseteq Q')$



$\begin{smallmatrix} A & B \\ C & D \end{smallmatrix} \xrightarrow{\Sigma} \{A, C\}$

$\begin{smallmatrix} A \\ B \end{smallmatrix} \xrightarrow{\Sigma} \{A\}$

$\begin{smallmatrix} A & C & C \\ DLAH & DLAH \end{smallmatrix} \xrightarrow{\Sigma} \{A\}$



$\begin{smallmatrix} A & B & C & D \\ CD \end{smallmatrix} \xrightarrow{\Sigma} \{E\}$

5-3/

$$E: Q' \rightarrow Q' \quad P(Q) \rightarrow P(Q) \quad \begin{array}{l} \text{- follow all} \\ \text{C-transitions} \end{array}$$

Trace Tree DFA

$Q'$  = things at the bottom of a tree  
set = a set of states of  
the NFA =  $P(Q)$

$g_0'$  = the top of the tree  
= the set that has only the first state  
 $= E(\emptyset, g_0) \in P(Q)$

$\delta'$  = maps the bottom of the tree to the next level  
= set of all next states of each state in the level of the tree

$$\delta'(Q_i, c) = \bigcup_{q_i \in Q_i} \delta(q_i, c)$$

$F$  = any level of tree with some accepting state  
= any set with an element in  $F$   
=  $\{Q_i \mid \underbrace{Q_i \in P(Q)}_{Q_i \in P(Q)=Q'} \text{ and } Q_i \cap F \neq \emptyset\}$

= (set-of-gs  $\rightarrow$   
for each  $g_i$  in set-of-gs  
if dfa.F.apply( $g_i$ ) then  
return true  
return false)

5-y)

$E(\text{set } \langle Q \rangle g_i)$

queue  $\langle Q \rangle$  next = ~~empty~~  $g_i$

set  $\langle Q \rangle$  seen = empty

while  $(\text{not } \langle Q \rangle \text{ is empty})$

$\delta(\text{next}, \text{first}, \varepsilon)$  add those

to next unless in seen

return seen

$E(A) = \text{least fixed point of}$

$E^*(A)$

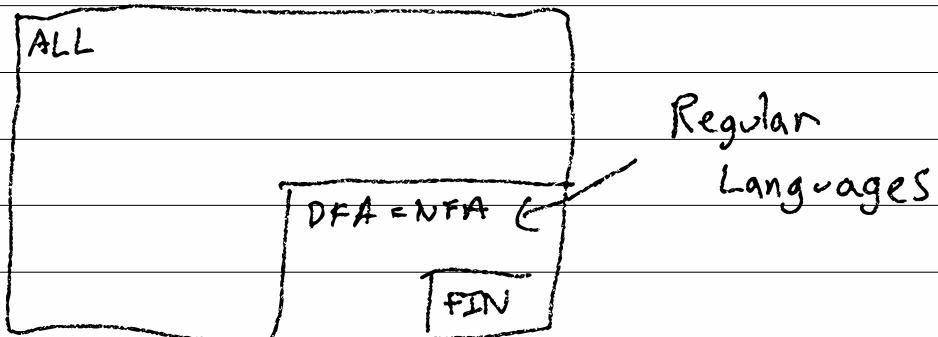
$E^*(A) = A \cup \bigcup_{g_i \in A} \delta(g_i, \varepsilon)$

6-1)  $\forall N \in \text{NFA}, \exists D \in \text{DFA}.$  compile :

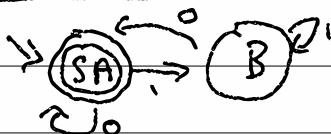
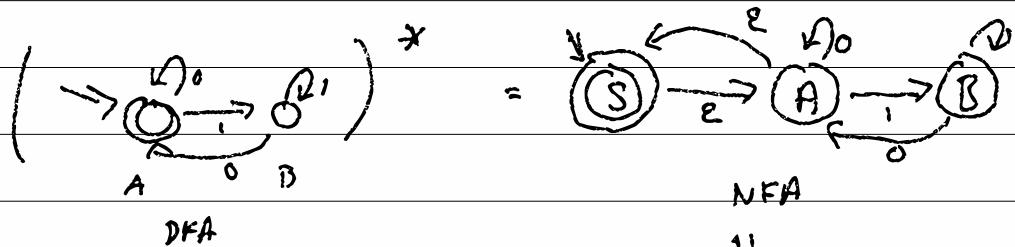
$$N = D$$

$\forall D \in \text{DFA}, \exists N \in \text{NFA}.$

$$D = N$$



Program f ...      'f; g'  
 program g ...      compositional



## 6-2 | Regular Expressions

$r ::=$	$\epsilon$	$ $	EMPTY
	$\emptyset$	$ $	NULL
	$c$	$ $	Char
$r \cup r$		$ $	CUP
$\star r$		$ $	STAR
$r \circ r$		$ $	CIRC

## interface Register { }

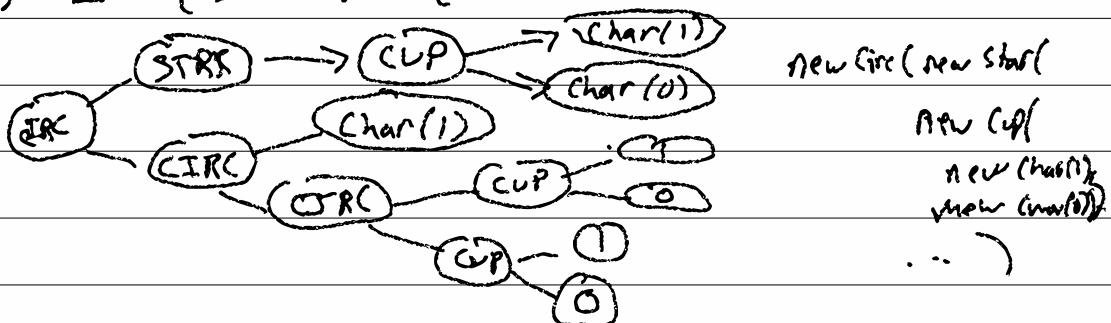
class RE\_emptv) impl REGEX { . . .

RE\_NULL()

Re-char (char c)

Re-cup (Regex lhs , Regex rhs)

$(1\cup 0)^* \circ 1 \circ (1\cup 0) \circ (1\cup 0)$  - "3rd formal B.F."



6-3 |  $L : RE \rightarrow \text{ALL} = P(\Sigma^*)$

fix the language if doesn't  
(or come)

$$L(\epsilon) = \{\epsilon\}$$

$$L(\emptyset) = \emptyset$$

$$L(c) = \{c\}$$

$$L(r \cup r') = L(r) \cup L(r')$$

$$L(r \circ r') = L(r) \circ L(r')$$

$$L(r^*) = L(r)^*$$

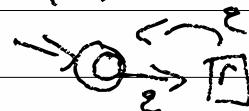
class Up {  
     $L()$   $\in$   
    return up(lhs, L(), rhs, L());  
}

compile : RE  $\rightarrow$  NFA

$$\text{compile } (\epsilon) = \xrightarrow{\epsilon} \textcircled{0}$$

$$\text{compile } (r^*) =$$

$$\text{compile } (\emptyset) = \xrightarrow{\epsilon} \textcircled{0}$$



$$\text{compile } (c) = \xrightarrow{\epsilon} \textcircled{0} \xrightarrow{c} \textcircled{0}$$

$$\text{compile } (r \cup r') = \xrightarrow{\epsilon} \textcircled{0} \xrightarrow{c} \boxed{r} \xrightarrow{\epsilon} \boxed{r'} \xrightarrow{\epsilon} \textcircled{0}$$

$$\text{compile } (r \circ r') = \xrightarrow{\epsilon} \boxed{r} \xrightarrow{\epsilon} \boxed{r'} \xrightarrow{\epsilon} \textcircled{0}$$

6-4

$$"\ " = \epsilon$$

$$\underline{\underline{\quad}} = \emptyset$$

$$"\ c" = c$$

$$"\ xyz^*" = x \circ y \circ (z^*)$$

$$"\ (xyz)^*" = (x \circ y \circ z)^*$$

$$"\ xyz" = x \circ y \circ z$$

$$"\ [abc]" = (a \cup b \cup c) \quad (a, b, c \in \Sigma)$$

$$"\ (a \mid b \mid c)" = \Rightarrow \quad (a, b, c, \in \Sigma^*)$$

[012]

(zero | one | two)

$$\bullet = \epsilon$$

$$"\ .^* \backslash. m; s" = \Sigma^* \circ ' . ' \circ ' m' \circ ' ; ' \circ ' s'$$

8-5/

gen : RE  $\rightarrow \Sigma^*$  or false

gen  $\epsilon = \epsilon$

gen  $\emptyset = \text{FALSE}$

gen  $c = 'c'$  flip coin

gen  $x \cup y = \text{gen } x \sqcup \text{gen } y$

gen  $x \circ y = \text{gen } x \circ \text{gen } y$

gen  $x^* = \boxed{\square}$

= gen  $(\epsilon \cup x \circ x^*)$   
"mjs"

equal : RE  $\times$  RE  $\rightarrow$  Bool

equal  $x \ y =$

NFA2DFA(compile  $x$ )  $\xrightarrow{\text{?}} \text{def equality?}$   
NFA2DFA(compile  $y$ )

$$(\bar{A} \cap B) \cup (A \cap \bar{B}) = \emptyset$$

G-6)

$$x + 0 = x$$

$$x = x$$

$$x \cdot 1 = x$$

$$\frac{a = b}{a+x = b+x}$$

$$\frac{a = b}{ax = bx} \quad x \neq 0$$

$$2(3x + 17) = 6x + 34 \quad \text{"algebra"}$$

$$3x + 17 = 3x + 17$$

$$3x = 3x$$

$$x = x$$

$$17 \cancel{x} \approx 17 \cancel{z}?$$

WZSS

$$\emptyset \cup x = x \cup \emptyset = x$$

$$\emptyset \circ x = x \circ \emptyset = \emptyset$$

$$\varepsilon \circ x = x \circ \varepsilon = x$$

$$\emptyset^* = \varepsilon \approx \varepsilon \cup \emptyset \circ \emptyset^* \quad x^* = \varepsilon \cup x \circ x^*$$
$$\approx \varepsilon \cup \emptyset = \varepsilon$$

$$x \circ (y \cup z) = x \circ y \cup x \circ z$$

NFA  $\approx$  DFA      in:  $N$  states      ( $Q$ )

out:  $z^N$  states      ( $P/Q$ )

6-7/ size : RE  $\Rightarrow$  Nat

$$\text{size } \emptyset = 1$$

$$\text{size } \varepsilon = 1$$

$$\text{size } c = 1$$

$$\text{size } (x \cup y) = \text{sz } x + \text{sz } y + 2$$

$$\text{size } (x \circ y) = \text{sz } x + \text{sz } y + 1$$

$$\text{size } (x^*) = \text{sz } x + 1$$

$$x \circ (y \cup z) = x \circ y \cup x \circ z$$



emacs \* .c

↓ bash

emacs .c

↓  $\Sigma^* \circ \omega C$

compile

→ o → o → o →

+

→ o → o → o

↓ N2D

Yf  $\Leftarrow$

304.c

← ↗ ↘

304.c main.c

dir

304.c

awesome-quotes.txt

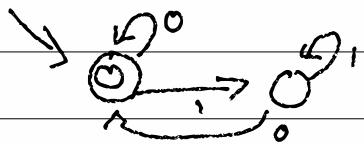
mycat.jpg

main.c

+

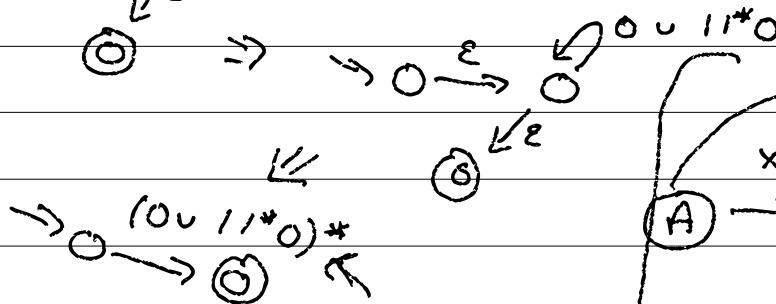
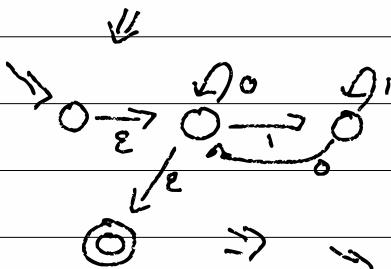
emacs

6-8] decompire : DFA  $\xrightarrow{(\text{NFA}) \text{ or }} \text{RE}$



$$\Sigma^* \circ 0 \cup \epsilon$$

$$(1 \cup 0)^* \circ 0 \cup \epsilon$$

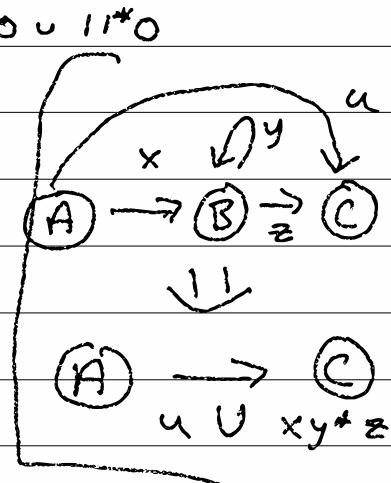


$$(0 \cup 11^* 0)^*$$

$\epsilon \quad 0 \quad 10 \quad 111110$

$011101111100$

11100011101100101010



G-9 / decompile : N-state NFA  
→ RE

START : N-NFA → (N+2)-GNFA

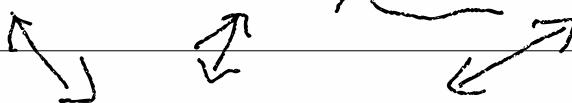
RIP : (N+1)-GNFA → N-GNFA

END : 2-GNFA → RE

decompile m = end ∘ rip<sup>n</sup> ∘ start (n)

7-1) DFA/NFAs  $\rightarrow$  RE

DFA<sub>s</sub>  $\leftrightarrow$  NFA<sub>s</sub>  $\leftrightarrow$  RE



Regular  
Languages

NFA  $\rightarrow$  RE

IN: n-NFA  $\rightarrow$  (n+2)-GNFA

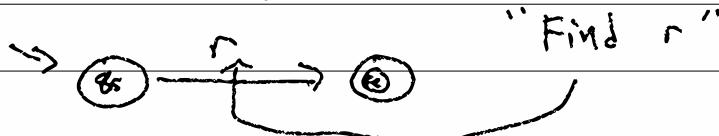
RIP<sup>n</sup>: (n+1)-GNFA  $\Rightarrow$  n-GNFA

OUT: 2-GNFA  $\rightarrow$  RE

GNFA =  $(Q, \Sigma, g_s, g_e, \Delta : (Q-g_e) \times (Q-g_s) \xrightarrow{e_Q} e_Q \rightarrow RE(\Sigma))$   
 $S: Q \times \Sigma \rightarrow Q$

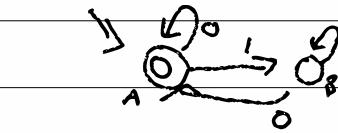
OUT: 2-GNFA  $\rightarrow$  RE

$(\{\Sigma, g_s, g_e\}, \Sigma, g_s, g_e, \{(g_s, g_e), \uparrow\})$   
 $= r = \Delta(g_s, g_e)$



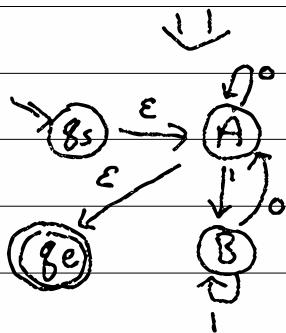
7-2) IN: NFA  $\Rightarrow$  GNFA (n+2) -  
 $(Q, \Sigma, g_0, \delta: Q \times \Sigma \rightarrow P(Q))$   $(Q', \Sigma, g_s, g_e,$   
 $F, \Delta: (Q' - g_e) \times (\Sigma - \{g_e\}) \rightarrow R_E)$

$$Q' = Q \cup \{g_e, g_s\}$$



$$\Delta(g_i, g_j) = r$$

$$\Delta(g_s, g_0) = \varepsilon$$



$$\Delta(g_s, g_j \neq g_0) = \emptyset$$

$$\Delta(g_f \in F, g_e) = \varepsilon$$

$$\Delta(g_i \notin F, g_e) = \emptyset$$

$$\Delta(g_i, g_j) = \cup \{\varepsilon_{g_{ij}}\}$$

$$\delta(g_i, c) \ni g_j \}$$

7-3/ RIP =  $(n+1)$ -GNFA  $\rightarrow$   $n$ -GNFA  
 main  $'(Q, \Sigma, g_s, g_e, \Delta)'$   $\downarrow$   $'(Q', \Sigma, g_s, g_e, \Delta')$

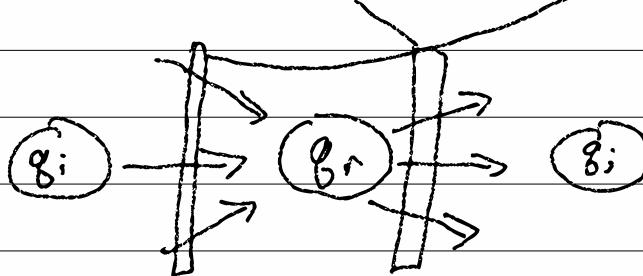
↓  
 jay  
 main

$$Q = Q' \cup \{ q \text{ gonna be killed} \}$$

↓  
 jay  
 main  
 ↓  
 exit  
 ↓  
 exit

$$\Delta' : \underbrace{(Q' - g_e)}_{g_r \in} \times \underbrace{(Q' - g_s)}_{g_r \in} \rightarrow \text{RE}$$

$$\Delta : \underbrace{(Q - g_e)}_{g_r \in} \times \underbrace{(Q - g_s)}_{g_r \in} \rightarrow \text{RE}$$

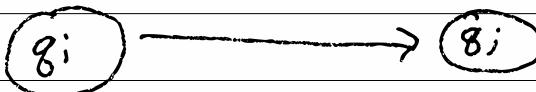


$$x \circ y^* \circ z \cup \alpha$$

$$\rightarrow x \circ z$$

$$\Delta'(q_i, q_j)$$

$$=$$

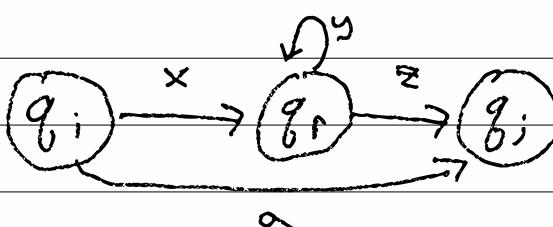


$$x = \Delta(q_i, q_r)$$

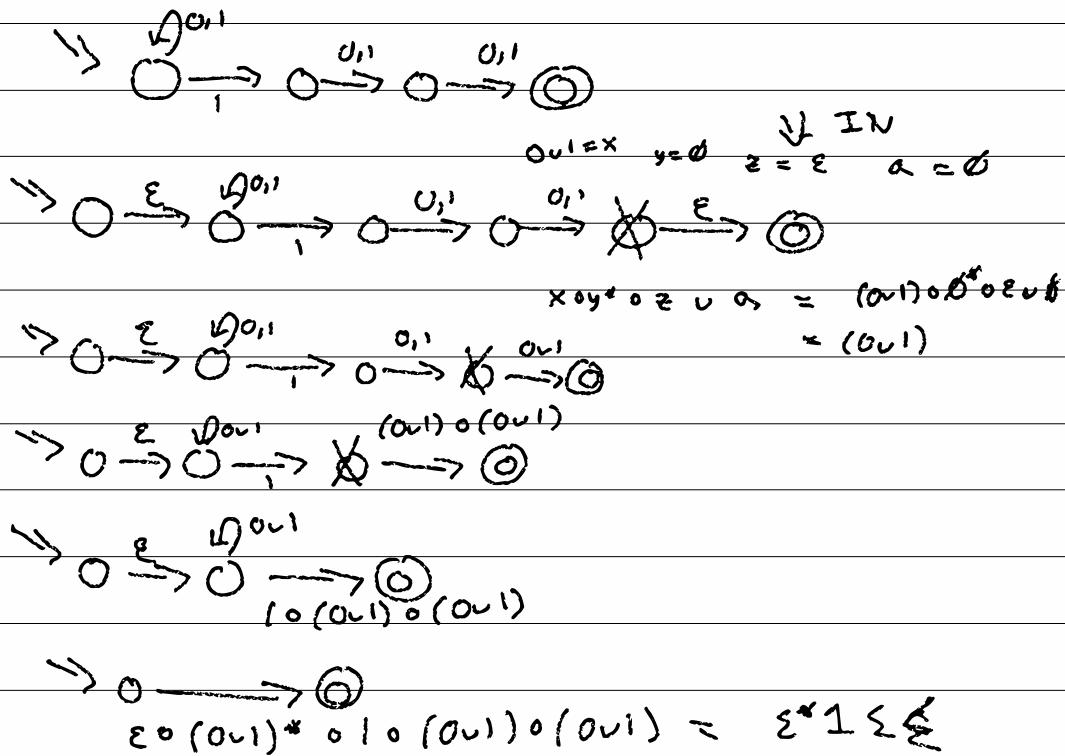
$$y = \Delta(q_r, q_r)^*$$

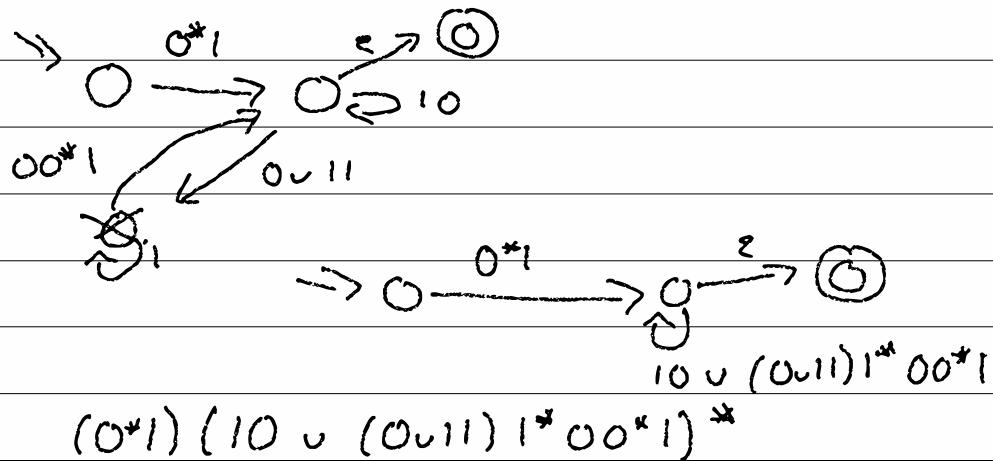
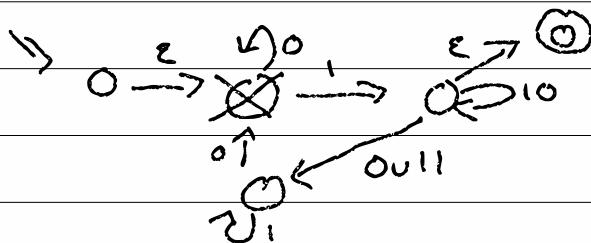
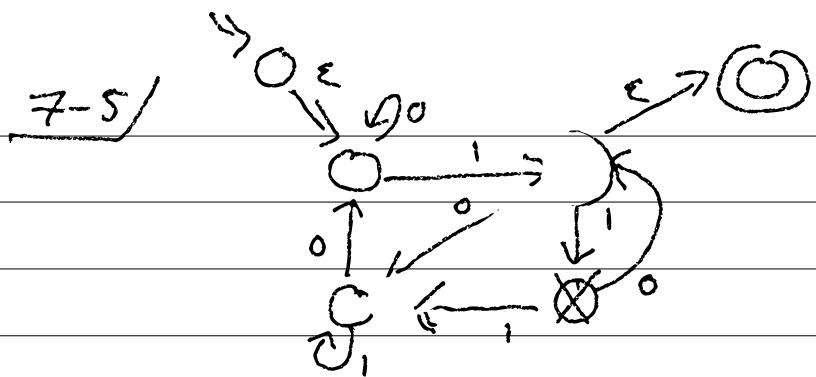
$$z = \Delta(q_r, q_j)$$

$$\alpha = \Delta(q_i, q_j)$$



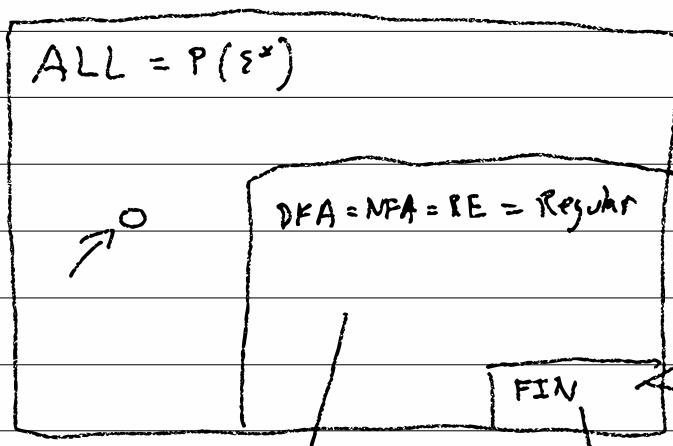
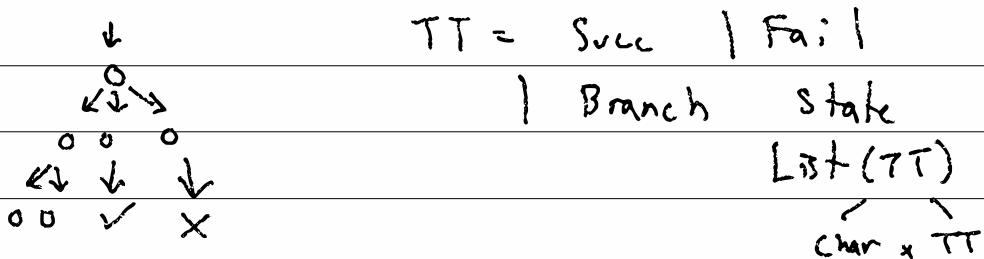
~~7-4/ int main () {~~  
 int ~~x = 8;~~  
~~int y = f(x, z);~~  
 ...  
 int y = x + 2 \* 2;  
 ↓  
 int f(int ~~z~~){  
 int ~~z~~;  
 return ~~z + 2 \* 8;}~~





8-1 ε Object → Char (char c)  
 Upsilon () Epsilon ()  
 UTF-8

TT



$\{\text{all strings ending in } 0\}$

$\{\emptyset, 00\}$

$A \in REG \iff \exists d \text{ DFA. } L(d) = A$

$\neg A \in REG \iff \neg (\exists d) \iff \forall d. d \text{ DFA} \wedge$

$\neg \exists x. P(x) \iff \forall x. \neg P(x) \quad L(d) \neq A$

$\neg \forall x. P(x) \iff \exists x. P(x)$

8-2/ How can we know stuff about  
infinite sets?

$\forall x \in A, P(x)$

$P: DFA \rightarrow \text{Prop}$

$\Rightarrow z_0 \in Q$

$P: IP(\Sigma^*) \rightarrow \text{Prop}$

$\neg P(B)$  (where  $B \in IP(\Sigma^*)$  and we "hope"  
 $B$  isn't in DFA)

Q. What is P?

1. Prove  $\nexists RFG, P(A)$

Pumping  
lemma

2. Prove  $\exists B \in \text{RELL}, \neg P(B)$

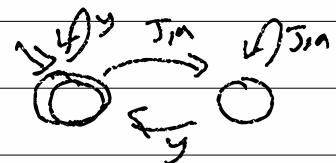
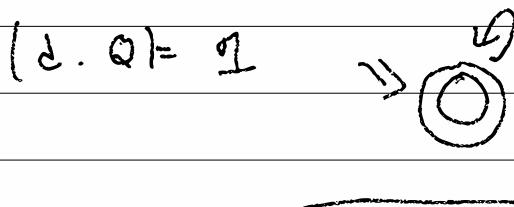
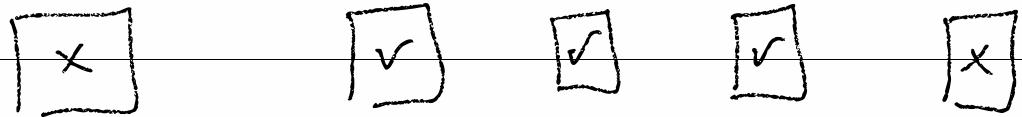
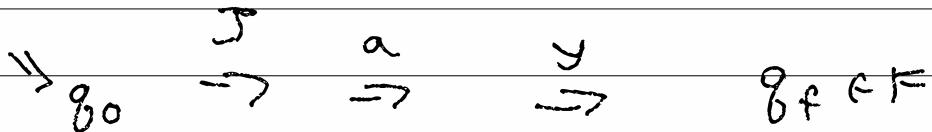
conclude  $B \notin \text{REG}$ .

$$\Sigma = \dots$$

8-3)

"Jay" & d

$\Sigma \ni \{s, a, y\}$



Daphne wins

pick a number of states ; 4

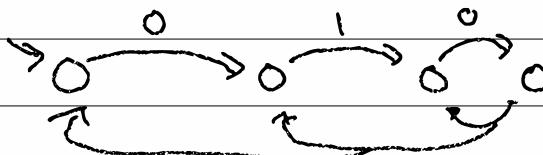
she picks a char

I say what state we go to

I win if I never say same state

she wins if I repeat

How many turns to win?

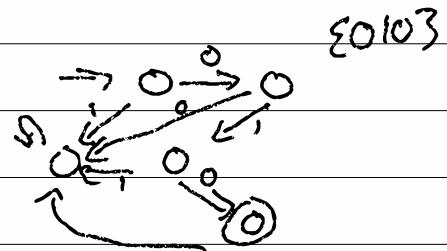
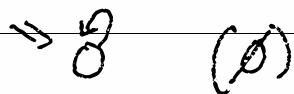
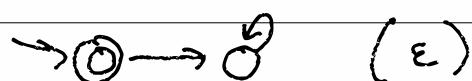


$4 \Rightarrow N \Rightarrow$

|@|

$\delta: Q \times \Sigma \rightarrow Q$  Total Fun  
 $\xrightarrow{\text{from}} \xrightarrow{\text{to}}$

All DFAs have a loop



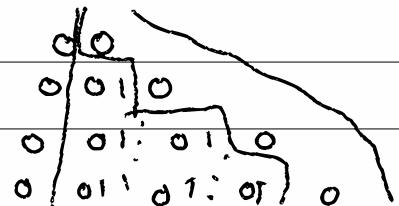
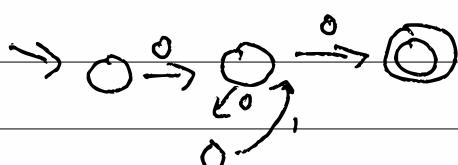
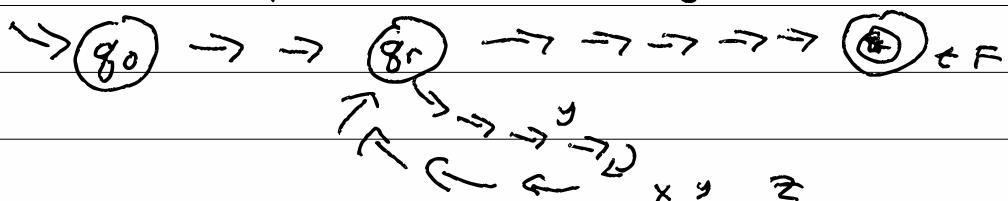
Some have "exciting" loops

$\exists s \in (\text{DFA})$ ,  $s = \dots \dots \dots \dots \dots$



$$\begin{aligned} \epsilon &= A \\ &= \epsilon \circ \epsilon \circ \epsilon \end{aligned}$$

$$0110 = \overbrace{ABA\bar{A}}^z = \epsilon \circ 01010$$



8-5/ If a machine hasn't an existing loop ...

They are all finite  
 $L(m) \in FIN$

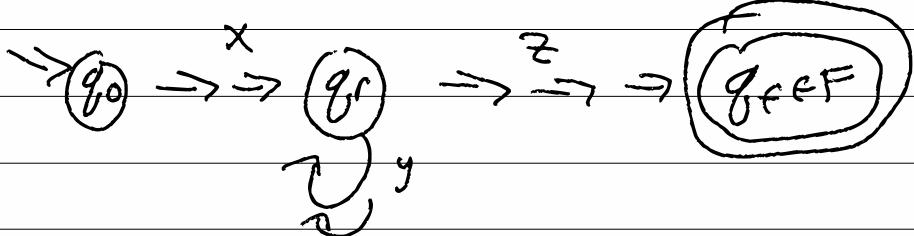
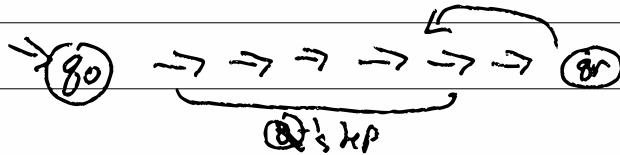
If it does have one ...

then it must be infinite (regular)

If there is an existing loop ...

what is the shortest string to  
"find" ;+ m?

$$|s| = |Q| \quad (s \in L(\delta))$$



$\in \text{ALL}(\text{a language}) (\mathcal{P}(\Sigma^*))$

8-6/ RPP(A) = Regular Pumping Property

? ( $\exists p \in \mathbb{N}, \quad / / \quad p = |Q|$

$\forall s \in A \quad | \quad |s| \geq p \quad )$

$\exists (x, y, z \in \Sigma^*) \quad | \quad s = xyz \quad \wedge$

$|xy| \leq p$

$|y| > 0 \quad )$

$\forall i \in \mathbb{N},$

$x \circ y^i \circ z \in A$

)

$\forall d \in \text{DFA}, \exists r \in \text{RE}. \quad L(d) = L(r) \quad - \text{Cof}$

$\neg \text{RPP}(B) :=$

$\forall p \in \mathbb{N},$

$\exists (s \in B \quad | \quad |s| \geq p \quad )$

$\forall (x, y, z \in \Sigma^*) \quad | \quad s = xyz \wedge |xy| \leq p \wedge |y| > 0 \quad \text{and} \quad$

$\exists i \in \mathbb{N},$

$x \circ y^i \circ z \notin B$

8-7) Need: an infinite space problem

```
O* 1 { while (getc() == '0') {  
    ungetc()  
    if (getc() == '1') { net false }  
    return getc() == EOF;  
} = 232 vint2_+
```

$\forall n \in \mathbb{N}$ ,

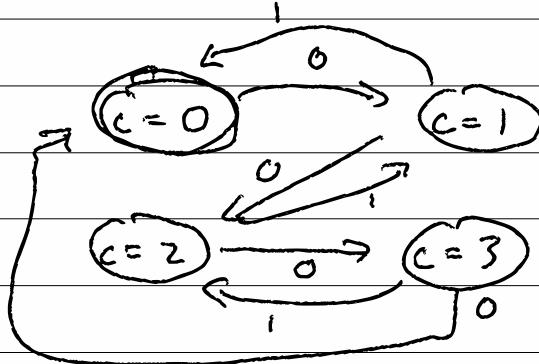
$0^n 1^n \in B$

vint count = 0;

while (in == 0) count++

while (in == 1) count--  
if (in == EOF) < count--  
return count == 0 ;

{ net false } }



q-1) RPP(A)  $\rightarrow$  Language =  $P(\Sigma^*) \approx A\bar{I}$

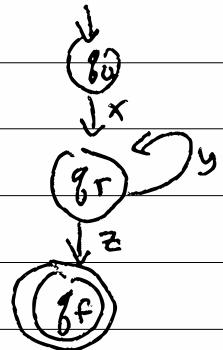
$$\exists p \in N, \quad |I|_p = |Q|$$

$$\forall s \in A \mid |s| > p \}.$$

$$\exists (x, y, z \in \Sigma^*) \mid s = xyz$$

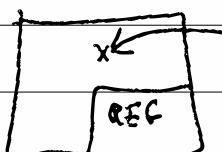
$$\forall i \in N, \quad |xy| < p$$

$$xyiz \in A \quad |y| > 0$$



$\neg RPP(A) =$

ALL



$$\forall p \in N,$$

$$\exists (s \in A \mid |s| > p)$$

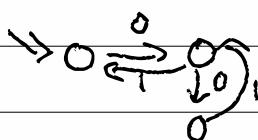
$$\forall (x, y, z \in \Sigma^*) \mid s = xyz \wedge |y| < p$$

$$\exists i \in N$$

$$xyiz \notin A$$

$$0^n 1^n \quad \text{ie} \quad x \in 0^n 1^n$$

$$\text{iff } \exists n \in N, \quad x = 0 \overbrace{0 \dots 0}^{\text{long part}} 1 \dots 1$$

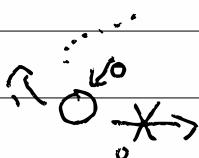


$$\text{int count} = 0$$

while (see 0)  $c++$

while (see 1)  $c--$

return  $\text{count} = 0$



9-2)  $\neg \text{RPP}(\mathcal{O}^n, \mathcal{O}^n) =$

$\forall p \in \mathbb{N},$

$\exists (s \in \mathcal{O}^n)^n \mid |s| \geq p)$

choose  $s: s = x = z$

$s = \mathcal{O}^{p/2} \cup \mathcal{O}^{p/2}$

$s = \mathcal{O}^p, |s| = 2p \geq p$

$\forall (x, y, z \in \Sigma^*) \mid s = xyz \text{ and}$

$|xy| \leq p \text{ and}$

$|y| > 0$ )

$x = \mathcal{O}^a \quad a+b+c = p \quad a+b < p$

$y = \mathcal{O}^b \quad d = p \quad b > 0$

$z = \mathcal{O}^c \cup d$

$\exists i \in \mathbb{N}, xyiz \in \mathcal{O}^n, \mathcal{O}^n$

$xyiz = \mathcal{O}^a \mathcal{O}^b \cup \mathcal{O}^c \cup d = \mathcal{O}^{a+b+c} \cup d$

$a+b+c = d$

$b(i-1) \stackrel{-p}{=} 0 \quad -p \quad i=0 \quad \boxed{i=1}$

fun  $n:$

if  $n = 0 : \text{ret } p \circ$

$p \circ : \mathbb{P} \circ$

else: let  $m = n-1$

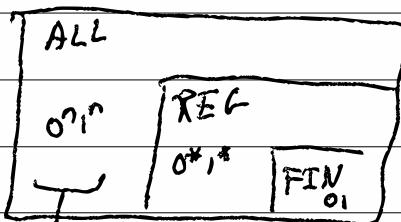
$p \circ : \text{th. } p_n \rightarrow p_{(n+1)}$

let  $p_m = \text{rec } m \Rightarrow \forall n: p_n$

$p \circ p_m$

induction on  $N:$

9-3/ There is stuff outside REG?



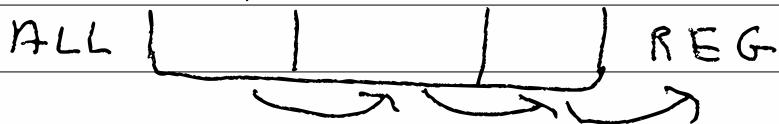
All computers  
are DFAs.

if  $0^n 1^n \in \text{REG}$

~~then~~<sup>and</sup>  $B \in \text{REG}$

then  $0^n 1^n \circ B \in \text{REG}$

$\Sigma_1 \quad \Sigma_0 \quad \text{CFL}$



$0^n 1^n \in \text{REG}$

w where  $\text{count}(0,w) = \text{count}(1,w)$

ww where  $w \in \Sigma^*$

010101  $\in \Sigma^*$

wwR where  $w \in \Sigma^*$

$$\underline{q-y} / \quad 0^x 1 0^y 1 0^{x+y}$$

$$01\ 0100 \Rightarrow "1+1=2" \quad \checkmark$$

$$001\ 0001\ 00000 \Rightarrow "2+3=5" \quad \checkmark$$

$$001\ 0010 \Rightarrow "2+2=1" \quad \times$$

A.p.

$$\exists s . \quad 0^p 1 0^p 1 0^{2p}$$

$$\nexists x y z . \quad x = 0^a \quad y = \underline{0^b} \quad z = 0^c 1 0^p 1 0^{2p}$$

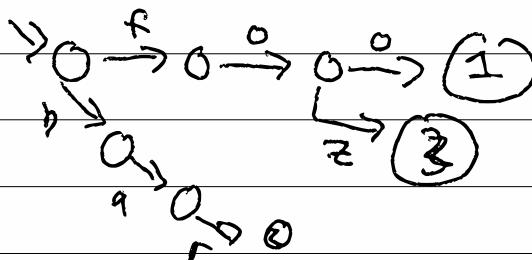
$$\exists ; . \quad \begin{matrix} \text{change this} \\ \text{can't change} \end{matrix}$$

$$\underbrace{z+z=4}$$

$$\Rightarrow \Rightarrow$$

$$\underbrace{g+z=10}$$

12-1 / trie       $\text{foo} \leftrightarrow 1$       digital  
                    $\text{bar} \leftrightarrow 2$       search  
                    $\text{baz} \leftrightarrow 3$       tree



CFGs

start variable       $0^n 1^n$       & REG

#       $S \rightarrow \epsilon$       ← rule, productions, transitions  
 D       $S \rightarrow OSI$       rule =  $\underline{V} \rightarrow \underline{\text{rhs}}$   
 variables      rhs =  $(V \cup \epsilon)^*$   
 non-terminals       $\xrightarrow{\text{terminal}}$   
 Symbols

$\overset{B}{\underset{000111}{S \rightarrow OSI \rightarrow 00S11 \rightarrow 000S111}}$

" derivation "       $\overset{B}{\underset{000111}{\swarrow}}$

$w \in L$       iff       $\exists d. S \xrightarrow{*} w$

$$\frac{L(2)}{M} \left( \begin{array}{l} S \Rightarrow 01S \\ S \Rightarrow SS \end{array} \right) = \emptyset$$

Context-free grammar

$$0101 \in \xrightarrow{\quad} \text{alphabet } (V, \Sigma, R, S)$$

$\downarrow$  finite set       $\downarrow \in V$

$$B \Rightarrow \epsilon$$

$$B \Rightarrow B \cup N \cup B$$

$$N \Rightarrow \epsilon \qquad P(V \times (V \cup \Sigma)^*)$$

$$N \Rightarrow 0N \qquad (V \Rightarrow P(V \cup \Sigma)^*)$$

$$\Sigma(S, 01S),$$

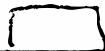
$$R = \{S, SS\} \quad V = \{\Sigma\} \quad \Sigma = \{0, 1\} \quad S = S$$

REG

DFA

REX ( $\cup, \cdot, ^*$ )

CFL



CFG

12-3 / A B

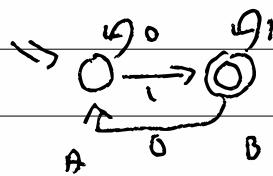
$\rightarrow A \cup B : S \Rightarrow A, S$

$S \Rightarrow B, S$

$A \circ B : S \Rightarrow A, S \quad B, S$

$A \cap B : \times$

DFA  $\Rightarrow$  CFG

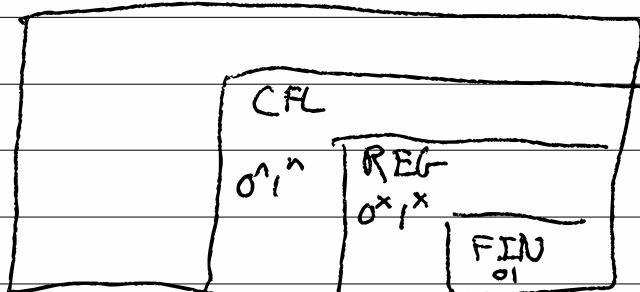
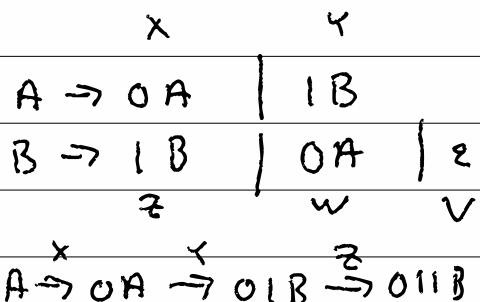


01101

$S = g_0$

$g_i \rightarrow c g_j \text{ iff } \delta(g_i, c) = g_j$

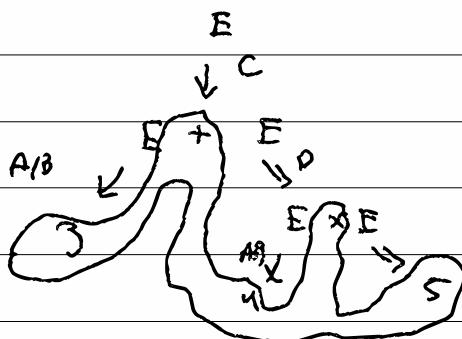
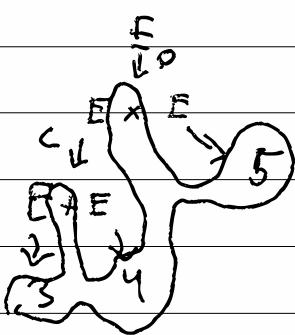
$g_f \rightarrow \epsilon \text{ iff } g_f \in F$



P D C D

$$12 - 4 / E \Rightarrow O \quad | \quad I \quad | \quad E + E \quad | \quad E \times E$$

$$3 + 4 \times 5$$



fowim  $E \Rightarrow n$

fowin  $O = O$

$I = I$

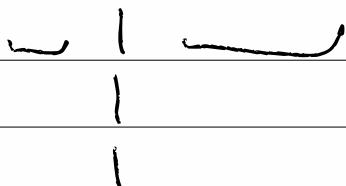
Plus(L, R)  $\Leftarrow$  fowim L + fowim R

Mult(L, R)  $\Leftarrow$  fowim L  $\times$  fowim R

ambiguous = There are multiple  
trees (derivations)

if cond

for the same string



(Z-S) Nice! amb  $\rightarrow$  unamb

not possible

$V, \Sigma, R, S$

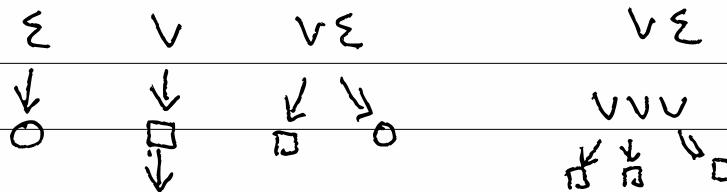
$$V \Rightarrow (V \cup \Sigma)^*$$

$\epsilon$

$\epsilon \Sigma V$

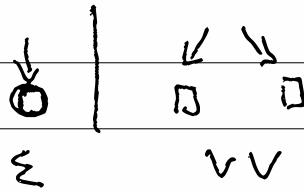
$V$

VVVVV



want: "pure" binary trees

Noam



Chomsky-Normal Form

$$\begin{array}{c|c} V \rightarrow \Sigma & A \rightarrow C \\ V \rightarrow VV & A \rightarrow BC \\ S \rightarrow \epsilon & \#S \end{array}$$

GFG  $\rightarrow$  CNF

(2-5)  $S \rightarrow \epsilon \mid OSI$

— Add a new start

$R \rightarrow S$

$S \rightarrow \epsilon \mid OSI$

— remove  $\epsilon$ -rules

$R \rightarrow S \mid \epsilon$

$S \rightarrow OSI \mid OI$

remove unit rules

$R \rightarrow OSI \mid OI \mid \epsilon$

$S \rightarrow OSI \mid OI$

add extra vars for  $\geq 2$

$R \rightarrow TI \mid OI \mid \epsilon$

$S \rightarrow TI \mid OI$

$T \rightarrow OS$

add terminals "names"

$R \rightarrow TB \mid AB \mid \epsilon$

$S \rightarrow TB \mid AB$

$T \rightarrow AS$

$A \rightarrow O$

$B \rightarrow I$

12-6 / 00111 + 0<sup>n</sup>1<sup>n</sup>

S  $\Rightarrow \epsilon$

S  $\Rightarrow 0S1$

A

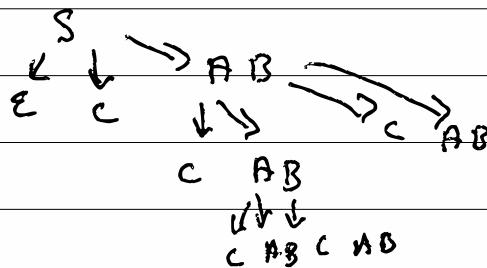
B

A  $\Rightarrow c$  + 1

B  $\Rightarrow$  + 2

A  $\Rightarrow BC$  + 2

A  $\Rightarrow$  done



S  $\Rightarrow XYZ$

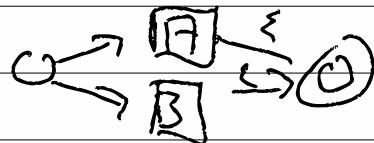
X  $\Rightarrow ZZS$  | 0

Y  $\Rightarrow XYX$  | 1

Z  $\Rightarrow YSX$

$$A = \{x, y, z\}$$

$$B = \{x, y, z\}$$



$$Q = \{S, E, (0, x), (0, y), (0, z), (1, x), (1, y), (1, z)\}$$

$$= \{S, E\} \cup 0 \times A \cup 1 \times B$$

unionstate = start | end | fromA  $g_A$   
| fromB  $g_B$

$$\delta(\text{start}, \epsilon) = \{ \text{fromA } g_0, \text{ fromB } g_0 \}$$

$$\delta(\text{fromA } g_i, c) = \text{fromA } (\delta_B(g_i, c))$$

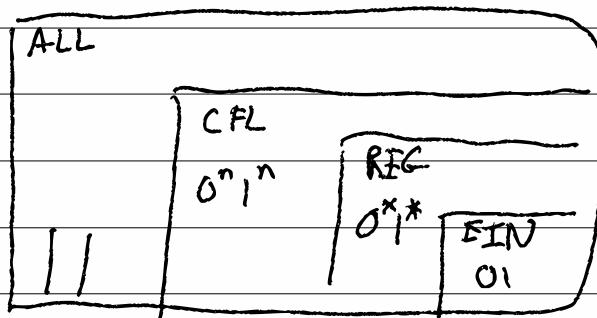
12-7 / new DFA ("blat",  
 $q_i \rightarrow q_i = 0 \text{ if } q_i = 1,$   
sigma, delta,  
 $F$ )



complement (DFA  $\perp$ )  $\Sigma$

new DFA ( $\perp$ ,  $q_i \rightarrow !d.F(q_i)$ )  
( $d.Q$ ,  $d.\Sigma$ ,  $d.\delta$ ,  $d.F$ )

(5-1)



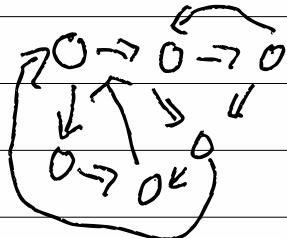
REGULAR

$$\text{DFAs} \longleftrightarrow \text{Regex}$$
$$\epsilon \qquad \qquad \qquad \{\}$$

CFL

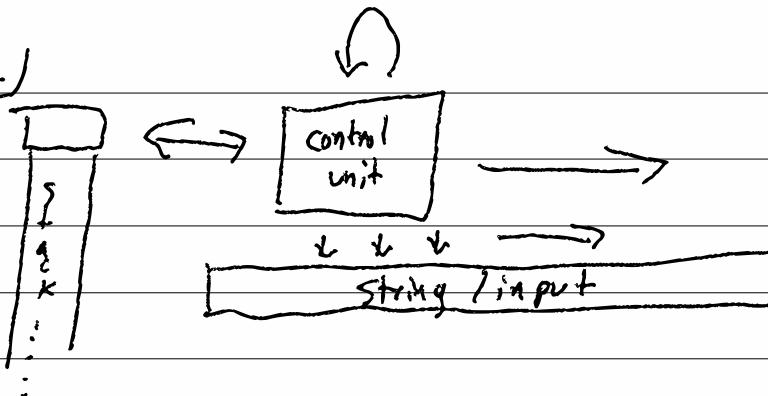
$$\boxed{\text{PDA}} \longleftrightarrow \text{CFG}$$

push-down n automata



$\Sigma^* 1 \Sigma \Sigma$

15-2)



$$\text{DFA} : Q \times \Sigma \rightarrow Q$$

$$\text{Deterministic PDA} : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma$$

$$\text{PDA} : Q \times \Sigma \times \Gamma \rightarrow P(Q \times \Gamma)$$

$\Sigma \cup \{\epsilon\}$

$$Q \times \Sigma \times \Gamma \xrightarrow{\quad} Q \times \Gamma \quad (\text{ignored stack})$$

$$Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma \quad (\text{pop})$$

$$\times \Sigma \rightarrow Q \times \Gamma \quad (\text{push})$$

$$\times \Gamma \rightarrow \times \Gamma \quad (\text{replace})$$

$$\epsilon \times \Gamma \rightarrow \times \Gamma$$

$$\delta(q_i, c) = q_j$$

$$[q_i]_c w \rightarrow [q_j]_w$$

$$c \in \Sigma \quad w \in \Sigma^*$$

$$\delta(q_i, c, \alpha) \ni (q_j, \beta)$$

$$\beta \in [q_j]_w \rightarrow \beta \gamma [q_j]_w$$

$$\alpha \in \Gamma_\Sigma \quad c \in \Sigma$$

$$\beta \in \Gamma^* \quad w \in \Sigma^*$$

config

15-3/ simulate : PDA  $\times \cancel{S} \rightarrow \text{config}$   
 $\text{sim } (\mathcal{Q}, \Sigma, \Gamma, q_0, \delta, F) \quad (\Gamma^*, q_f, \Sigma^*) =$

let  $\alpha : \beta \in g$  in

let  $c = w \in S$  in

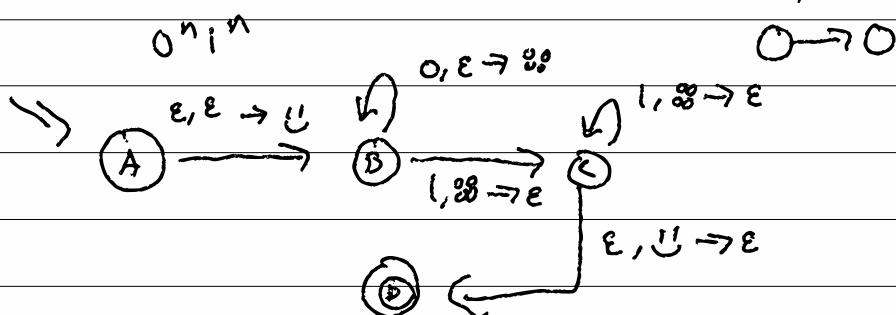
let  $(q_j, \alpha) \in \delta(q_i, c, \alpha)$  in  
 $(\alpha = \beta, q_j, w)$

accepts : PDA  $\times \Sigma^* \rightarrow \text{bool}$

accepts  $p \mid s = \text{while } c, s \notin \Sigma \text{ do}$

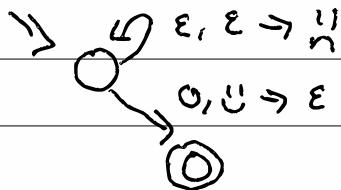
return  $\left( \begin{array}{l} \text{sim } c, q_i \in p, F \\ \text{where } c_0 = (\Sigma, p, q_0, S) \end{array} \right)$

$c, \alpha \Rightarrow \alpha$



$\epsilon[A]0011 \rightarrow \cup[B]0011 \rightarrow \cup^{oo}[B]011 \rightarrow \cup^{oo}oo[B]11$   
 $\rightarrow \cup^{oo}[C]1 \rightarrow \cup[C]\epsilon \rightarrow \epsilon[D]\epsilon \rightarrow \text{YES}$

15-4/



CFG  $\rightarrow$  PDA

$$\begin{array}{l} S \Rightarrow \epsilon \\ S \Rightarrow 0S1 \\ S \Rightarrow Xyz \ 011\ 3yx \end{array} \Rightarrow \begin{array}{c} 0 \rightarrow 0 \rightarrow 0 \\ \downarrow \\ 0 \end{array}$$

$$\Gamma = \Sigma \cup V$$

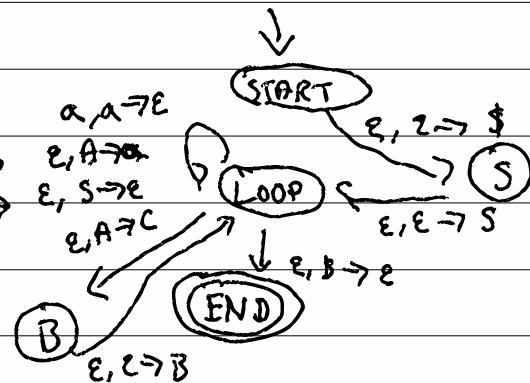
CNF  $\rightarrow$  PDA

$$A \Rightarrow BC \quad (B, C \neq S)$$

$$S \Rightarrow \epsilon$$

$$A \Rightarrow a$$

$$A[\text{Loop}] \Rightarrow C[B] \Rightarrow CB[\text{Loop}]$$



$$\$[L]0011 \rightarrow \$[S]0011 \rightarrow \$S[L]0011 \rightarrow \$ISO[L]0011$$

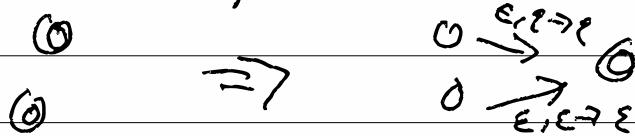
$$\$1\$R[L]11 \leftarrow \$1ISO[L]011 \leftarrow \$1\$[L]011$$

$$\$1\$[L]11 \rightarrow \$1[L]1 \rightarrow \$[L] \rightarrow [\text{END}] \rightarrow \checkmark$$

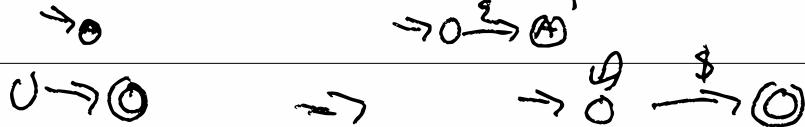
15-5/ PDAs are too complicated

so, we'll simplify with some rules

- Make a single accept



- Guarantee stack is empty on accept



- Always push or pop

push :  $\epsilon$ ,  $\Gamma$

pop :  $\Gamma$ ,  $\epsilon$

X ignore :  $\epsilon$ ,  $\epsilon$

X replace :  $\Gamma$ ,  $\Gamma$

Every symbol pushed is eventually popped

15-6/  $V = (\mathbb{Q} \times \mathbb{Q})$   $S = (g_0, g_f)$

$$(r, +) \in \delta(p, a, \varepsilon)$$

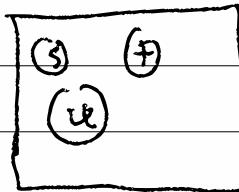
$$(g, \varepsilon) \in (s, b, +)$$

$$\overline{(p, g) \rightarrow a \underset{\in \Sigma}{\in} (r, s) \underset{\in V}{\in} b \underset{\in \Sigma}{\in}}$$

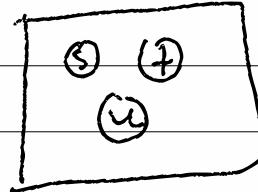
$$(p, p) \rightarrow \varepsilon$$

$$(p, g) \Rightarrow (p, r) (r, g) \quad \forall p, g, r$$

A  $\leftarrow$  cast



B  $\leftarrow$  bst



shown cast, bst

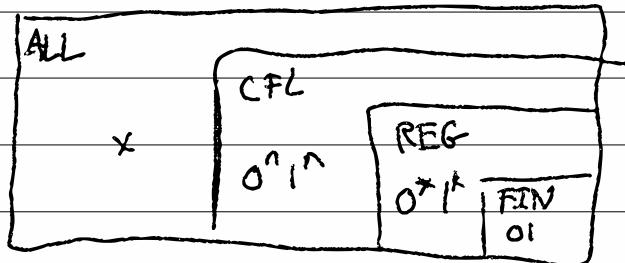
A  $\cup$  B  $\leftarrow$  ~~cast, bst~~

shown  $\langle x, y \rangle$  = start

| and A x

| and B y

16-1



DFA<sub>s</sub>  $\longleftrightarrow$  REX

$0^n 1^n \in \text{REG}$

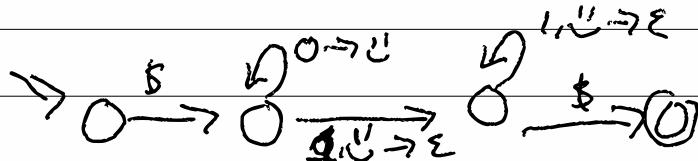
PDA<sub>s</sub>  $\longleftrightarrow$  CFG

$x \notin \text{CFL}$

CFL = ALL

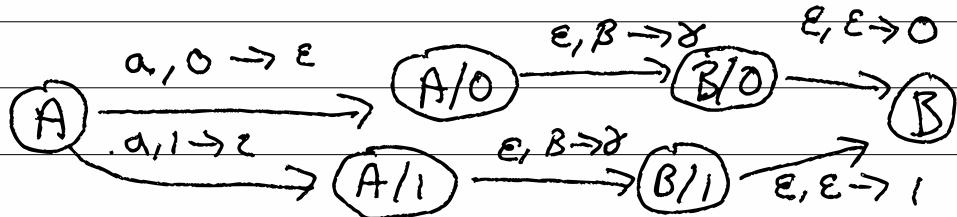
1.	$P \dots \forall x \in \text{REG}, P(x)$	$\forall c \in \text{CFL}, P(c)$
2.	$\exists x \in \text{ALL}, \neg P(x)$	$\exists x \in \text{ALL}, \neg P(x)$
3.	$\Rightarrow x \notin \text{REG}$	$\Rightarrow x \notin \text{CFL}$

$0^n 1^n \rightarrow \xrightarrow{\quad} \textcircled{0} \xrightarrow{\quad} 1 \xrightarrow{\quad} 2 \xrightarrow{\quad} 3$



16-2 Look at 2nd th, by

$\alpha, ?\beta \rightarrow ?\gamma$



$$y \beta Q[A] \omega \Rightarrow y/B[A/0] \omega \Rightarrow y\gamma[B/0] \omega \\ \Rightarrow y\gamma 0[B] \omega$$

$$0^n 1^n \qquad 0^n 1^* 0^n \qquad 0^n 1^n 0^n$$

$$0^n 1^* 0^y \text{ s.t } x+y=2n$$

$$\overbrace{0^n 1^n 0^n}^{0^n 1^* 0^n} \xrightarrow{\epsilon [ ] 0^n \rightarrow \epsilon [ ] \epsilon} \begin{cases} \epsilon [ ] 0^n 1^n 0^n \\ 0^n [ ] 1^n 0^n \\ \epsilon [ ] 0^n \end{cases}$$

$$\epsilon [ ] 0^n 1^n 0^n \rightarrow 0^n [ ] 1^n 0^n \rightarrow 0^{n-1} 00 [ ] 1^n 0^n \\ \rightarrow 00 [ ] 0^n \rightarrow$$

16-3/ RPP

$\forall A \in \text{REG},$

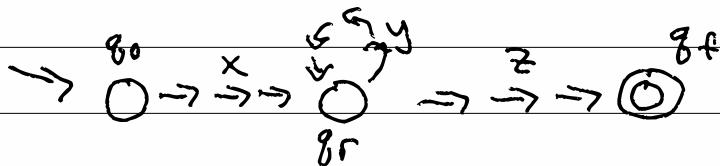
$\exists p \in \mathbb{N},$

$\forall (s \in A \mid |s| \geq p)$

$\exists (x, y, z \in \Sigma^* \mid |xy| \leq p \wedge |y| > 0)$

$\forall i \in \mathbb{N},$

$xy^iz \in A.$

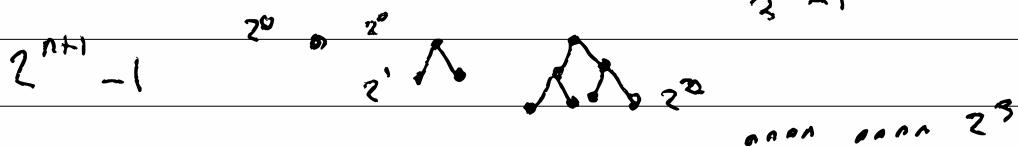


CFG  $G$  (CNF) =  $(V, \Sigma, R, S)$   $S \Rightarrow \epsilon \in R$

$V = \{v_0, \dots, v_n\}$   $S = v_0$   $v_i \Rightarrow \alpha \in R$

$v_i \Rightarrow v_x v_y$

$v_0$	$v_1$	$v_2$	....	$v_n$	$\rightarrow$	$v_r$	$v_r \in V$
$v_1$	$v_2$	$v_2$		$\vdots$		$\vdots$	
$v_2$				$v_n$		$v_r$	



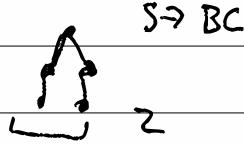
(6-y) minimum length of tree of height  $n$

$S \rightarrow E$

$$\bullet = 0$$

$S \rightarrow \alpha$

$$= 1$$



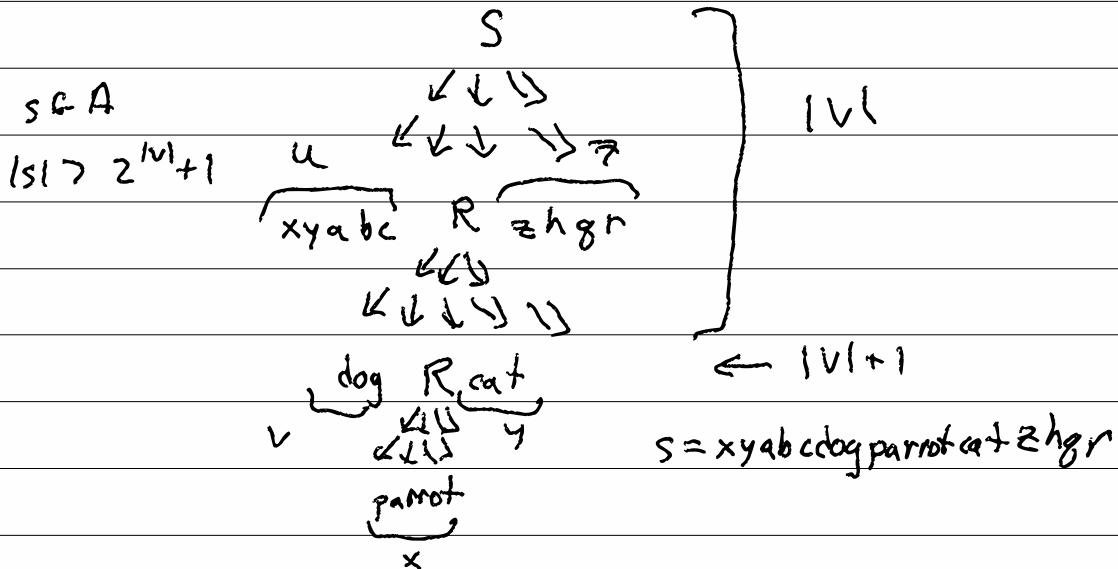
$S \rightarrow BC$

$$n=4$$

$$\Rightarrow$$

$$\text{len} \geq 3$$

If the len of a string is  $> 2^{|V|} + 1$   
then the height must be  $|V| + 1$



$$S \rightarrow u R z$$

$$R \rightarrow v R y$$

$$R \rightarrow x$$

16-5) Context-free pumping property

$\forall A \in \text{CFL},$

$\exists \epsilon \in \Sigma^*$

$\exists p \in \mathbb{N},$

$|s| = 2^{l(s)} + 1$

$\forall s \in A \quad |s| \geq p$

$\exists (u, v, x, y, z \in \Sigma^*) \quad |s = uvxyz|$

$|vxy| \geq p$

$|vy| > 0$

$\forall i \in \mathbb{N},$

$uv^i xy^i z \in A$

$s \Rightarrow uRz$

$s \Rightarrow uRz \rightarrow uxz$

$(i=0)$

$R \Rightarrow vRy$

$s \Rightarrow uRz \Rightarrow uvRyz \rightarrow uv^2 R y^2 z$

$R \Rightarrow x$

$\rightarrow uv^2 xy^2 z \quad (i=2)$

$0^n, n \in \mathbb{N} \in \text{CFL}$

$\stackrel{p}{\sim} s = 0011$

$|vxy| = 2 \leq 4$

$u = 0 \quad v = 0 \quad x = \epsilon \quad y = 1 \quad z = 1$

$|vy| = 2 > 0$

$uv^i xy^i z \in A ? \quad 00^i 1^i 1 = 0^{i+1} 1^{i+1} \in A$

$$16-6 \quad 0^n 1^n 0^n = A$$

given  $p$

$$0^p 1^p 0^p$$

given  $u, v, x, y, z$

$$\begin{aligned} u &= 0^{\vec{w}} & v &= 0^{\vec{v}} & x &= 0^{\vec{x}} & y &= 0^{\vec{y}} \\ z &= 0^{\vec{z}} 1^n 0^p \end{aligned}$$

$u x y$  has repetition

$$\boxed{0^p 1^p 0^p} \rightarrow u x y = 1 \text{ symbol}$$

$$\boxed{0^p 1^p 0^p} \rightarrow u x y = 2 \text{ symbols}$$

$$\boxed{0^p 1^p 0^p} \rightarrow 3 \text{ symbols}$$

$$0^{p+i} 1^p 0^p$$

$$0^p 1^{p+i} 0^{p+i}$$

$$u x y = 0^a 1^p 0^b \quad | \quad | = a + p + b \leq p$$

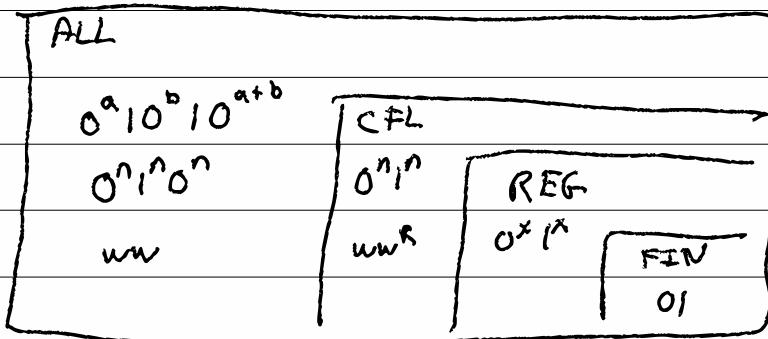
DFAunion = pair  $\langle X, Y \rangle$

NFAunion = start

state from left  $X$  ("x")

state from right  $Y$  ("y")

END



17-1 Finite : 01

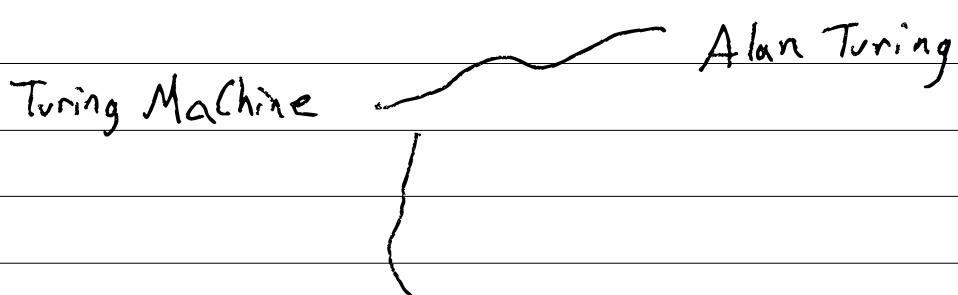
Regular:  $0^* 1^*$

Context-free:  $0^n 1^n$

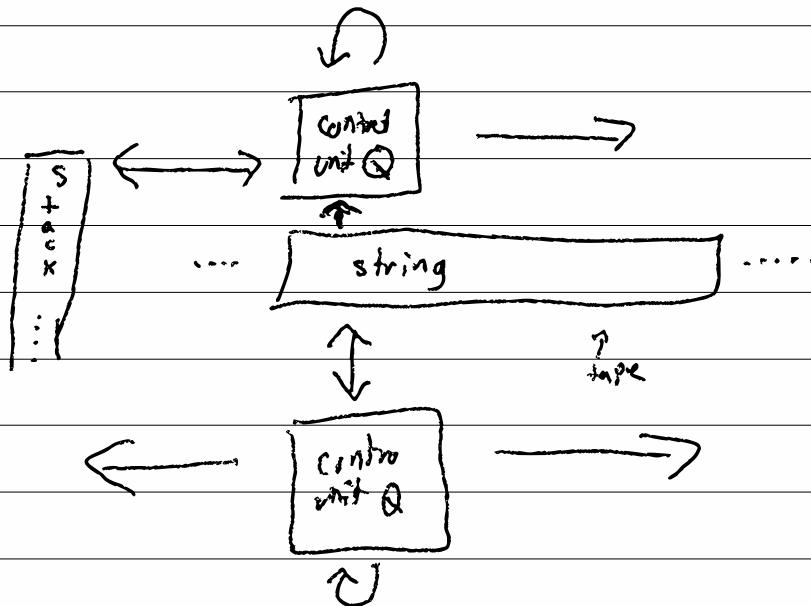
Context-free pumping property  
CFPP

context-free pumping property

$\Rightarrow$  CFG, PDS



Goal: Effective ~~math~~ algorithm for true math.



17-2  $\delta: Q \times \Sigma \rightarrow Q \times \Gamma \times \{L, R\}$

$\Sigma = \{q_0, q_f\}$

states:  $Q$

input:  $\Sigma$

$\sqsubset \in \Sigma$

config:

$\Gamma^* \subset Q \times \Gamma^*$

tape:  $\Gamma$  ( $\Sigma \subseteq \Gamma$ )  $\sqsubset \in \Gamma$

$q_0 \in Q$

$q_a \in Q$  (accepting)

$q_r \in Q$  (rejecting)

$c_0 = \varepsilon[q_0] \sqsubset$

for  $w \in \Sigma^*$  input

$w \in L(m)$  iff

$\varepsilon[q_0] \sqsubset \Rightarrow^* x[q_a]y$

$\delta(q_i, a) = (q_j, b, R)$

$x[q_i]ay \Rightarrow x[b][q_j]y$

$\delta(q_i, a) = (q_j, b, L)$

$x[c][q_i]ay \Rightarrow x[q_j]cb[y]$

$x[q_i]y \Rightarrow \sqsubset x[q_i]y \sqsubset$

left: tape  $\rightarrow$  tape

left  $\varepsilon = \varepsilon : \sqsubset$

left  $(x : c) = x$

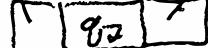
right: tape  $\rightarrow$  tape

right  $\varepsilon = \sqsubset : \varepsilon$

right  $(c : y) = y$

$\perp \leftarrow \overline{0} \leftarrow \boxed{1} \leftarrow \boxed{0}$

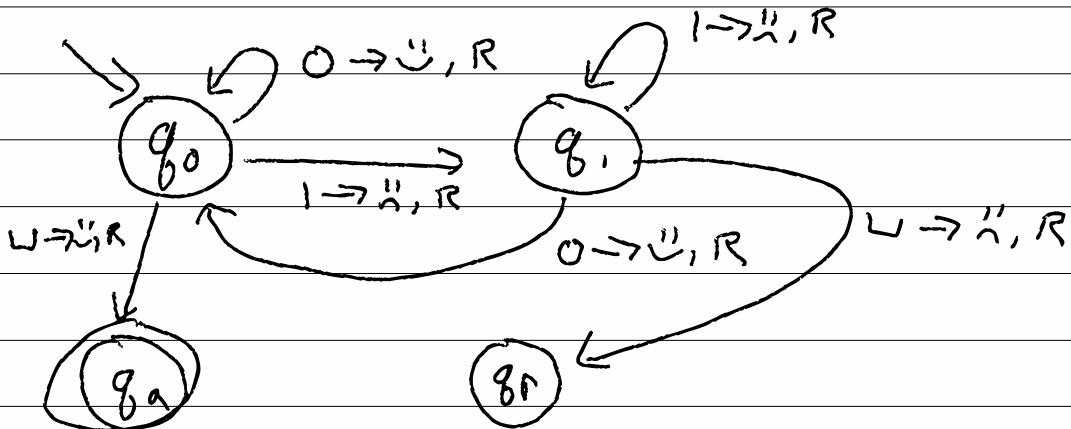
011 [q<sub>2</sub>] 101



$\boxed{1} \rightarrow \boxed{0} \rightarrow \boxed{1} \rightarrow \perp$

(DFA)

173)



$[A]^{110} \rightarrow [B]_{10} \rightarrow [B]_0 \rightarrow [A]_\varepsilon \rightarrow \checkmark$

$\varepsilon [q_0]^{110} \rightarrow 1 [q_1]_{10} \rightarrow 1 1 [q_1]_0 \rightarrow \dots$   
 $\rightarrow 1 1 1 [q_1]_\varepsilon \rightarrow \omega 1 1 1 [q_1] \omega \rightarrow \omega 1 1 1 1 [q_1]$

input:  $(Q, \Sigma, q_0, \delta: Q \times \Sigma \rightarrow Q, F \subseteq Q)$

output:  $Q' = Q \cup \{q_a, q_r\}$

$\Sigma' = \Sigma$

$\Gamma = \varepsilon \cup \{\omega\}$

$q'_0 = q_0$

$\delta'(q_i, c) = (\delta(q_i, c), \omega, R)$  if  $c \notin \omega$

$\delta'(q_i, \omega) = (q_a, \omega, R)$  if  $q_i \in F$

$(q_r, \omega, R)$  if  $q_i \notin F$

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

17-4 /  $w \# w$  where  $w \in \Sigma^*$

I saw a zero

$[] 01 \# 01 \rightarrow \sim [] 1 \# 01 \rightarrow \sim 1 [ ] \# 01$

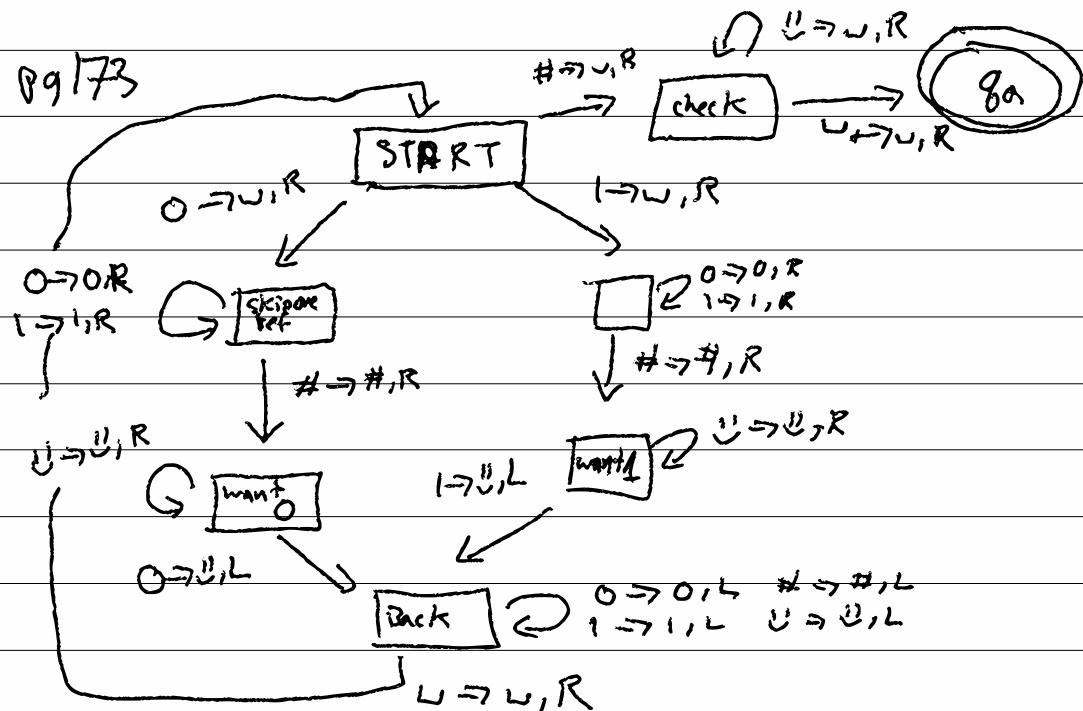
$\sim [ ] \# 01 \rightarrow \sim [ I \text{ expect a zero} ] 01 \rightarrow \sim 1 [ I \text{ am happy} ] \# 01$

$\rightarrow \sim [ H ] 1 \# 01 \rightarrow [ H ] \sim 1 \# 01$

$\rightarrow \sim [ B ] 1 \# 01 \Rightarrow \sim [ I \text{ saw a } 1 ] \# 01$

$\sim \# [ I \text{ expect a } 1 ] 01 \quad \sim \# 0 [ \text{ looked } 1 ] 1$

Pg 173



17-5/ A computable function  
 $f$

is a Turing - machine  
and

$$f(x) = y$$

if  $f$

$$\in [g_0]x \Rightarrow^* w[g_a]y$$

$$\text{add} \begin{matrix} 1 \\ 0 \end{matrix} = \begin{matrix} 1 \end{matrix}$$

$$\begin{matrix} 1 \end{matrix} = \begin{matrix} 10 \end{matrix}$$

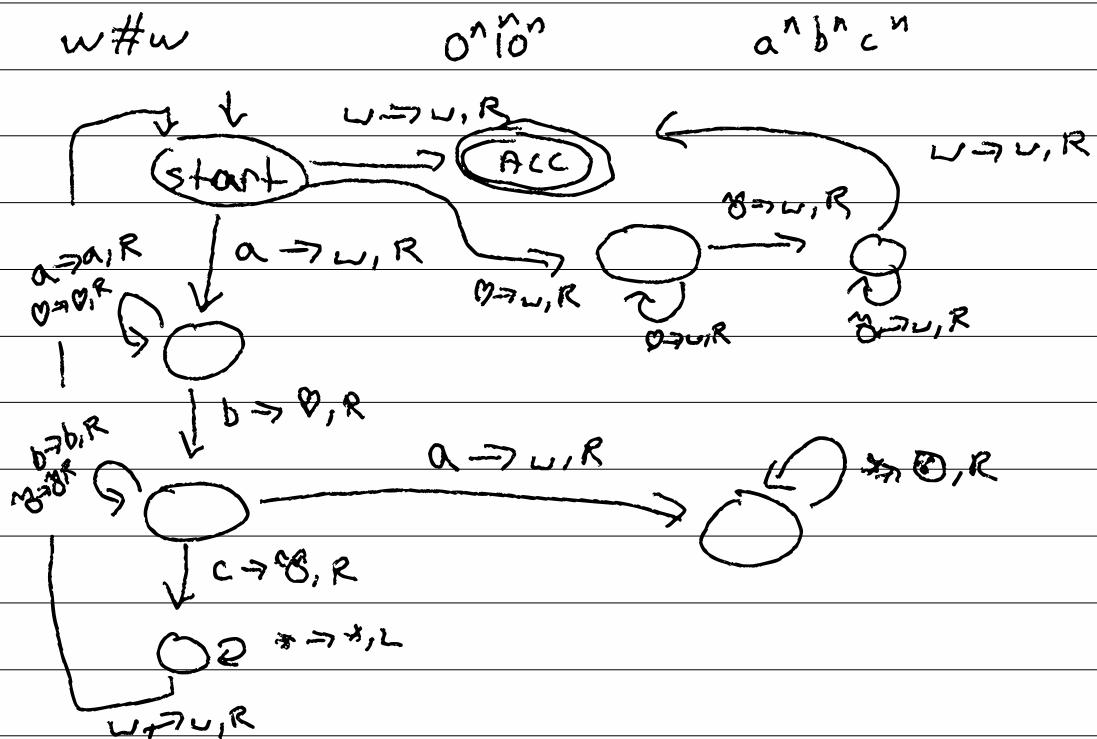
$$\begin{matrix} 10 \end{matrix} = \begin{matrix} 11 \end{matrix}$$

$$\begin{matrix} 11 \end{matrix} = \begin{matrix} 100 \end{matrix}$$

( $\geq \Sigma$ )

18-1]  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$Q = \{q_0, q_f\}$



$$O^x + O^y = O^{x+y}$$

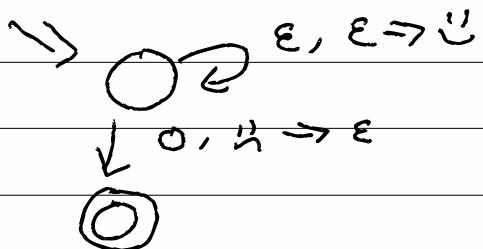
$$f(O^x + O^y) = O^{x+y}$$

18-2) When a DFA runs, how long will it take on input  $w$ ?

$|w|$

When a PDA on input  $w$ ?

$2^{|w|}$  (loop on the stack)



On input  $w$ , what is the destiny of a Turing machine?

ε(ACC) & (REJ, LOOP, DIVERGE)

ACC       $\epsilon [q_0] w \Rightarrow \Rightarrow \Rightarrow \Rightarrow x [q_a] y$

REJ       $\epsilon [q_0] w' \Rightarrow \Rightarrow \Rightarrow \Rightarrow x' [q_r] y'$

LOOP       $\epsilon [q_0] w'' \Rightarrow \Rightarrow z [q_i] \Leftarrow \Rightarrow \Rightarrow z [q_i] u$

D  
IVE  
RGING       $\forall x, q_i, y, \epsilon [q_0] w \Rightarrow^* x [q_i] y$

implies  $x [q_i] y \Rightarrow x' [q_i] y'$   
 $\exists x' q_i y'.$

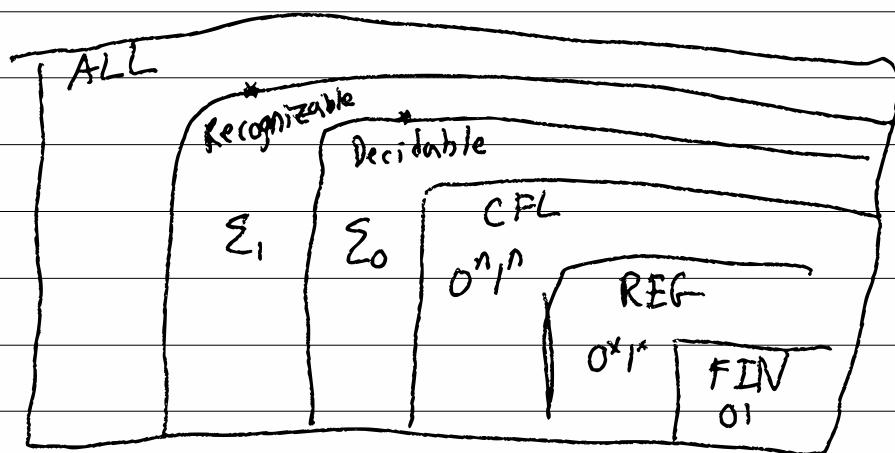
18-3 / A TM is either a

recognizer — may LOOP on  
some input

decider — always ACC or REJ  
never LOOP

a recognizable language,  $A$ , means  $\exists m \in \text{recognizers}$   
 $L(m) = A$

a decidable language,  $B$ , means  $\exists m \in \text{deciders}$   
 $L(m) = B$



Cog

CoC

CiC

# (8-4) A Turing enumerator ...

DFA<sub>s</sub>

PDA<sub>s</sub>

TM<sub>s</sub>

REX<sub>s</sub>

CFG<sub>s</sub>

enumerators

$(Q, \Sigma, \Gamma, q_0, \delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}, g_{P \in \overline{A}})$

really  $Q$   
not  $Q - \{q_0\}$

$w \in L(e)$  iff  $e[q_0] \xrightarrow{\epsilon} \star \xrightarrow{g_p} w$

"recognizer" enumerator  $\Rightarrow$  Outback

"decider" enumerator  $\Rightarrow$  shortest-to-longest  
(lexicographic)

## 18-5) DFA union

NFA union

R&G v

$\xrightarrow{NFA}$

$(Q_x, \Sigma, \delta_{0x}, \delta_x, F_x)$

$(Q_y, \Sigma, \delta_{0y}, \delta_y, F_y)$

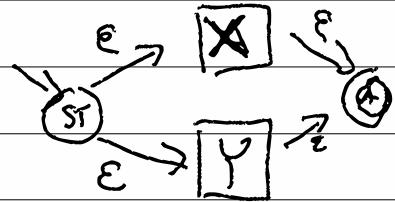
$\Rightarrow$  Pair  $\langle Q_x, Q_y \rangle$

$(Q_x \times Q_y, \Sigma)$

$(\delta_{0x}, \delta_{0y})$ ,

$$\delta((q_x, q_y), c) = (\delta_x(q_x, c), \delta_y(q_y, c))$$

$Q_x \times F_y \cup F_x \times Q_y$ )



$(Q_x, \Sigma, \delta_{0x}, \delta_x, F_x)$   
 $(Q_y, \Sigma, \delta_{0y}, \delta_y, F_y)$

$\emptyset = \{\epsilon \text{ START}, \text{Acc}\}$  ~~if  $c \in \emptyset$~~   
 $\cup \quad \epsilon 03 \times X$   
 $\cup \quad \epsilon 13 \times Y$

$$\delta(\text{START}, \epsilon) = \{ (0, \delta_{0x}), (1, \delta_{0y}) \}$$

$$c = \emptyset$$

$$\delta(\text{ACC}, \epsilon) = \emptyset$$

$$\delta((0, q_x), c) = \{ 03 \times \delta_x(q_x, c) \cup$$

(if  $q_x \in F_x$  then  $\epsilon \text{ Acc} \} \text{ if } c = \epsilon \}$

interface NFAUnionState  $\langle X, Y \rangle$

NUS\_START () =  $\langle \text{nat} \rightarrow X \text{ or } Y \rangle$

NUS\_ACC ()

NUS\_FromX (X)

NUS\_FromY (Y)

## 19-1/ Closure properties

A set  $A$

and operation  $f : A \rightarrow \underline{A}$

" $A$  is closed under  $f$ "

The regular languages are closed under

$C, \cup, \cap, \circ, ^*$

The CFLs

$\cup, \circ, ^*$

What are  $\Sigma_0$  (decidable)

$\Sigma_1$  (recognizable) languages  
closed under?

Is  $\Sigma_0$  closed under complement? ✓

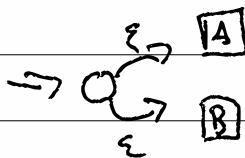
$\Sigma_1$

## 19-2 / Union

DFA<sub>s</sub>  $Q = Q_A \times Q_B$

(did both at same time)

NFA<sub>s</sub>



TMs — can't do NFA (deterministic)

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

$Q \rightarrow Q_A \times Q_B$  tape  $\rightarrow$  two tapes

TMs — simple (like assembly) DFA<sub>s</sub>

↑

↑

↑

MTMs

C

NFA

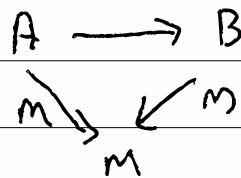
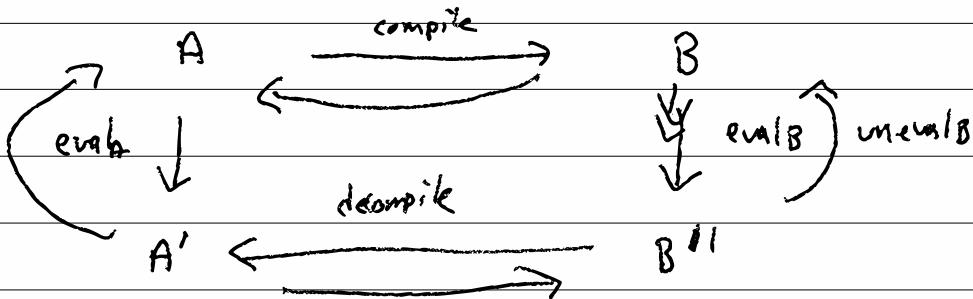


# 19-3/ Compiler correctness

$\forall A \in \text{input. } \exists B \in \text{output. } m(A) = m(B)$

$\xrightarrow{\text{C, assembly}}$        $\xrightarrow{\text{asm, binary}}$   
 $\text{NFA}$                    $\text{DFA}$

bi-simulation



Galois connection

(a-y) TM w/ "stay"

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

$$\delta_{SPTM} : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

$$\delta(q_i, a) = (q_j, b, S)$$

$$u[q_i]av \Rightarrow u[q_j]bv$$

$$\delta(q_i, a) = (q_j, b, S)$$

$$\Rightarrow$$

$$(q_k, c, L/R)$$

$$\delta(q_i, a) = (q_{aj}, b, R)$$

where  $\delta(q_i, b) = ?$

$$u[q_i]av$$

$$ua[q_{almost j}]v$$

$$\leftarrow$$

$$u[q_i]bv$$

$$\forall x \in \Gamma$$

$$\delta(q_{aj}, x) = (q_j, x, L)$$

19-5) MTMs (exist to do  $v/n$ )

$$\delta_A : Q_A \times \Gamma_A \rightarrow Q_A \times \Gamma_A \times \{L, R\}$$

$$\delta_B : Q_B \times \Gamma_B \rightarrow Q_B \times \Gamma_B \times \{L, R\}$$

$$\delta_{A \cup B} : (Q_A \times Q_B) \times \Gamma_A \times \Gamma_B$$

$$\rightarrow (Q_A \times Q_B) \times (\Gamma_A \times \{L, R\}) \times (\Gamma_B \times \{L, R\})$$

$$\delta_{A \cup B} ((q_a, q_b), (t_a, t_b)) =$$

$$\text{let } (q'_a, t'_a, d'_a) = \delta_A (q_a, t_a)$$

$$(q'_b, t'_b, d'_b) = \delta_B (q_b, t_b) \text{ in}$$

$$((q'_a, q'_b), (t'_a, d'_a), (t'_b, d'_b)) \text{ if } q'_a = \text{Acc}$$

then Acc

Multi-tape Turing Machine

$$\delta : Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{L, R\})^k$$

$$\delta(q_i, a, \alpha) =$$

$$(q_j, (b, L), (\beta, R))$$

$$\begin{matrix} u \\ x \end{matrix} \left[ \begin{matrix} q_i \\ \alpha \end{matrix} \right] \begin{matrix} a \\ v \end{matrix} \Rightarrow \begin{matrix} u \\ x \beta \end{matrix} \left[ \begin{matrix} q_j \\ \beta \end{matrix} \right] \begin{matrix} cbv \\ y \end{matrix}$$

19-6      MTM config       $\xrightarrow{\text{complete}}$       STM config

eval  $\downarrow$        $\Gamma = \{\#\} \cup$   $\downarrow$   
 step/  $\Rightarrow$   $\cup \Gamma_A \cup (\Gamma_A \times \Sigma^*)$   $\downarrow$   
 new config  $\cup \Gamma_B \cup (\Gamma_B \times \Sigma^*)$  new s config

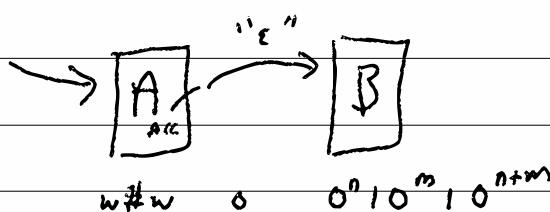
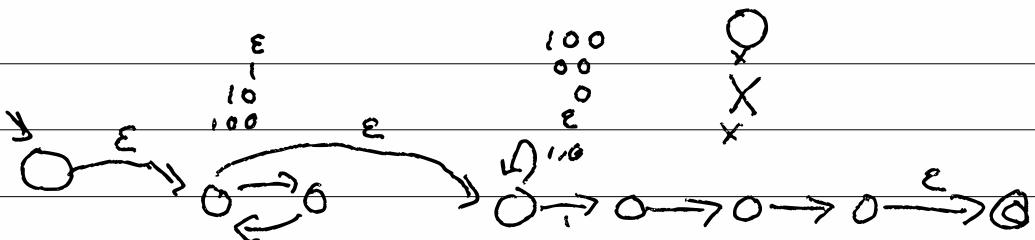
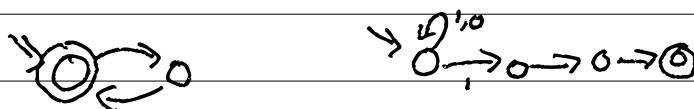
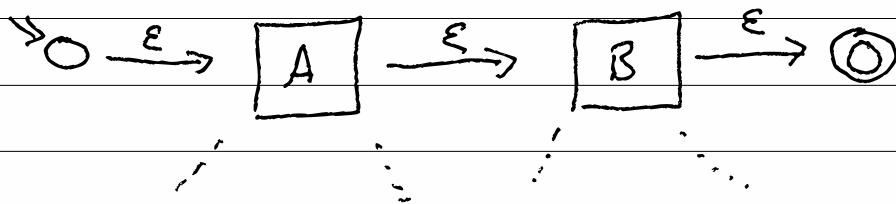
$a \in [q_i]_{\alpha y}$   $\xrightarrow{\quad}$   $[q_i]_{\alpha y} \neq x \circ y$   
 $x$   $(q_i, \alpha)$   $(q_i, \alpha, \alpha)$   
 $\downarrow$   $(q_i, (b, L))$   $(q_i, (L, L), (B, R))$   
 $a \in [q_i]_{\beta y}$   $\leftarrow$   $(q_i)$   
 $x \in [q_i]_{\beta y} \neq x \circ y$

$\Sigma_0$  is closed under  $\cup$  and  $\cap$  ✓  
 $\Sigma_1$  ✓



20+1  $\Sigma_0 : C, n, v, \circ, *, \epsilon$   
 $\Sigma_1 : \square, n, v,$

$f(g)$  is a comp fun &  $g(x)$  is a comp  
 $f(g(x))$  is comp



$0^n \# 0^m \quad 0^n \# 0^m 0^l 0^k 0^l 0^m$

20-2) Given TM M for lang A  $(L(M) = A)$   
N for lang B

Then

type 0: concat (M, N)  $\Leftarrow$  Given input x,  
input (x) split input into  $x = uv$  in all possible  
ways (nondeterministically) try next

type 1: u run M(u) (if REJ, then REJ)

type 2: v run N(v) (if REJ, then try next)

type 3: sim type not Acc if both Acc

not REJ

$\epsilon$  abc - REJ

a bc - M.ACCEPT

ab c - N.Loops

abc  $\epsilon$  - Acc

$\Sigma_0 : 0 \rightarrow \checkmark$

$\Sigma_1 : 0 \rightarrow \checkmark$

# Z03/ Non-deterministic Turing Machine

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

$$\text{Non-det} : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$$

$$2\text{-choices} : Q \times \Gamma \rightarrow (Q \times Q) \xrightarrow{\text{fork()}} \begin{cases} L \\ R \end{cases} \Rightarrow \text{rel(config)}$$

$$\underbrace{\delta(q_i, a) = (q_j, b, R)}_{x[q_i]ay \Rightarrow x[b]q_jy} \xrightarrow{(a, b)} \begin{cases} \omega \\ R \end{cases} \quad P(\text{config}_q, \text{config})$$

$$\underbrace{\delta(q_i, a) = (q_j, q_k)}_{x[q_i]ay \Rightarrow x[q_j]ay} \xrightarrow{(a \in h)} \begin{cases} 0=0 \\ x=y \\ a=b \\ ax=by \\ a(x+y)=b(y+ax) \end{cases}$$

= : rel (num-sentences)

$$\frac{x=y}{a+b} \quad \frac{x=y}{a+b+y}$$

$$\frac{a=b}{x=y} \quad \frac{a=b}{a(x+y)=b(y+ax)}$$

$$\underbrace{\delta(q_i, a) = (q_j, q_k)}_{x[q_i]ay \Rightarrow x[q_k]ay} \xrightarrow{(1 \in h)} \text{"oracle"-model}$$



reject if

$\Rightarrow^x []$

## 20-4] Forking semantics

F-config = Sequence of config

acc if

for input  $w$

F-config<sub>0</sub> =  $["\varepsilon [q_0]w"]$

$\Rightarrow^* ["x[q_0]y", c_2 \dots c_n]$

$$S(q_i, a) = (q_j, b, R)$$

$["x[q_j]y", c_2 \dots c_n]$

$$["x[q_i]ay", c_2 \dots c_n]$$

$\Rightarrow$

$[c_2 \dots c_n]$

$$[c_2 \dots c_n, "xb[q_j]y"]$$

$$S(q_i, a) = (q_j, q_k)$$

$$["x[q_i]ay", c_2, \dots, c_n]$$

$$\Rightarrow [c_2, \dots, c_n, "x[q_i]ay", "x[q_k]ay"]$$

$[c_1, \dots, c_n]$

Fconfig<sub>0</sub>  $\longrightarrow$

config

$c_1 \# c_2 \# \dots \# c_n$

$\downarrow$   
run

$\downarrow$   
 $x[q_i]y$

$\wedge$   $\Rightarrow x[q_j]y$   $\Rightarrow Q$

Fconfig<sub>1</sub>  $\leftarrow$

config

$\Gamma' = \Gamma \cup Q$

O  $\rightarrow$  ND  $\rightarrow$  2ND  $\rightarrow$  BT

$\rightarrow$  Forking

✓

20-5/

Acc

$$- \quad \delta(q_i, a) = (\text{qAcc}, \cup, \omega)$$

REJ

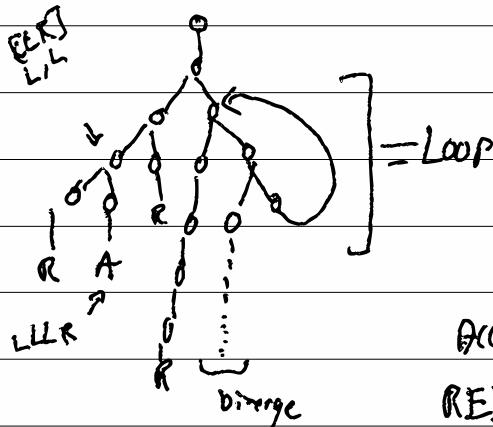
$$- \quad \delta(q_i, a) = (\text{qREJ}, \cup, \cup)$$

?

$$- \quad \delta(q_i, a) = (q_j, b, L/R)$$

?

$$- \quad \delta(q_i, a) = (q_j, q_k)$$



$$\text{map} = \{L, R\}^*$$

explain in

lexicographic  
order

Acc if some map leads to Acc  
REJ if entire "level" (length) is REJ

$$\text{if ans on level } k \quad 2^k + 2^{k-1} + 2^{k-2} + \dots = \sum_{i=0}^k 2^i$$

$k 2^k$

tape 0:  $\omega$

tape 1: current map  $\rightarrow$  add 1

tape 2: simulation tape

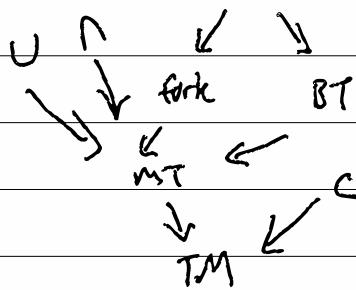
20-6

0  $\leftarrow$  B

↓

ND

2 ND

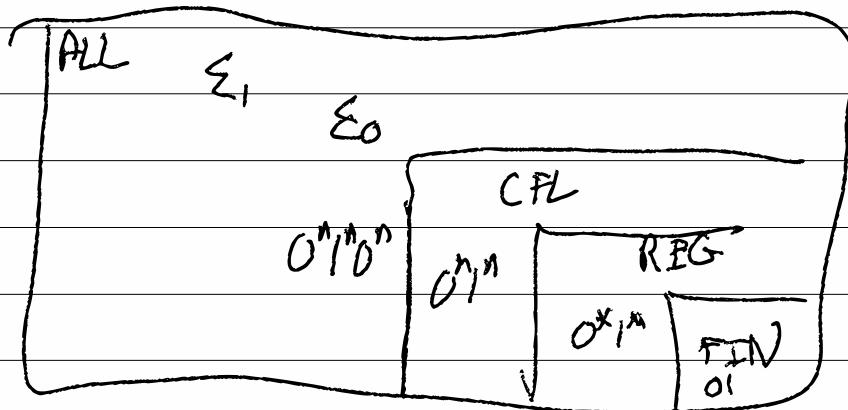


$w \in A^*$  iff

$w = w_1 \dots w_n$

where  $w_i \in A$

21-1)  $\Sigma_0$  (decidable - Y/N)  $C, U, n, o, *$   
 $\Sigma_1$  (recognizable - Y/N/L)  $\square U, n, o, *$

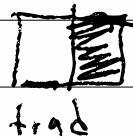


## Church-Turing Thesis

"Algorithm means Turing Machine or  $\lambda$ -calculus term"

Human = Math.  $\approx$  Mathz  
 TM  $\xleftrightarrow{\text{def}}$   $\lambda$

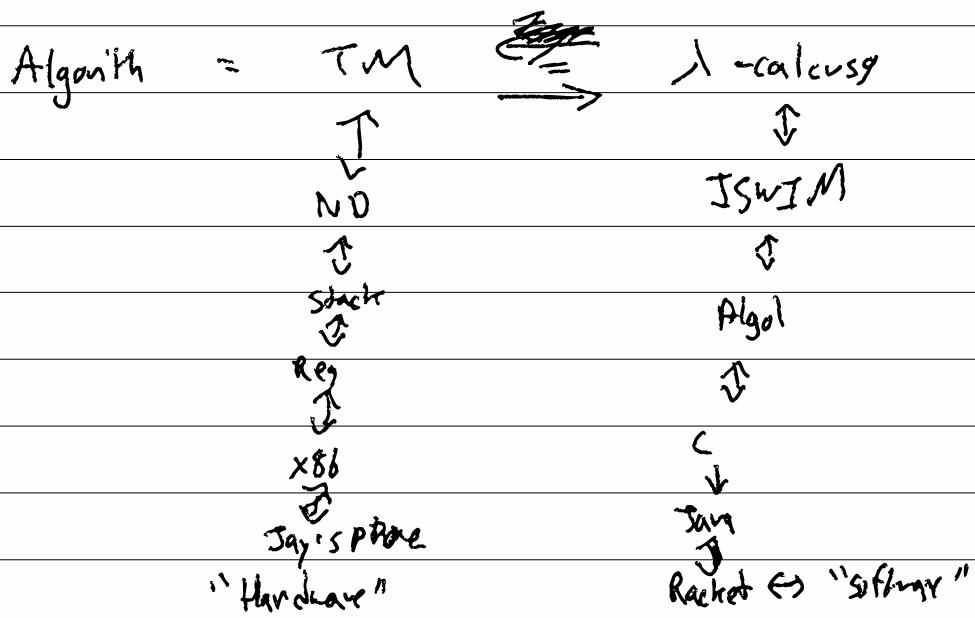
$$0 \xleftarrow{(0,1)} \xrightarrow{(1,-1)}$$



$$(0.5, 0.5) \xrightarrow{(0.3, 0.7)} (0.1, 0.8)$$

21-2 Humans caring means  $N(0, +1)$

Love means Occipital / TinderMath<sub>23</sub>  
Movie taste means Netflix advice



21-3/ 10th: "Devise an effective method to find the integer roots of any polynomial of any number of variables"

$$ax^2 + bx + c = 0$$

$$\Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= -b \pm \sqrt{b^2 - 4ac} \over 2a$$

$$dx^3 + \dots$$

$$ex^4 + \dots$$

"one variable"

$k=2$

$$c_0 = c \quad c_1 = b$$

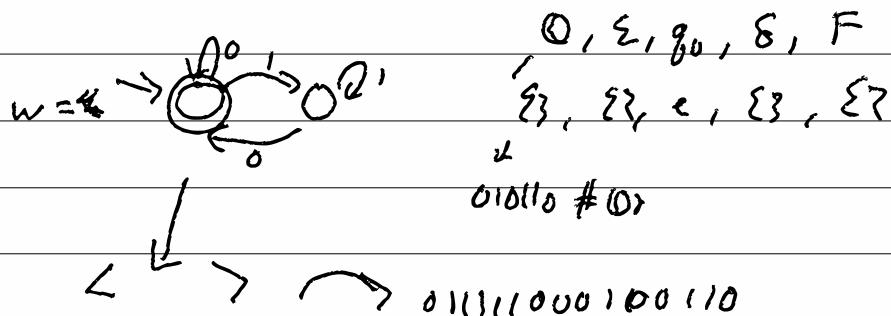
$$c_2 = a$$

$$\sum_{i=0}^k c_i x^i = 0$$

$$\text{try } x = 0, \pm 1, -1, \pm 2, -2, \pm 3, -3$$

$$K \cdot \frac{c_{\max}}{c_k}$$

21-y)   $\Sigma_0 \neq \Sigma_1$



DFA( $\text{REF}$ )  $\subseteq$  TM( $\Sigma_0$ )

$$\delta'(q_i, c) = (\delta(q_i, c), \omega, R)$$

$$\delta'(q_i, \omega) = \begin{cases} \text{acc} & \text{if } q_i \in F \\ \text{rej} & \text{o.w.} \end{cases}, \forall \omega$$

$\text{A}_{\text{DFA}} \ni \langle M, w \rangle$  s.t.  $M$  is a DFA and  $w \in L(M)$

$\text{A}_{\text{NFA}}$

$\text{A}_{\text{CFL}}$

$\text{A}_{\text{TM}} \ni \langle M, w \rangle$  s.t.  $M$  is a TM and  $w \in L(M)$

Universal Turing Machine

ZI-S/  $A_{TM} \in \Sigma_0$  or  $\in \Sigma_1$ ?

look at code

look for loops

None  $\rightarrow \Sigma_0$  yes  $\rightarrow \Sigma_1$

$A_{TM} / \langle M \text{ runs forever}, 0110 \rangle$

= should = Reject ( $\Sigma_0$ )

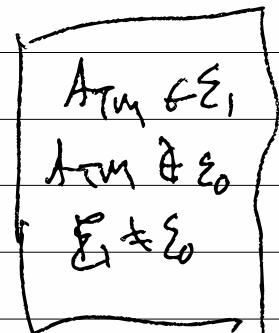
reject or loop ( $\Sigma_1$ )

problem  $\in \Sigma_0$

Make a new TM  $D(\langle M \rangle) =$

"Run  $A_{TM}$  on  $\langle M, \langle M \rangle \rangle$ .

Output the opposite of  $A_{TM}$ ."



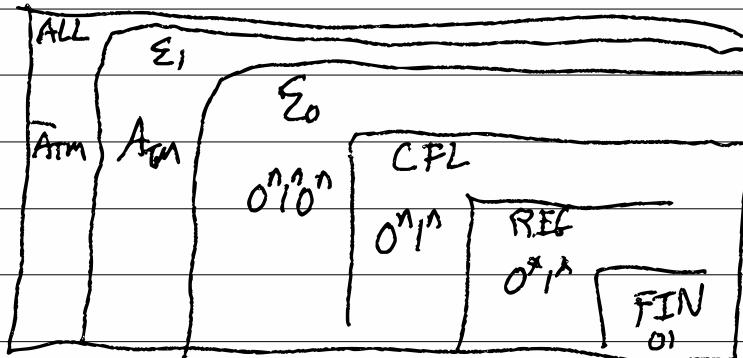
$D(M) = \begin{cases} \text{accept} & \text{if } M \text{ does not accept } \langle M \rangle \\ \text{reject} & \text{if } M \text{ does not accept } \langle M \rangle \end{cases}$

Run  $D$  on  $\langle D \rangle = \begin{cases} \text{accept if } D \text{ d.n.a } \langle D \rangle \\ \text{reject if } D \text{ d.a. } \langle D \rangle \end{cases}$

Liar's Paradox

Drinker's Paradox

1-22



$A \in \Sigma_0$  iff  $A \in \Sigma_1 \wedge A \in \Sigma_2$   
 $\Rightarrow$ : trivial

$\Leftarrow$  : imagine  $L(M_A) = A$ ,  $L(M_N) = \overline{A}$

$$M(x) = \begin{cases} Y & \text{if } x \in A \\ N & \text{if } x \notin A \end{cases}$$

M runs  $M_4$  and  $M_N$  simultaneously  
and uses the first answer

know:  $A_{TM} \notin \Sigma_0$  know:  $B \in \Sigma_0$  iff  $B \in \Sigma$ ,  
 know:  $A_{TM} \in \Sigma_1$ ,  $\neg(A_{TM} \in \Sigma_1) \wedge \overline{B} \in \Sigma_1$ ,  
 $\wedge \overline{A_{TM}} \in \Sigma_1$ )  
 $\Rightarrow A_{TM} \notin \Sigma_1 \vee \overline{A_{TM}} \in \Sigma_1$

222 /  $\text{ATM} \ni \langle M, w \rangle$  iff  $M$  accepts  $w$

$\text{ATM} \ni x$  ; if  $x \in \langle M, w \rangle$   
or  $x = \langle M, w \rangle$  and  
 $M$  does not accept  $w$   
not "rejects"

$\text{ATM}$  must predict when a  
TM will loop / diverge

predict if TM  $M$  "halts"

$\text{ATM}$  is the "Halting Problem"  
is Unrecognizable

22-3    `bool x = read()`

`[obj] o = NULL;`      `Dog`

`if (x) {`      `Typed Racket`

`o = new Cat(); }`      `[occurrence`

`else {`      `typing`

`o = new Dog(); }`      `→ TypeScript`

`... (ignore or x)`      `Hack)`

`if (x) {`

`o.purr(); }`

`else {`

`o.bark(); }`

$\widehat{A}_{TM} \circ U^{\star 1^{\star}}$

# 22-4) Sizes of infinity

$\aleph_0$

$x_1, \dots, x_{\aleph_0}$

$N$	$\mathbb{Z}$	$\mathbb{Q}$	$\mathbb{R}$	$X$
$0, 1, 2$	$-2, +3$	$\frac{1}{4}, \frac{2}{13,1}$	$\pi, e, 0.\overline{3}$	$\infty$

$$\rightarrow p^{\infty} =$$

$$R \cup \{+\infty, -\infty\}$$

$$\left| \{ \text{kisses, hugs, puppy dogs} \} \right| = 3$$

Peano

$$(0^* 1^*)$$

$$N = 0 \uparrow S N$$

$$f : \begin{array}{c} X \\ \xrightarrow{\quad} \\ Y \end{array}$$

$$\emptyset \quad P$$

$$P(\emptyset) = \{\emptyset\}$$

$$P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$$

$$P(\{\emptyset, \{\emptyset\}\}) = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$$

$$\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$$

onto :

$$S(x) = x \cup \{x\}$$

$$\forall b \in Y, \exists a \in X, f(a) = b$$

$$\text{same size } (A, B) := \exists f : A \rightarrow B, \text{ onto}(f) \wedge \text{one-to-one}(f)$$

$$\Rightarrow$$

$$|A| = |B|$$

22-5  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \dots\}$

$|N|$

=

| Even numbers |

$\{0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, \dots\}$

$$f(a) = 2 \times a$$

$$2 \times a = 2 \times b$$

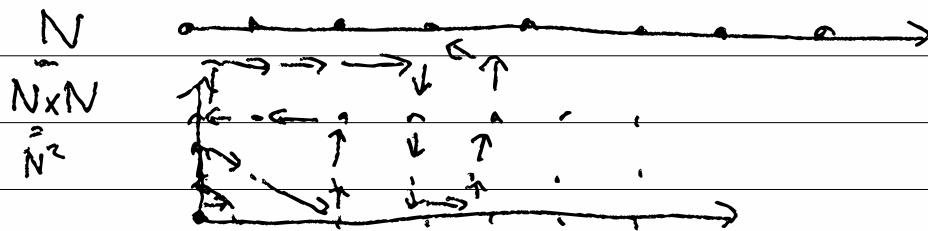
$$\Rightarrow a = b$$

$$\text{A.e. } \exists n. \quad 2 \times n = \underbrace{e}_{2 \times m}$$

$$|N| = \aleph_0$$

(Georg Cantor)

"countable"



$$f(x) = (y, z)$$

$$f(x, y) = \frac{1}{2}(x+y)(x+y+1) + y \quad (\text{Cantor Pairing function})$$

$$\underline{22-6} \quad |N| = |N \times N|$$

$$|N| = |N^k| \quad (k \geq 1)$$

$$|N| = |\mathbb{N}^{k+1}| \quad \text{given} \quad |N| = |N^k|$$

smt g :  $\underbrace{N \times \dots \times N}_{k+1} \rightarrow N$  ex f :  $N \times \dots \times N \xrightarrow{k} N$

$$g(\text{before} \dots, \text{last}) = \\ \text{center}(f(\text{before}), \text{last})$$

$$\text{sort} : (A \xrightarrow{\times} A \rightarrow \text{Bool}) \times \text{List}(A) \rightarrow \text{List}(A)$$

property-based testing

$$f : N \rightarrow \text{List}(A) \quad f \xrightarrow{\text{one onto}} (\text{bijection})$$

fuzz / enumerative testing

Fair Enumeration

$$\text{22-7) } |\Sigma_0| \text{ and } |\Sigma_1| \text{ and } |\text{TM}_S| = |N|$$

$$|\Sigma_0| \leq |\Sigma_1| \leq |\text{TM}_S| \leq |N|$$

$$\begin{aligned} \text{TM} &= Q \times \Sigma^* \times \Gamma \\ &= (N)^k \times N^{g^k} \times g^k \times g^r \\ &= (N)^k \times \{0, \dots, k\} \times (\{0, \dots, k\}^k \times \{0, \dots, g\}^k \times \\ &\quad \{0, \dots, k\}^k \times \{0, \dots, g\}^k \times \{L, R\}) \\ &\times \{0, \dots, k\} \times \{0, \dots, k\} \\ &\leq N \times N \times N \\ &\approx N \end{aligned}$$