

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

$$s(0) = 0$$

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

$\cup = \Sigma^*$ ← 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())));

137

1-6 Every alphabet has a lexicographical ordering of the strings in Σ^*

$\Sigma = \{0, 1\}$ $\Sigma^0 = \{0, 1\}$
 $\Sigma^1 = \{00, 1\}$
 $\Sigma^2 = \{000, 10, 11, 100\}$
 $\Sigma^3 = \{0000, 100, 110, 111, 1000, 1010, 1100, 1011\}$
 new list elements
 $(\text{Zero}, \text{One});$

$|A|^i$ where $i \in \text{layer } 2^{\text{th}}$ looking at $\underbrace{111}_{2^3}$

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

$$\text{lexi } \beta \circ = \varepsilon$$

$$\text{lex}; \quad B \quad \mid = 0$$

$$\log B_z = 1$$

$$\text{lex: } B \quad 6 = 141$$

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 Σ Universe Σ^*

{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
$\overline{\text{FIN}}$	

$\emptyset \in \text{FIN}$

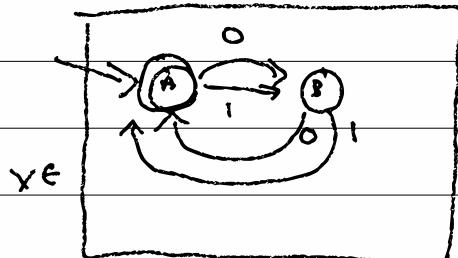
$\forall x \in \Sigma^*, \{x\} \in \text{FIN}$

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

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

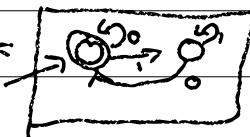
All even strings =

DFA - a deterministic finite automata



Σ or Σ^*

even numbers =



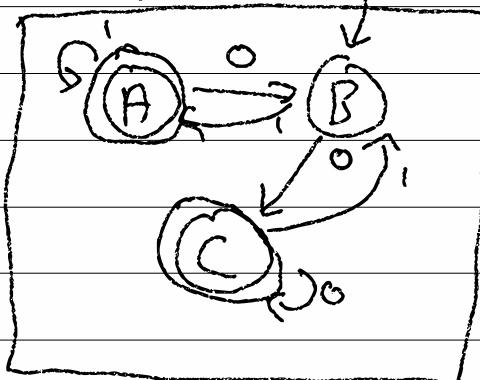
○ - states $\Sigma A, B \}$

→ ○ - start state A

○○ - accepting states $\Sigma A \}$

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

x - labels are Σ



$\Sigma A, B, C \}$

$\Sigma A, C \}$

$\Sigma = \Sigma_0, 1 \}$

A	0	1
B	C	A
C	C	B

2-3) $x \in \text{DFA} (\underbrace{\text{states}, \text{alphabet}, \text{start}, \text{accepting}}_{\text{states } Q, \Sigma, q_0 \in Q, F \subseteq Q}, \delta: Q \times \Sigma \rightarrow Q - \text{transitions})$

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

config update function : config $\times \text{DFA} \rightarrow \text{config}$
 $[q]w \rightarrow [q']w'$

$[q_i]x \rightarrow [q_j]y \text{ iff } \delta(q_i, x) = q_j$
 $x \in \text{DFA} \text{ iff } [q_0]x \Rightarrow \Rightarrow \Rightarrow \Rightarrow [q_f] \in$
 and $q_f \in F$

0110 $\in \text{EvenLen}$;iff $[A]0110 \rightarrow [B]110 \rightarrow [A]10$
 $\rightarrow [B]0 \rightarrow [A] \in AF\{A\}$ ✓

class DFA Σ

.. $Q, \Sigma, F, q_0, \delta \dots$

public bool accepts (String x) {

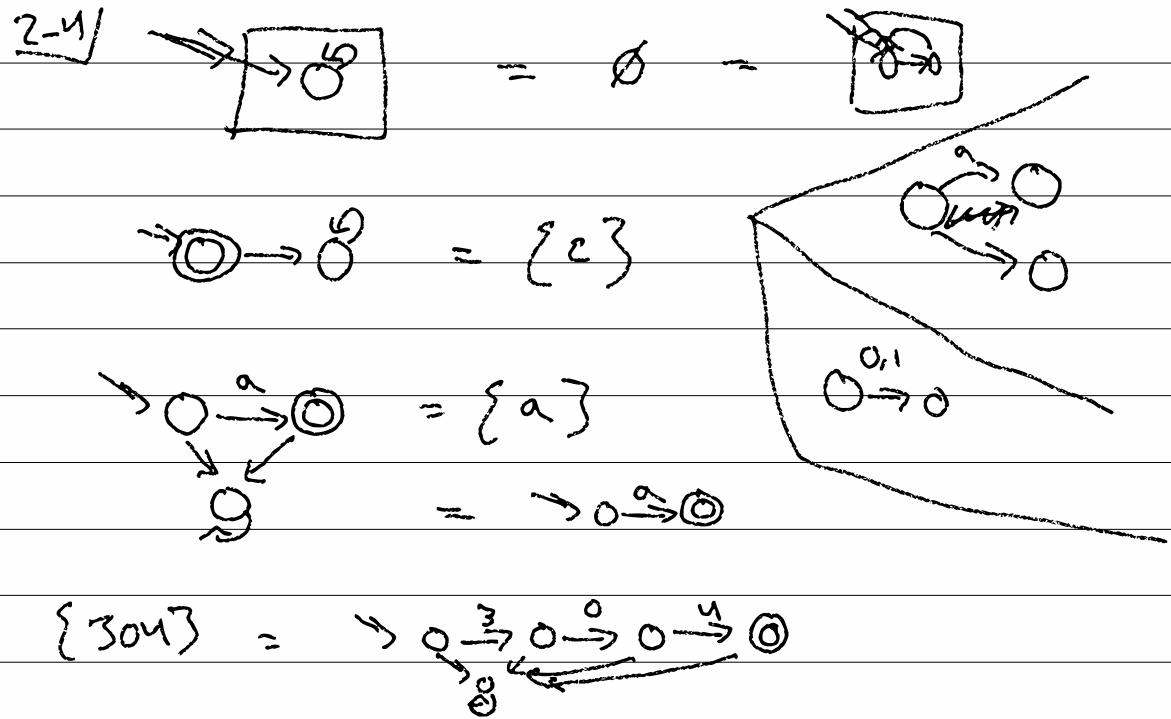
State $q_i = q_0;$

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

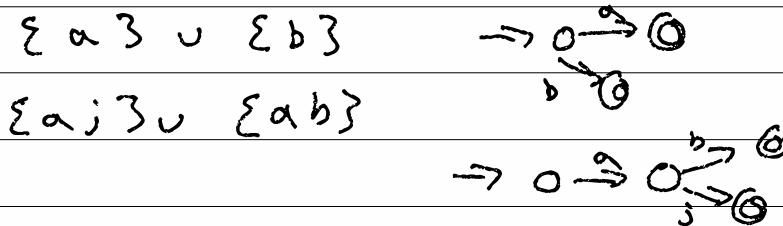
$q_i = \delta(q_i, x, \text{first}());$

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

return $F, \text{in}(q_i);$ } }



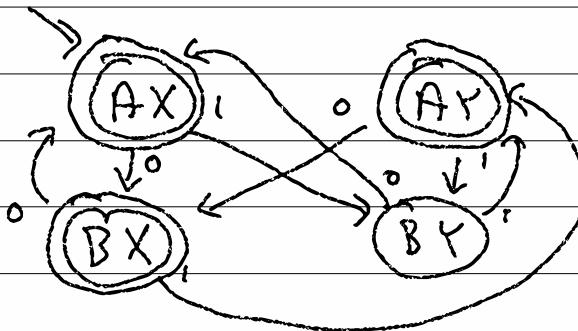
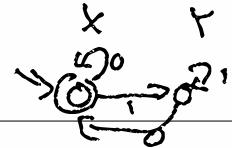
$A \cup B \leftarrow \text{DFA}$ (if $A \in \text{DFA}$ and $B \in \text{DFA}$)



2-5 Even Len



Is Even



0	✓
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:

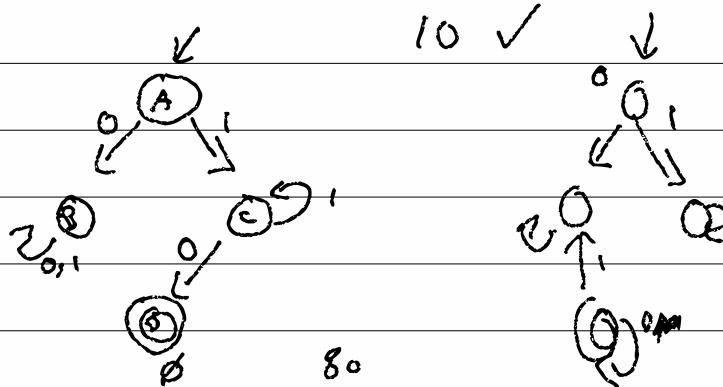
Q : Sstate \Rightarrow Bool

Σ : 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.'

$\Sigma = \{D\}$

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

$[D]$

$D \Rightarrow C, 0$

$\Sigma = \{E\}$

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

$[E]$

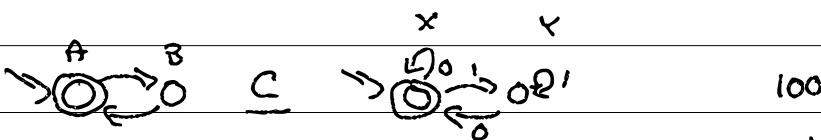
or, if not, No!

3-2/ subset

$A \subseteq B \iff \forall x \in U. x \in A \Rightarrow x \in B$

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

finite means naive works!



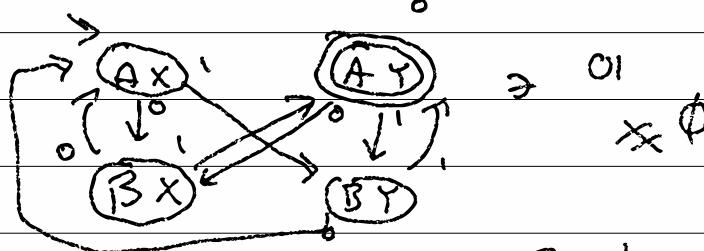
$$\boxed{\bar{A}} \subseteq \boxed{\bar{B}} \cap \boxed{\bar{A}} = \boxed{\bar{A}}$$

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

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
 $\text{model} = \text{theory}$

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

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

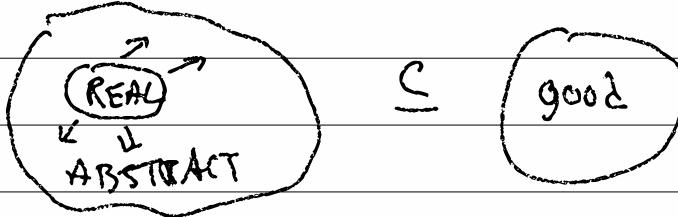
if $x > 0$ then

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

o.w.

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

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



<u>3-w)</u>	Finite	=	\emptyset	Σ^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
 $x \in A \wedge y \in B$

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

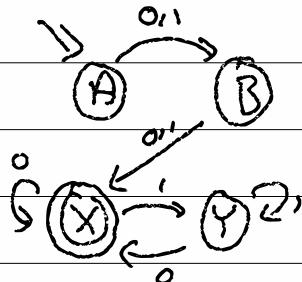
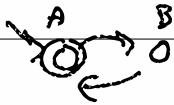
$$abcd = ab \circ cd$$

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

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

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

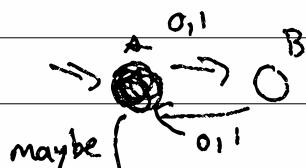
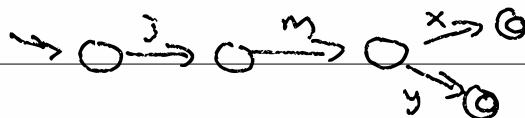
3-5/ Even Len \circ Even Num



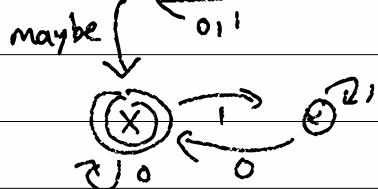
00110 ✓
0011 X

~~00110011~~

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

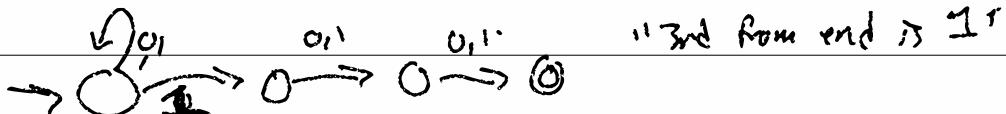


$\Sigma = \{0, 1\}$



$x \in \text{DFA}$ 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^* $\wedge B \in \text{DFA}$

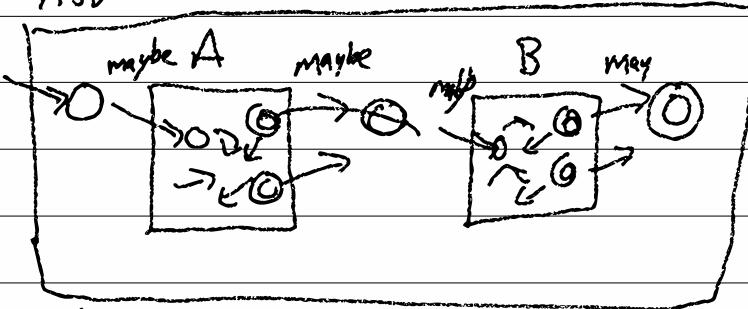
NFA ($N - \underline{\text{non}} \text{ D-deterministic}$)

$$\downarrow \qquad \qquad \qquad \downarrow$$

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

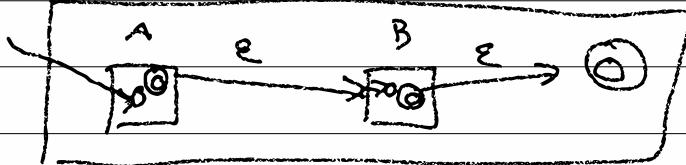
$$S: Q \times \Sigma \rightarrow Q$$

$A \circ B$



$A \circ B$

What does (NFA)
this mean?



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

4-2] what do NFAs mean?

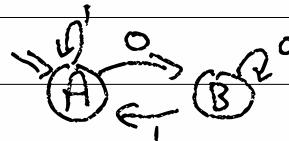
A DFA represents a set and
a set "is" a membership function

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

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

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

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



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

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

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

$$0A1A1A0B \rightarrow N$$

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

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

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

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

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

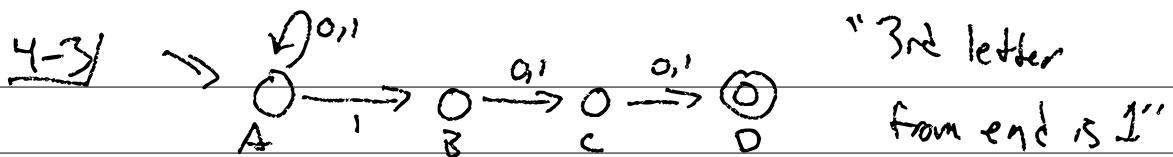
$$\text{Nvalid } g; \epsilon = Y$$

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

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

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

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



0 1 0 0	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 $\exists t \in \text{traces. } \text{str}(t) = w$.

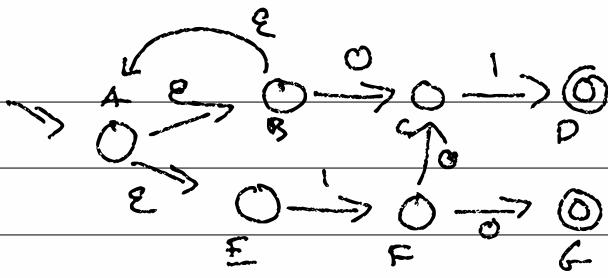
valid $\Leftrightarrow t \in F$
and last-state(t) $\in F$

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

figure all possible traces

check if valid and if strings match

check if past is in ϵF



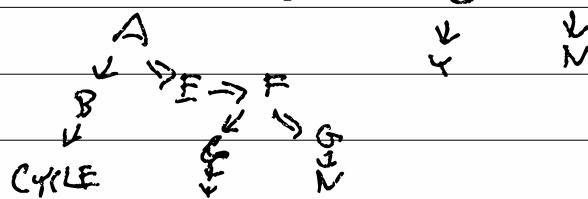
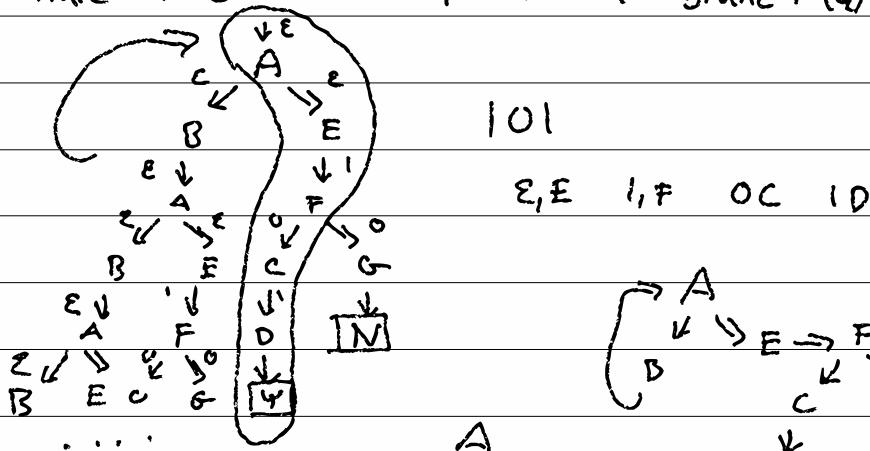
$(\epsilon, B)(0, C)(1, D)$ 01 = 0001

$(\epsilon, E)(1, F)(0, C)(1, D)$ 101 = 01001

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

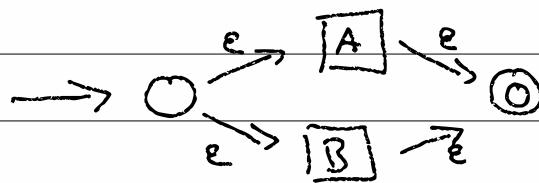
$(\epsilon, B)(\epsilon, A) \times$ where \times is valid
 \rightarrow valid

Trace Tree = T | N | Branch (ϵ, Q) (List TTI)



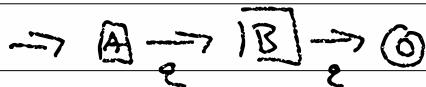
Forking model of NFAs (make TT)
 Backtracking model (explores TT)

4-5) $A \cup B$



$x \in A$
 $0 \rightarrow \square$
State X transitions
to THE start

$A \circ B$

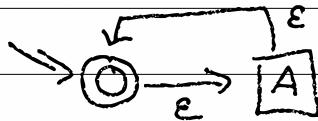


$\square \rightarrow 0$

All accepting states

of A transition to 0

A^*

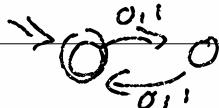


4-6) $\forall A$, $A \in \text{DFA} \Leftrightarrow A \in \text{NFA}$

\Rightarrow

\Leftarrow

DFA \Rightarrow NFA



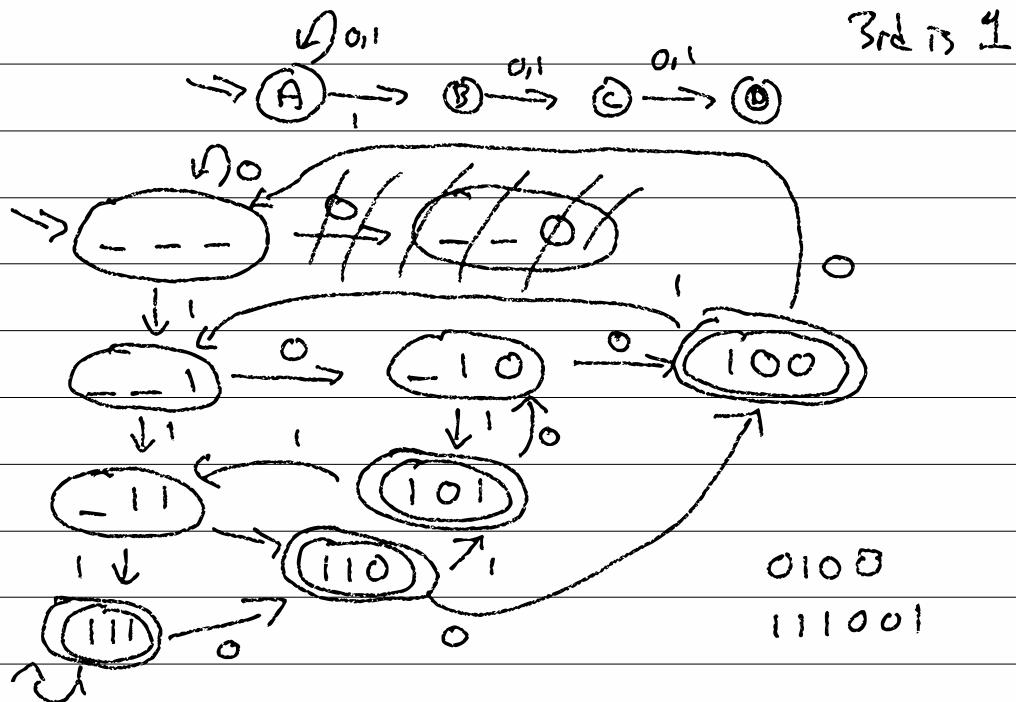
DFA $\delta: Q \times \Sigma \rightarrow Q$

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

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

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

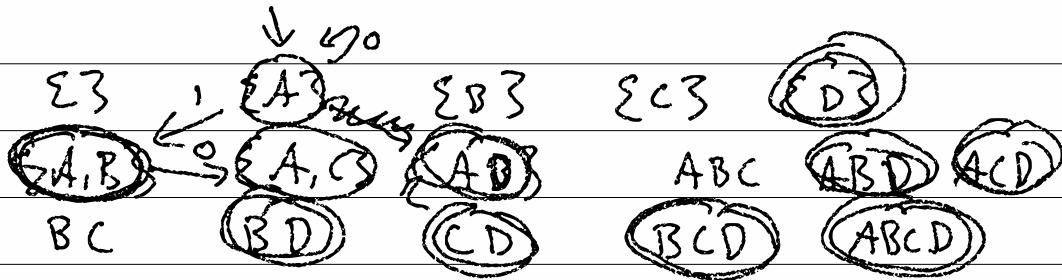
NFA \Rightarrow DFA



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

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

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



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

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

$$\delta'(\{q_1, \dots, q_n\}, c) = \bigcup \delta(q_i, c)$$

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

$$\delta' : \overbrace{Q_A \times Q_B}^{\Sigma} \times \Sigma \rightarrow Q$$

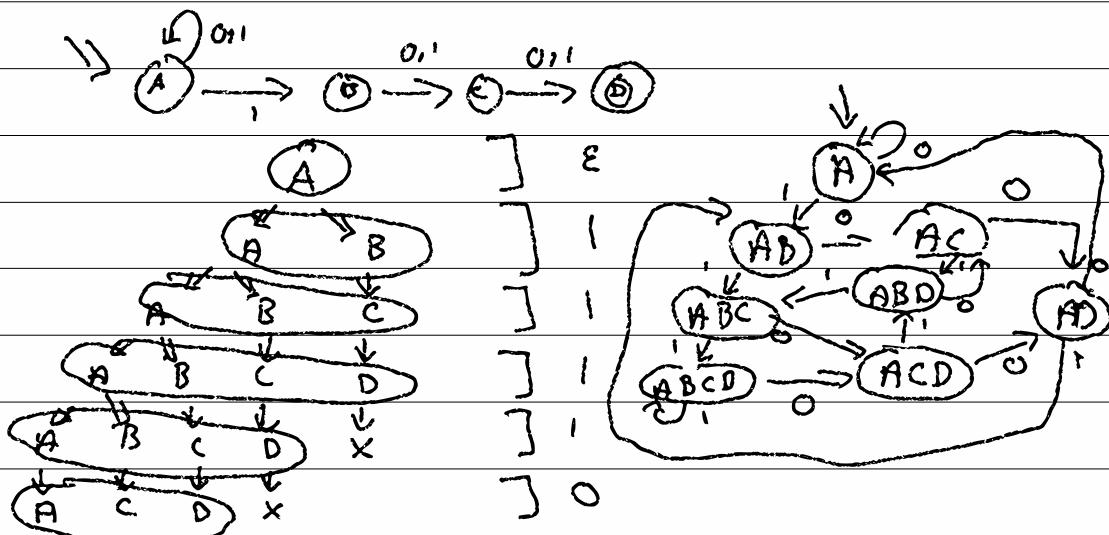
$$\delta'((q_a, q_b), c) = (\delta_A(q_a, c), \delta_B(q_b, c))$$

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

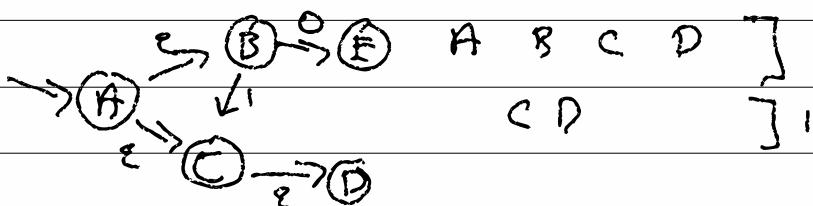
$(Q, \Sigma, q_0 \in Q,$
 $\delta: Q \times \Sigma \rightarrow P(Q), \Rightarrow \delta': Q' \times \Sigma \rightarrow Q',$
 $F \subseteq Q)$ $(Q', \Sigma, q'_0 \in Q',$
 $F' \subseteq Q')$



$\ll \text{vw}$
 $\begin{matrix} A & B \\ \downarrow & \downarrow \\ B & C \end{matrix}]^0$

$\begin{matrix} A \\ \downarrow \\ A \end{matrix}]^0$

$\begin{matrix} A & C & C & A \\ \downarrow & \downarrow & \downarrow & \downarrow \\ BAAH & BAAH & BAAH & BAAH \end{matrix}]^x$



5-3/

$$E: Q' \rightarrow Q' \quad P(Q) \xrightarrow{P(Q)} - \text{follow all C-transitions}$$

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
 $= E(\{\}) \in P(Q)$

δ' = 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
= $\{Q_i \mid \underbrace{Q_i \subseteq Q \text{ and } Q_i \cap F \neq \emptyset}\}_{Q_i \in P(Q) = Q'}$

= (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

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

$E^*(A)$

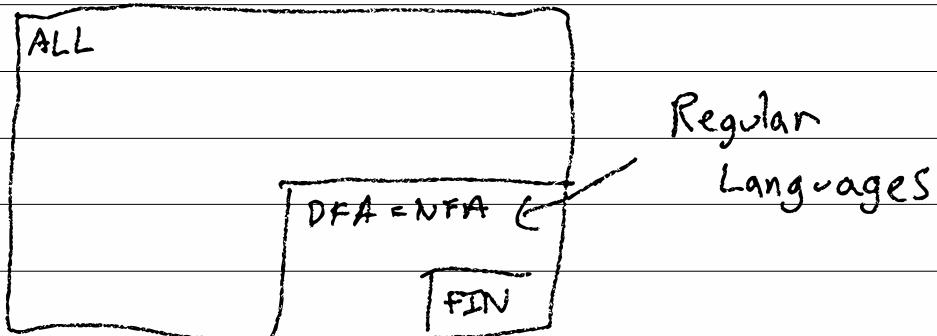
$E^*(A) = A \cup \bigcup_{g_i \in A} \delta(g_i, \varepsilon)$

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

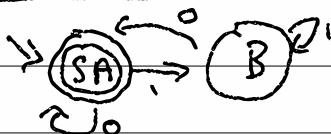
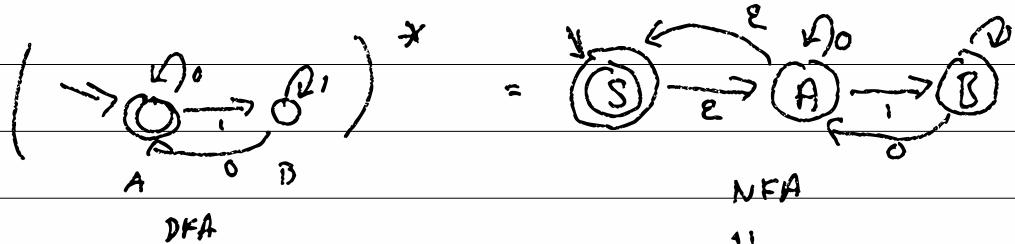
$$N = D$$

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

$$D = N$$



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



6-2 | Regular Expressions

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

interface Register { }

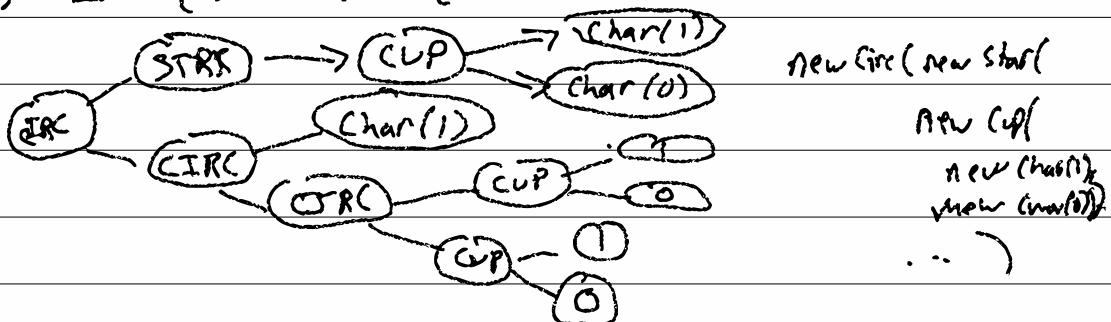
class RE_<empty> impl REGEX { . . . }

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 is F"



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

fix the language if doesn't
(or come)

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

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

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

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

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

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

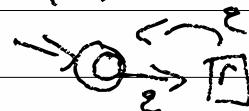
class Up {
 $L()$ \in
 return up(lhs, L(), rhs, L());
}

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

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

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

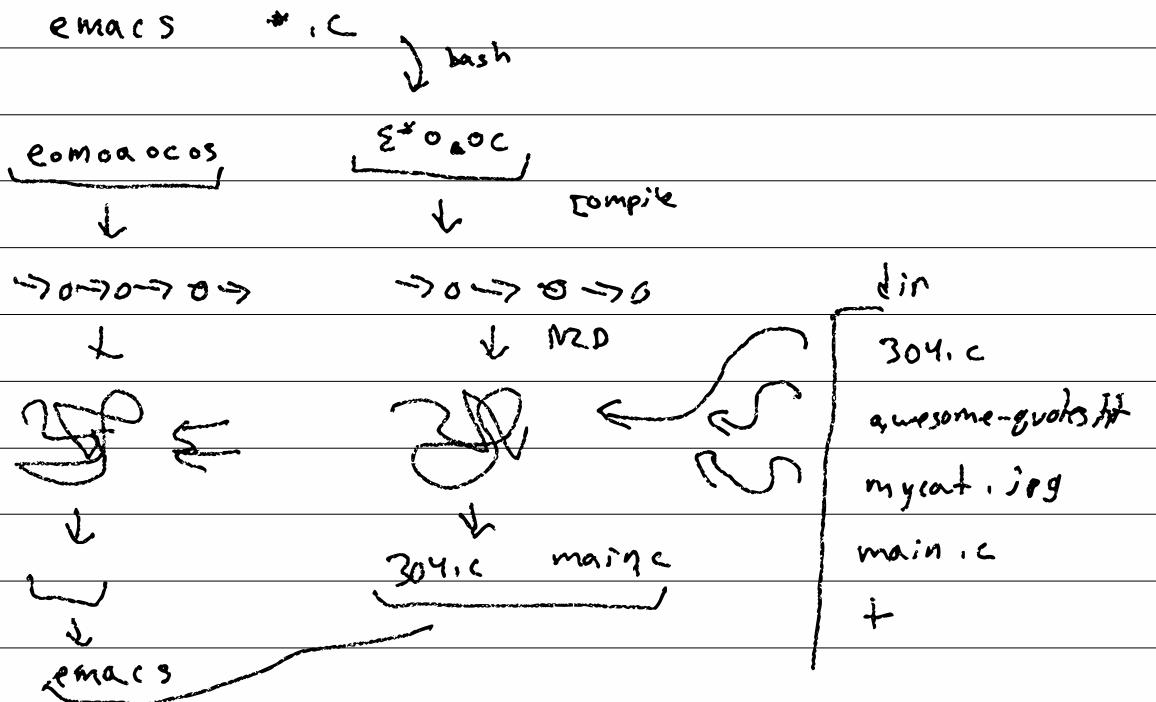
$$\text{size } c = 1$$

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

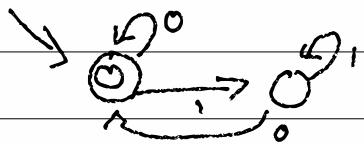
$$\text{size } (x \circ y) = \text{sz } x + \text{sz } y + 1$$

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

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

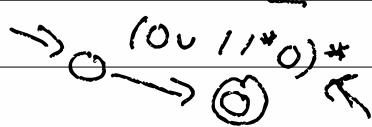
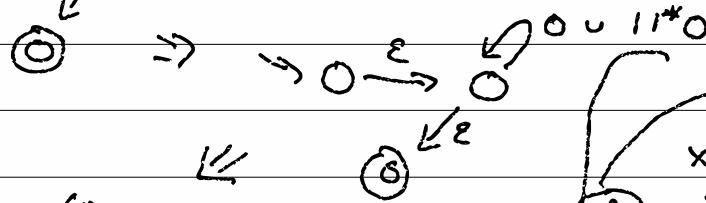
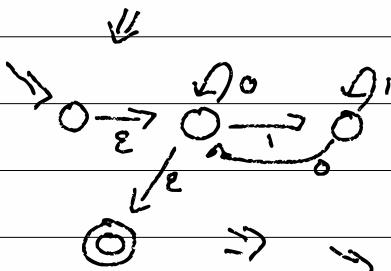


6-8] decompire : DFA $\xrightarrow{(\text{NFA}) \text{ or }} \text{RE}$



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

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

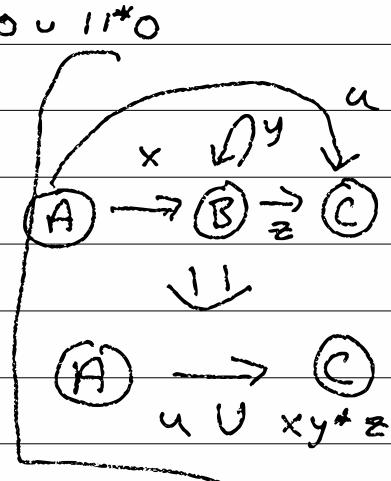


$$(0 \cup 11^*0)^*$$

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

011101111100

11100011101100101010



$$A \xrightarrow{u} C$$

$$u \cup xy^*z$$

G-9 / decompile : N-state NFA
→ RE

START : N-NFA → (N+2)-GNFA

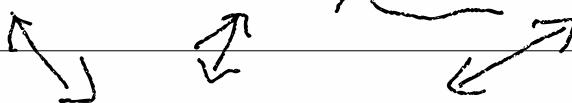
RIP : (N+1)-GNFA → N-GNFA

END : 2-GNFA → RE

decompile m = end ∘ ripⁿ ∘ start (n)

7-1) DFA/NFAs \rightarrow RE

DFA_s \leftrightarrow NFA_s \leftrightarrow RE



Regular
Languages

NFA \rightarrow RE

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

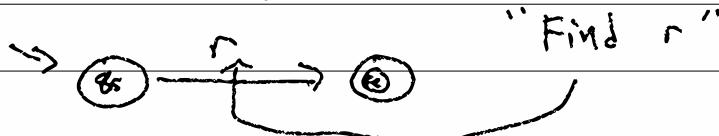
RIPⁿ: (n+1)-GNFA \Rightarrow n-GNFA

OUT: 2-GNFA \rightarrow RE

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

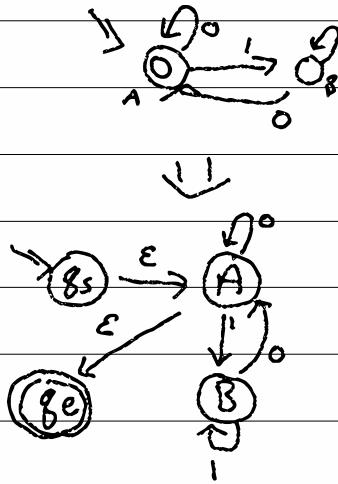
OUT: 2-GNFA \rightarrow RE

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



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

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



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

$$\Delta(g_s, g_o) = \epsilon$$



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

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

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

$$\Delta(g_i, g_j) = \cup \{\epsilon_{\text{for } g_e}\}$$

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

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

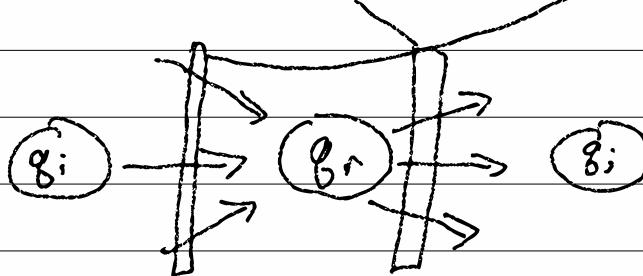
↓
 jay
 main

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

↓
 jay
 main
 ↓
 exit
 ↓
 exit

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

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

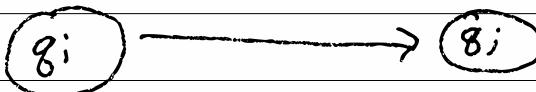


$$x \circ y^* \circ z \cup \alpha$$

$$\rightarrow x \circ z$$

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

$$=$$

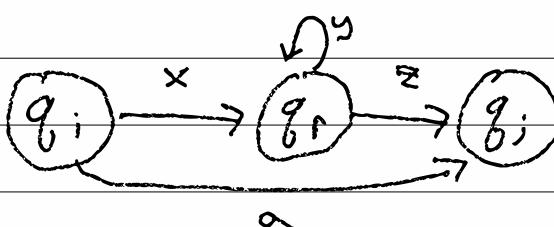


$$x = \Delta(q_i, q_r)$$

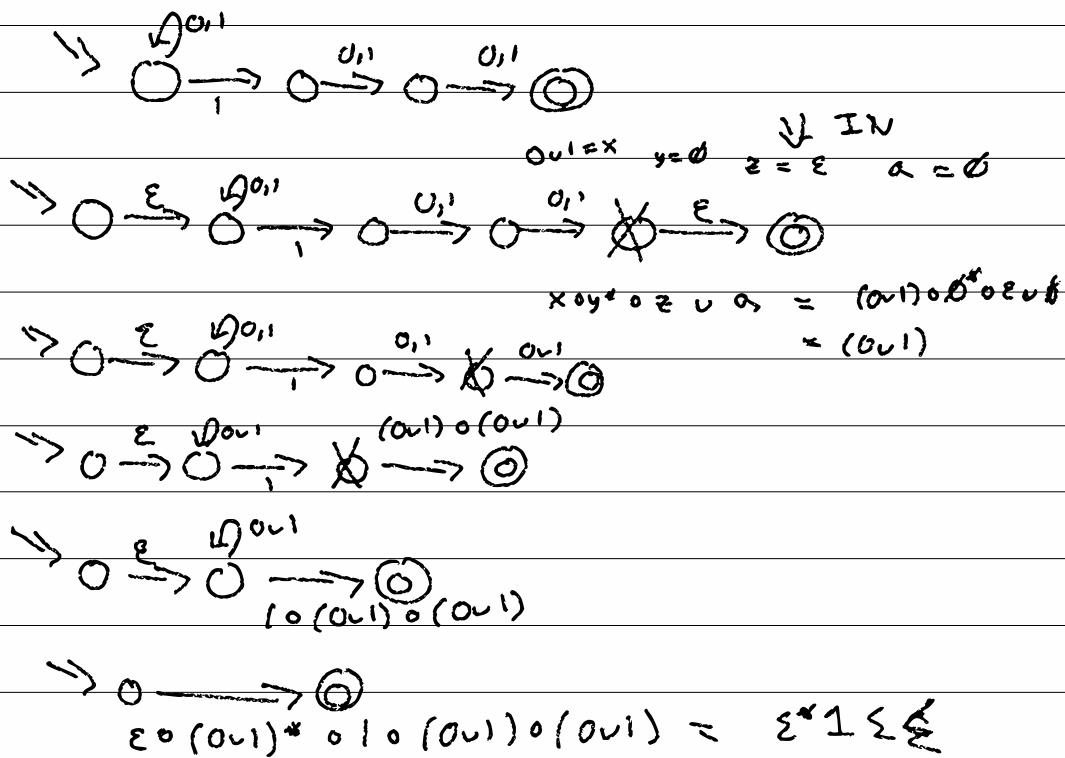
$$y = \Delta(q_r, q_r)^*$$

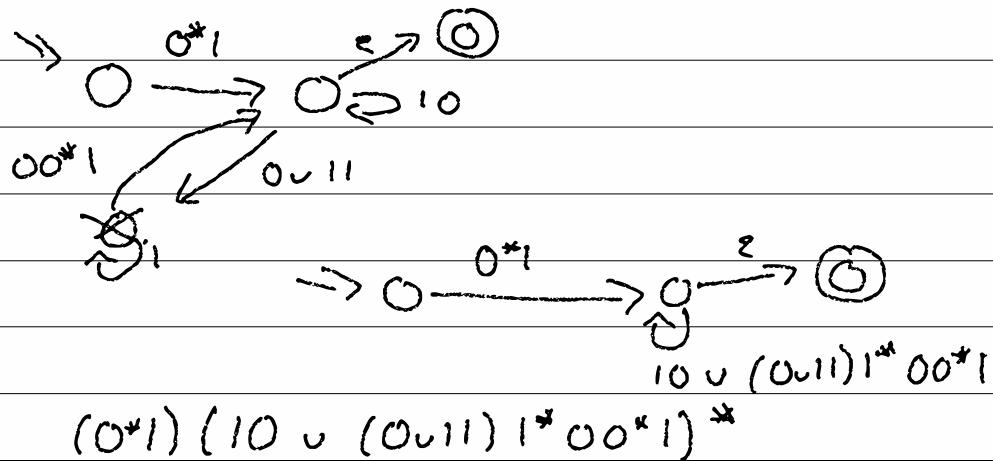
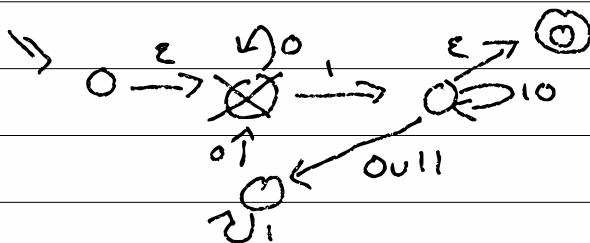
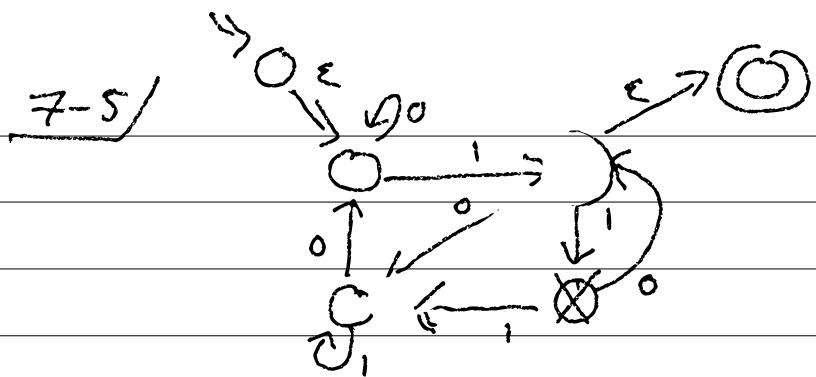
$$z = \Delta(q_r, q_j)$$

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



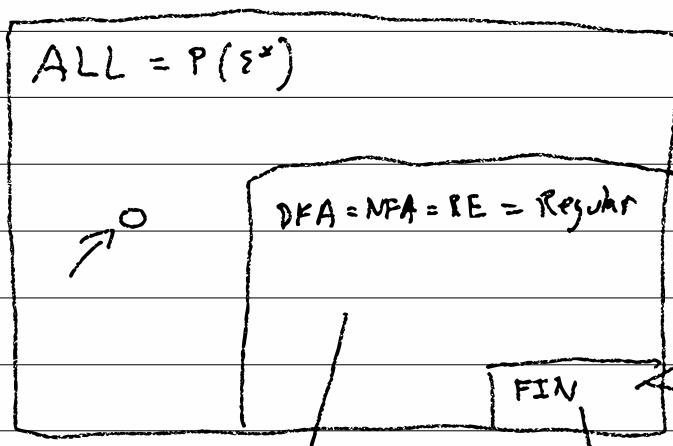
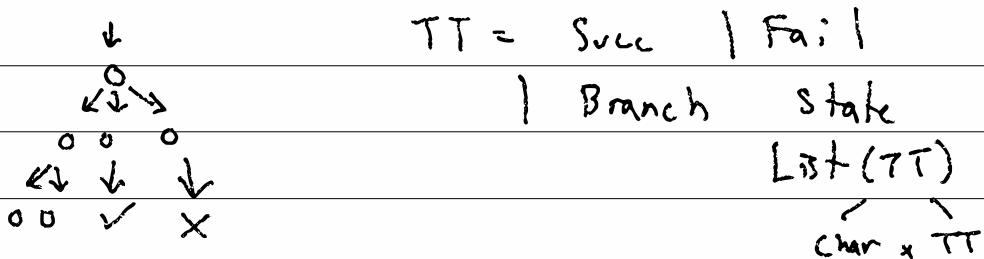
~~7-4/ int main () {~~
 int ~~x = 8;~~
~~int y = f(x, z);~~
 ...
 int y = x + 2 * 2;
 ↓
 int f(int ~~z~~){
 return ~~z + 2 * 8;}~~





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

TT



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

$\{ \epsilon, 00 \}$

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

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

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

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

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 $\exists B \in \text{ALL}, \neg P(B)$

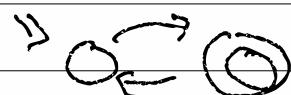
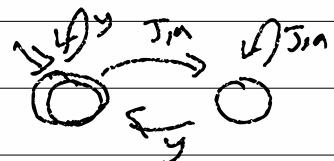
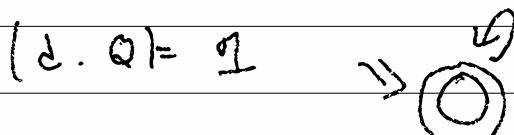
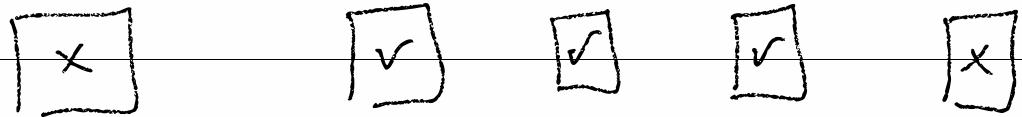
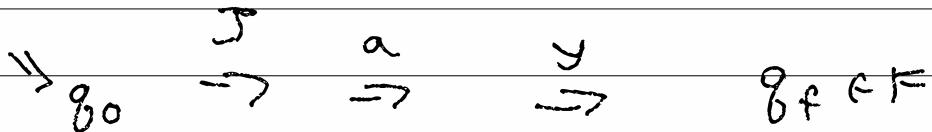
conclude $B \notin \text{REG}$.

$$\Sigma = \dots$$

8-3)

"Jay" & d

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



Daphne wins

pick a number of states ; 4

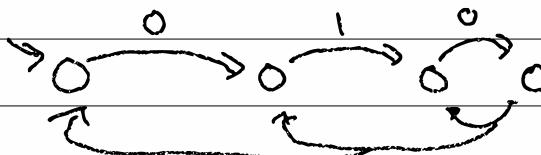
she picks a char

I say what state we goto

I win if I never say same state

she wins if I repeat

How many turns to win?

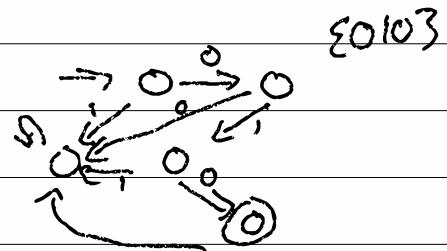
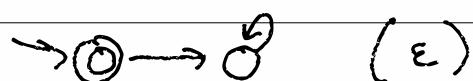


$4 \Rightarrow N \Rightarrow$

|@|

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

All DFAs have a loop



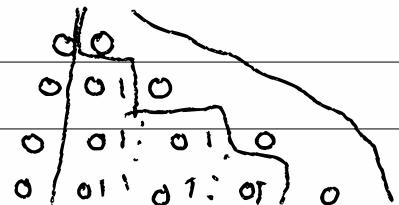
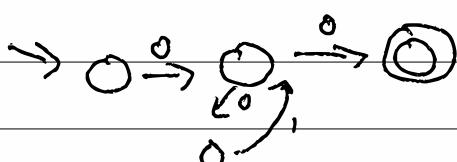
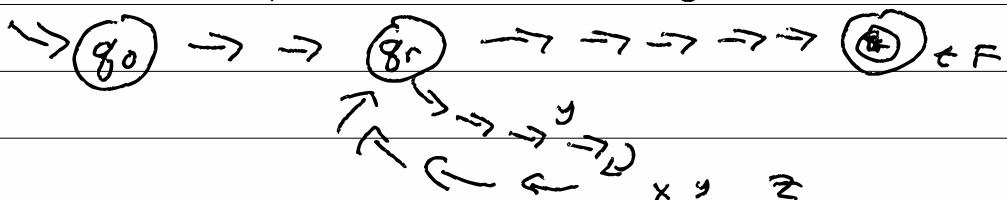
Some have "exciting" loops

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



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

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



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

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

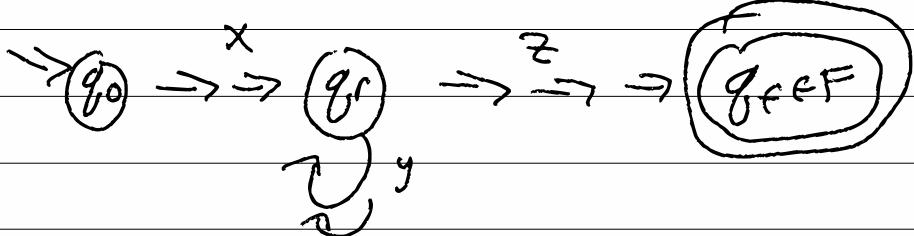
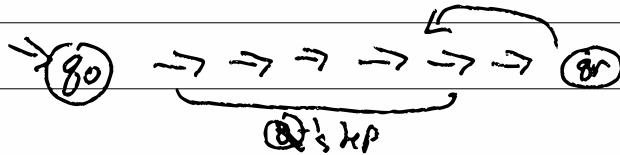
If it does have one ...

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_+
```

$\forall n \in N,$

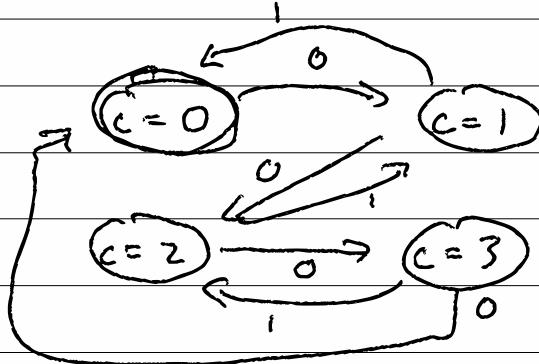
$0^n 1^n \in B$

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 = $P(\Sigma^*) \approx A\bar{A}$

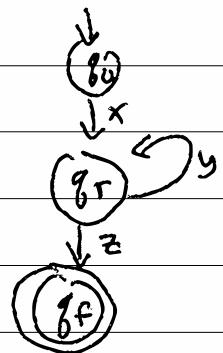
$$\exists p \in N, \quad |P| = |Q|$$

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

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

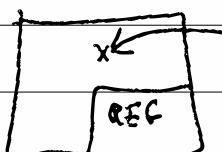
$$\forall i \in N, \quad |xy| < p$$

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



$\neg RPP(A) =$

ALL



$$\forall p \in N,$$

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

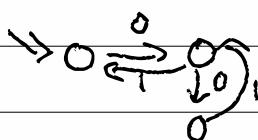
$$\forall (x, y, z \in \Sigma^*) \mid s = xyz \wedge |y| < p$$

$$\exists i \in N$$

$$xyiz \notin A$$

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

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

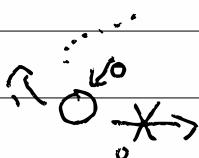


$$\text{int count} = 0$$

while (see 0) count++

while (see 1) count--

return 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 $s: s = x = z$

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

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

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

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

$|y| > 0 \quad)$

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

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

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

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

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

$a+b+c = d$

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

fun $n:$

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

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

else: let $m = n-1$

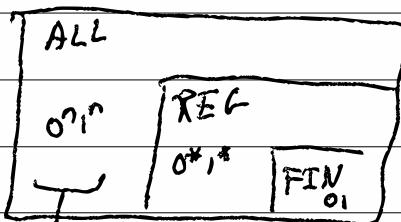
$p \circ : \text{th. } p_n \rightarrow p_{(n+1)}$

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?



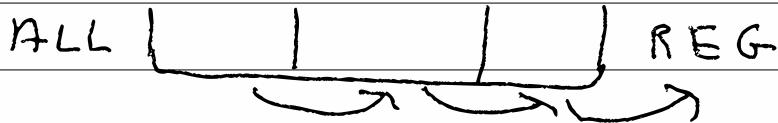
All computers
are DFAs.

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

~~then~~^{and} $B \in \text{REG}$

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 $\in \text{REG}$

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.

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

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

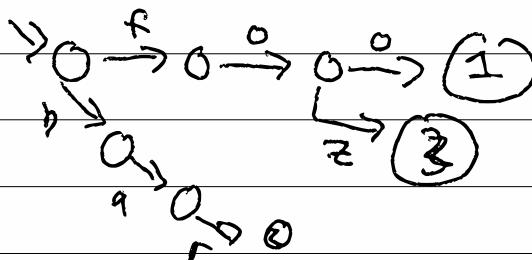
$$\exists ; . \quad \begin{matrix} \text{change this} \\ \text{can't change} \end{matrix}$$

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

$$\Rightarrow \Rightarrow$$

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

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 $S \rightarrow OSI$

variables
 non-terminals
 Symbols

rule = $\overbrace{V}^{\text{lhs}} \rightarrow \overbrace{V \cup \epsilon}^{\text{rhs}} \ast$
 $\text{rhs} = (\text{V} \cup \epsilon)^*$
 $\xrightarrow{\text{terminal}}$

$\overset{B}{S} \rightarrow \overset{B}{OSI} \rightarrow \overset{B}{0OSII} \rightarrow \overset{B}{000SIII}$
 "derivation" $\overset{B}{000III}$

$w \in L$ 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 $\downarrow \in V$

$$B \Rightarrow \epsilon$$

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

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

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

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

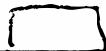
$$R = \{S, SS\} \quad V = \{\Sigma\} \quad \Sigma = \{0, 1\} \quad S = S$$

REG

DFA

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

CFL



CFG

12-3 / A B

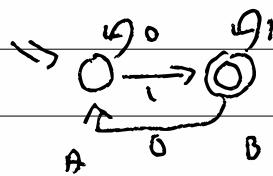
$\rightarrow A \cup B : S \Rightarrow A, S$

$S \Rightarrow B, S$

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

$A \cap B : \times$

DFA \Rightarrow CFG



01101

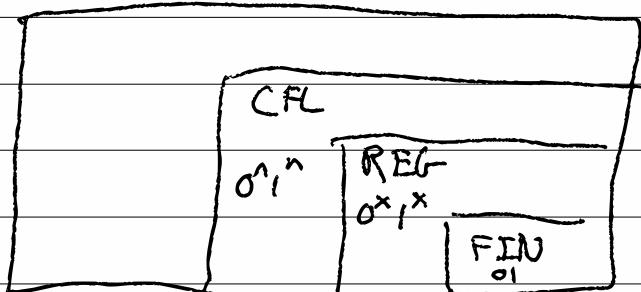
$S = g_0$

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

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

x	\times	γ
$A \rightarrow 0A$		$1B$
$B \rightarrow 1B$		$0A$
\vdash	w	ε

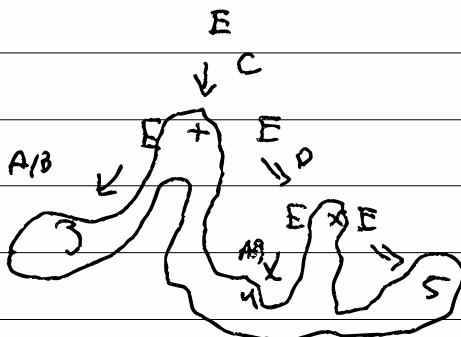
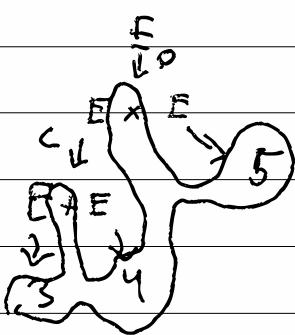
$$A \xrightarrow{x} 0A \xrightarrow{\gamma} 01B \xrightarrow{\varepsilon} 011B$$

$$\xrightarrow{\omega} 0110A \xrightarrow{\gamma} 01101B \xrightarrow{v} 0m$$


P D C D

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

$$3 + 4 \times 5$$



$$\text{fowim } E \Rightarrow n$$

$$\text{fowin } O = O$$

$$I = I$$

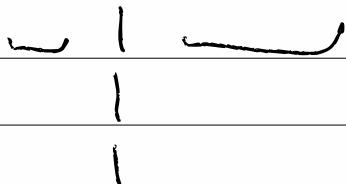
$$\text{Plus}(L, R) \Leftarrow \text{fowim } L + \text{fowim } R$$

$$\text{Mult}(L, R) \Leftarrow \text{fowim } L \times \text{fowim } R$$

ambiguous = There are multiple
trees (derivations)

if cond

for the same string



(2-5) Nice! amb \rightarrow unamb

not possible

V, Σ, R, S

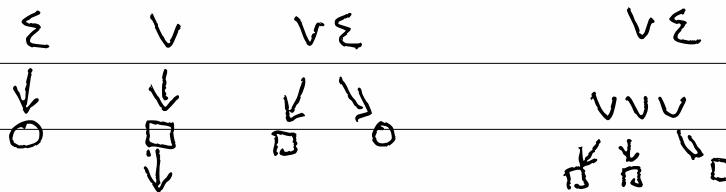
$$V \Rightarrow (V \cup \Sigma)^*$$

ϵ

$\epsilon \Sigma V$

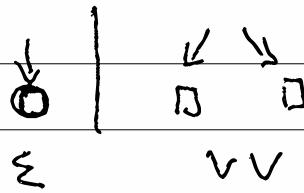
V

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 ϵ -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 ≥ 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 + 0ⁿ1ⁿ

S $\Rightarrow \epsilon$

S $\Rightarrow 0S1$

A

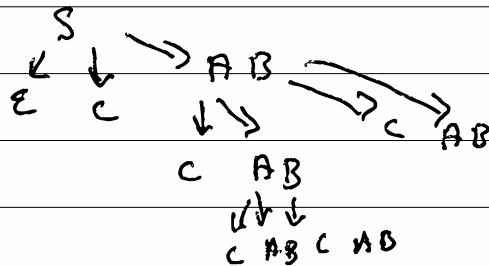
B

A $\Rightarrow c$ + 1

B \Rightarrow + 2

A $\Rightarrow BC$ + 2

A \Rightarrow done



S $\Rightarrow XYZ$

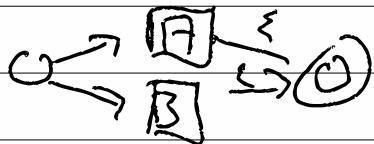
X $\Rightarrow ZZS$ | 0

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

$$\delta(\text{start}, \epsilon) = \{ \text{fromA } A_0, \text{ fromB } B_0 \}$$

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

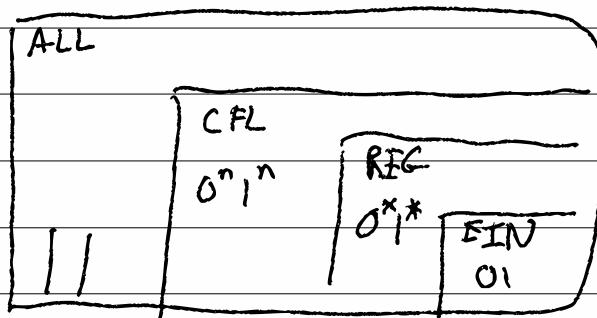
12-7 / new DFA ("blat",
 $q_i \rightarrow q_i = 0 \text{ if } q_i = 1,$
sigma, delta,
 F)



complement (DFA \perp) Σ

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

(5-1)



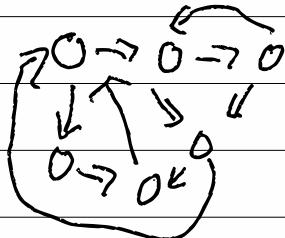
REGULAR

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

CFL

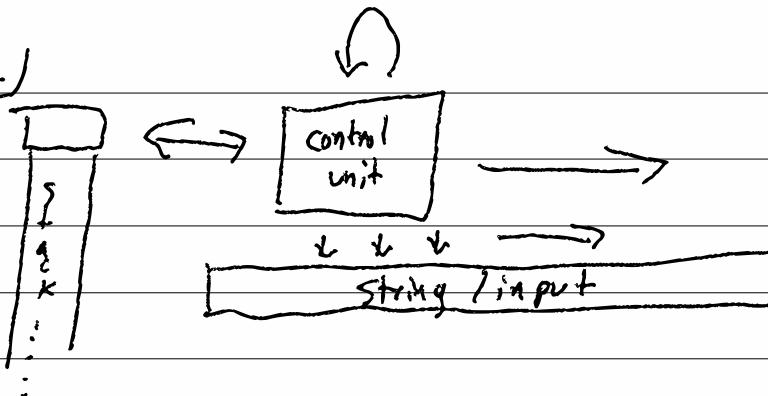
$$\boxed{\text{PDA}} \longleftrightarrow \text{CFG}$$

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

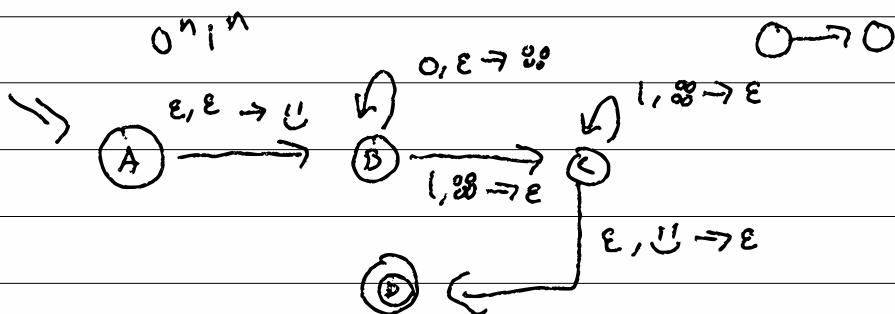
let $(q_j, \alpha) \in \delta(q_i, c, \alpha)$ in
 $(\alpha = \beta, q_j, w)$

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

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

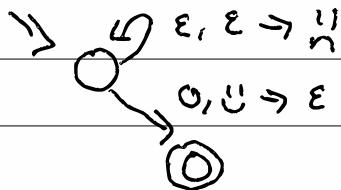
return $\left(\begin{array}{l} \text{sim } q_f \in \\ c, q_j \in p, c \end{array} \right)$
 where $c_0 = (\Sigma, p, q_0, S)$

$c, \alpha \Rightarrow \beta$



$\epsilon[A]0011 \rightarrow U[B]0011 \rightarrow U^00[B]011 \rightarrow U^0000[B]11$
 $\rightarrow U^00[C]1 \rightarrow U[C]\epsilon \rightarrow \epsilon[D]\epsilon \rightarrow \text{YES}$

15-4/



CFG \rightarrow PDA

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

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

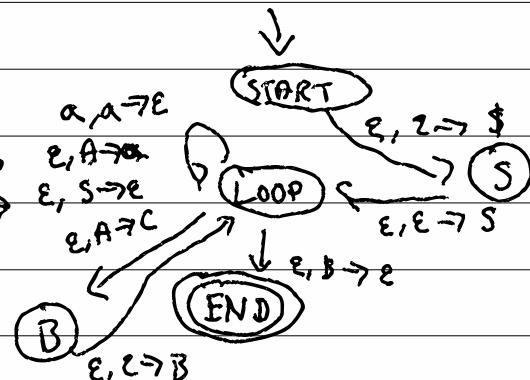
CNF \rightarrow PDA

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

$$S \Rightarrow \epsilon$$

$$A \Rightarrow a$$

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



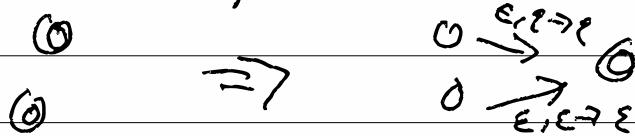
$$\$[L]0011 \rightarrow \$[S]0011 \rightarrow \$S[L]0011 \rightarrow \$ISO[L]0011$$

$$\begin{array}{ccccccc} \$1\$[L]0011 & \leftarrow & \$1ISO[L]0011 & \leftarrow & \$1\$O[L]0011 \\ \swarrow & & & & \downarrow \\ \$11[L]0011 & \rightarrow & \$1[L]0011 & \rightarrow & [END] \rightarrow \checkmark \end{array}$$

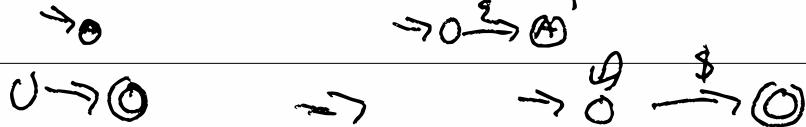
15-5/ PDAs are too complicated

so, we'll simplify with some rules

- Make a single accept



- Guarantee stack is empty on accept



- Always push on pop

push : ϵ , Γ

pop : Γ , ϵ

X ignore : ϵ , ϵ

X replace : Γ , Γ

Every symbol pushed is eventually popped

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

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

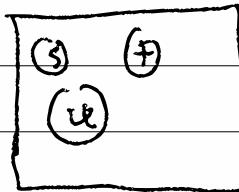
$$(g, \varepsilon) \in (s, b, +)$$

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

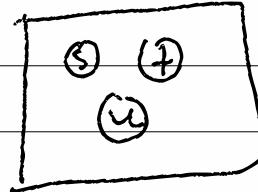
$$(p, p) \rightarrow \varepsilon$$

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

A \leftarrow cast



B \leftarrow bst



shown cast, bst

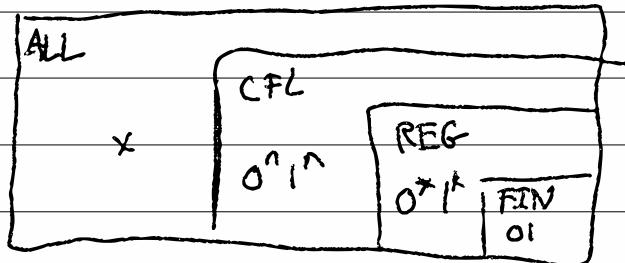
A \cup B \leftarrow ~~cast, bst~~

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

| and A x

| and B y

16-1



DFA_s \longleftrightarrow REG

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

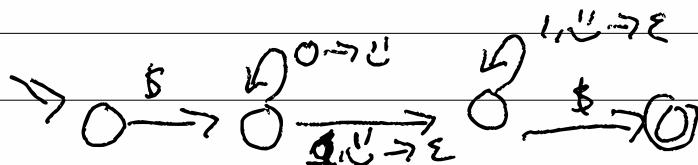
PDA_s \longleftrightarrow CFG

$x \notin \text{CFL}$

CFL = ALL

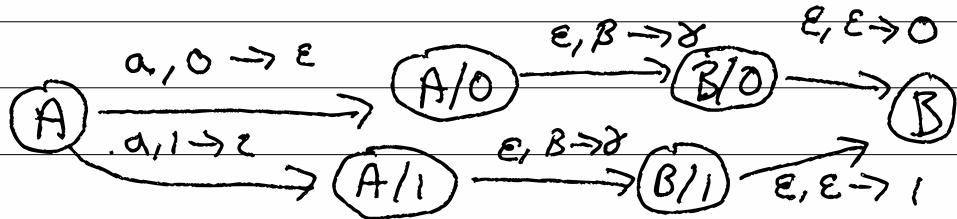
1.	$P \dots \forall x \in \text{REG}, P(x)$	$\forall c \in \text{CFL}, P(c)$
2.	$\exists x \in \text{ALL}, \neg P(x)$	$\exists x \in \text{ALL}, \neg P(x)$
3.	$\Rightarrow x \notin \text{REG}$	$\Rightarrow x \notin \text{CFL}$

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



16-2 Look at 2nd th, by

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



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

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

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

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

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

16-3/ RPP

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

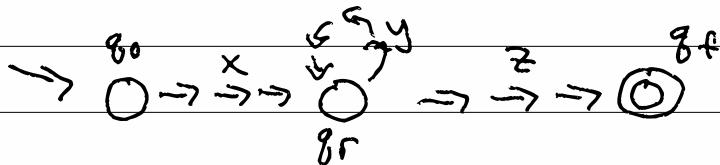
$\exists p \in \mathbb{N},$

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

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

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

$xy^iz \in A.$

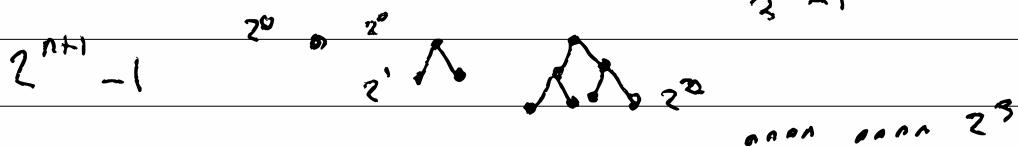


CFG G (CNF) = (V, Σ, R, S) $S \Rightarrow \epsilon \in R$

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

$v_i \Rightarrow v_x v_y$

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	



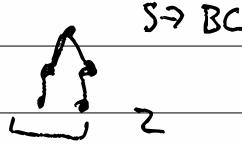
(6-y) minimum length of tree of height n

$S \rightarrow E$

$$\bullet = 0$$

$S \rightarrow \alpha$

$$= 1$$



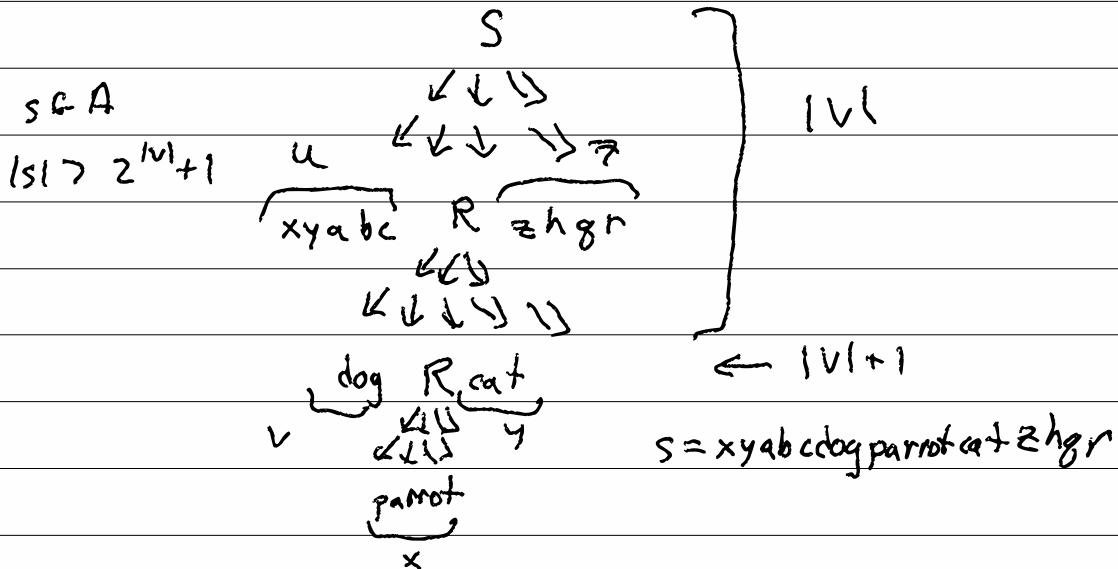
$S \rightarrow BC$

$$n=4$$

$$\Rightarrow$$

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

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

$$\forall A \in \text{CFL}, \quad \exists p \in \mathbb{N}, \quad \forall s \in A \quad |s| \geq p \quad \exists (u, v, x, y, z \in \Sigma^*) \quad |s = uvxyz| \quad |vxy| \geq p \quad |vy|^i > 0$$

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

$$uv^ixy^iz \in A$$

$$\begin{array}{ll} S \Rightarrow uRz & S \Rightarrow uRz \rightarrow uxz \\ R \Rightarrow vRy & S \Rightarrow uRz \rightarrow uvRyz \rightarrow uv^2Ry^2z \\ R \Rightarrow x & \rightarrow uv^2xy^2z \quad (i=2) \end{array}$$

$$0^n, n \in \mathbb{N} \in \text{CFL} \quad \overset{p=4}{\sim} \quad s = 0011 \quad |vxy| = 2 \leq 4$$
$$u=0 \quad v=0 \quad x=\epsilon \quad y=1 \quad z=1 \quad |vy|=2>0$$
$$uv^ixy^iz \in A? \quad 00^{i+1}1^{i+1} = 0^{i+1}1^{i+1} \in A$$

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

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

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

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

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

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

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

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

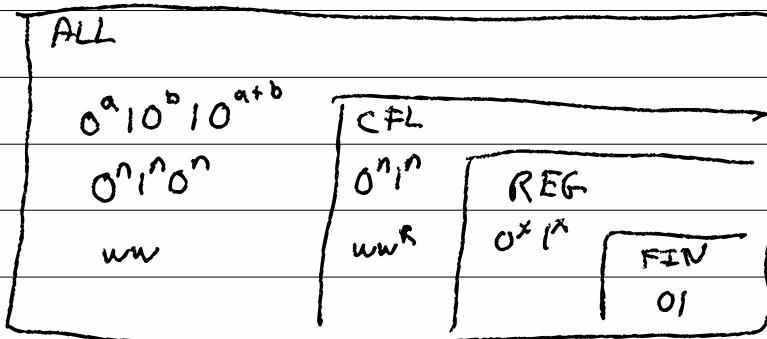
DFAunion = pair $\langle X, Y \rangle$

NFAunion = start

state from left X ("x")

state from right Y ("y")

END



17-1 Finite : 01

Regular: $0^* 1^*$

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

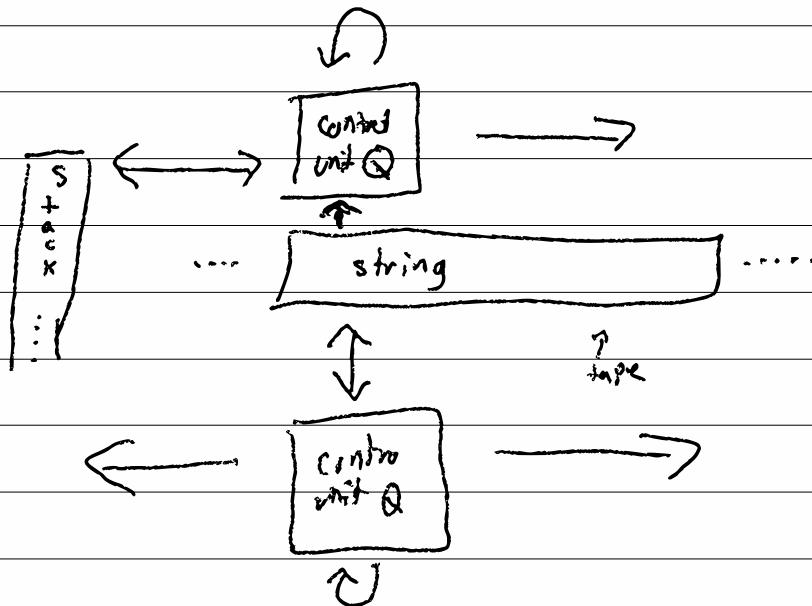
Context-free pumping property
CFPP

context-free pumping property

\Rightarrow CFG, PDS

Turing Machine Alan Turing

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: Σ

$\sqsubset \in \Sigma$

config:

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

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

$q_0 \in Q$

$c_0 = \varepsilon[q_0] \sqsubset$

$q_a \in Q$ (accepting)

for $w \in \Sigma^*$ input

$q_r \in Q$ (rejecting)

$w \in L(m)$ iff

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

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

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

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

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

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

left: tape \rightarrow tape

right: tape \rightarrow tape

left $\varepsilon = \varepsilon : \sqsubset$

right $\varepsilon = \sqsubset : \varepsilon$

left $(x : c) = x$

right $(c : y) = y$

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

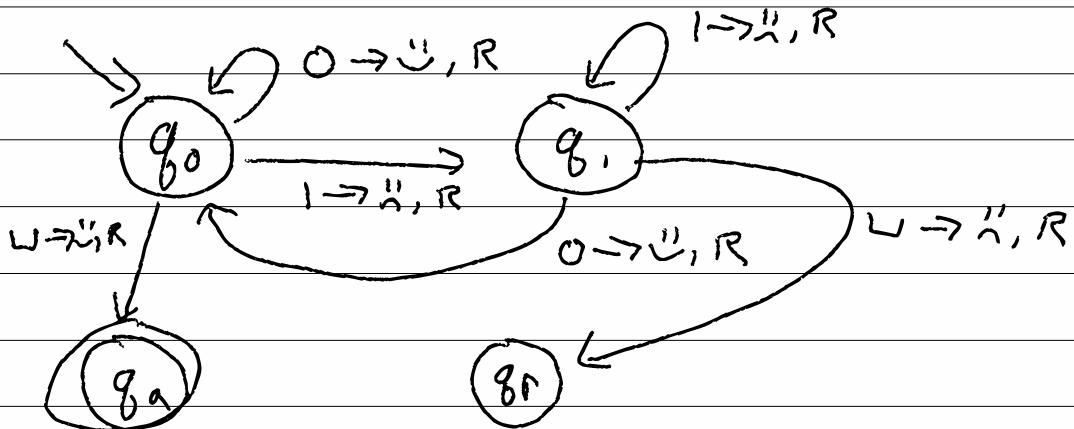
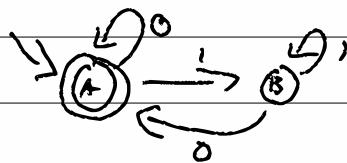
$011[q_f]101$



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

(DFA)

173)



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

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

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, ω, R) if $q_i \notin F$

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

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

I saw a zero

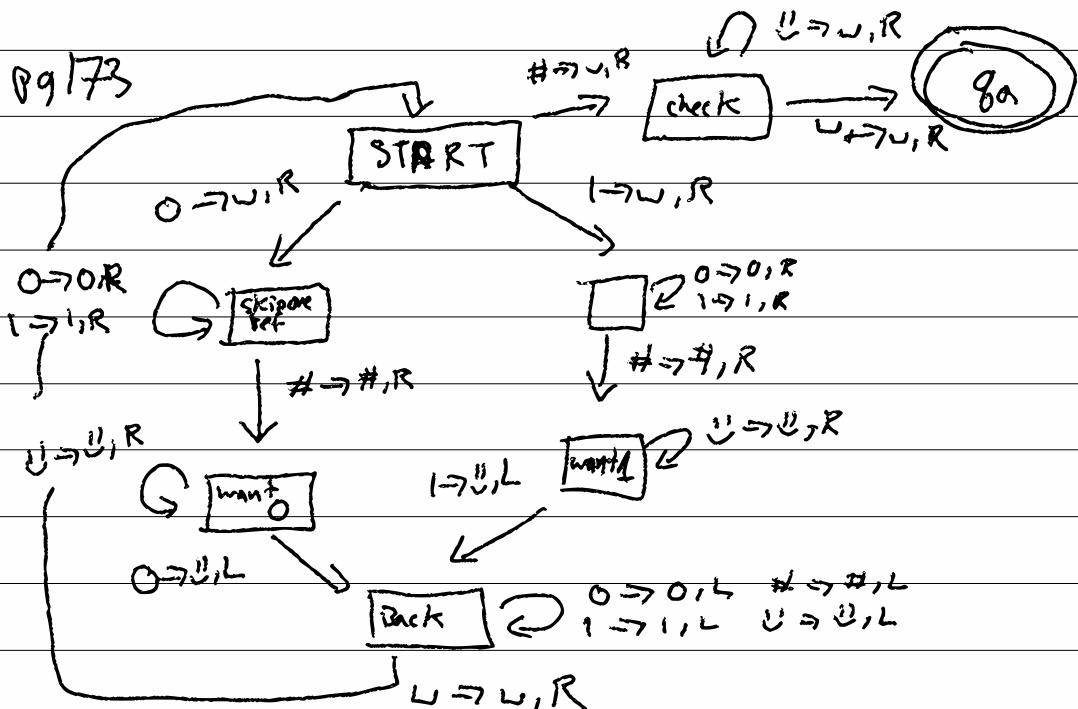
$[] 01 \# 01 \rightarrow \sim [] 1 \# 01 \rightarrow \sim 1 [] \# 01$

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

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

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

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



17-5/ A computable function
 f

is a Turing - machine
and

$$f(x) = y$$

if f

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

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

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

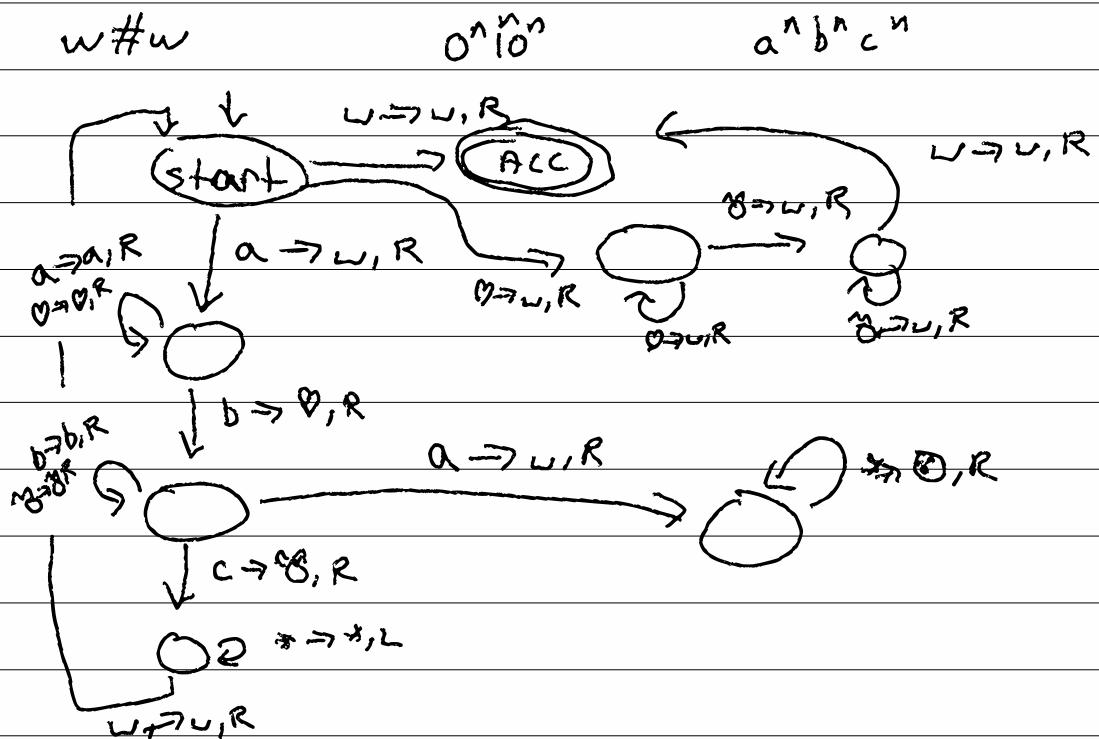
$$\begin{matrix} 10 \end{matrix} = \begin{matrix} 11 \end{matrix}$$

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

($\geq \Sigma$)

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

$Q = \{q_0, q_f\}$



$$0^x + 0^y = 0^{x+y}$$

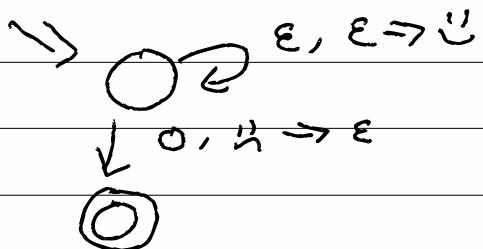
$$f(0^x + 0^y) = 0^{x+y}$$

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

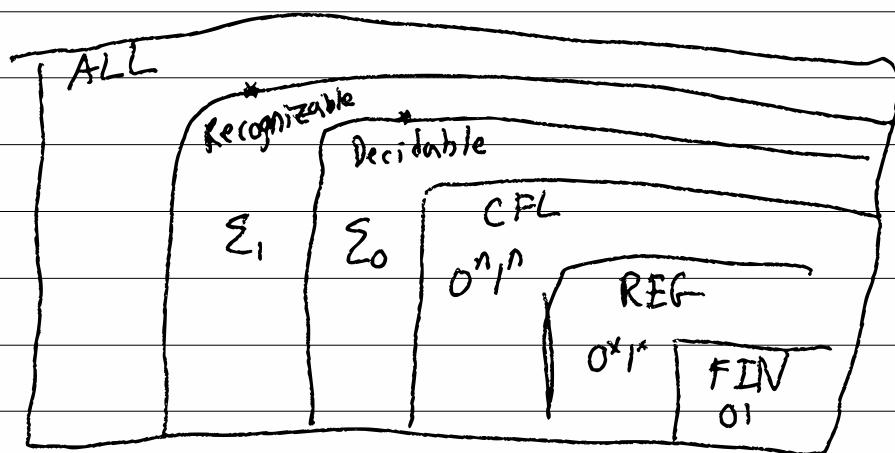
18-3 / A TM is either a

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_s

REX_s

PDA_s

CFG_s

TM_s

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}

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

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

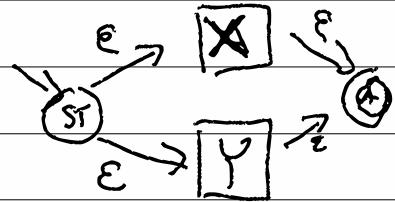
\Rightarrow Pair $\langle Q_x, Q_y \rangle$

$(Q_x \times Q_y, \Sigma)$

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

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

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



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

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

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

$$c = \emptyset$$

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

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

(if $q_x \in F_x$ then Acc) 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)