

L-1 effective math $\sum_{i=0}^{20} z_i \text{ vs } \sum_{i=0}^{\infty} z_i$

$$9x^3y^2z + 8xyz - 99xy^2z^4 = 0$$

which statements are true?

"All birds have wings"

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

Defining the set of true statements

Making a decision procedure

Generating a list

A statement is a string of characters from an alphabet
some finite set Σ

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

all the elements: $S = \{ \text{Pikachu, Charmander, Squirtle, Bulbasaur} \} = \{ C, P, B, S \}$

A string of Σ is a sequence of Σ
 $P P P P$ $C S C S C S S \underbrace{S}_{} = \epsilon$

1-3 A language is a set of strings
 $\{\epsilon, P, PP, PPP, PPPP\}^*$ - finite $\{P, P^2, \dots, P^{256}, \dots\}$

$x \in S$ - x is inside S $P \in \{\epsilon, P, PP\}$

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

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

$x \in \overline{Y}$ iff $x \notin Y$ (but $x \in U$ - inverse)
 ↗ complement or negation of Y

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

$PP \circ BC = PPBC$

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

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

$PPCE$

is a group

lexicographic ordering of Σ

$lo(\Sigma_0, 13) = \underbrace{\epsilon, 0, 1, 00, 10,}_{600, 001, 010, 011, 100, 101, 110, 111}$...

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

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

1-3) $\text{lexi} : \text{num} \rightarrow \text{string of } \Sigma \quad |\Sigma| = 2$

$\text{lexi } 0 = \epsilon \quad \text{lexi } 1 = 0 \quad \text{lexi } 2 = 1$

$\text{lexi } 3 = 00$

$\text{lexi } n = \text{size of } \Sigma$

if $n < \text{size}^0$ then ret ϵ often

$(n - \text{size}^0) < \text{size}^1$ then convert $(n - \text{size}^0)$ into ϵ

$(n - \text{size}^0) - \text{size}^1 < \text{size}^2$ convert of len 2

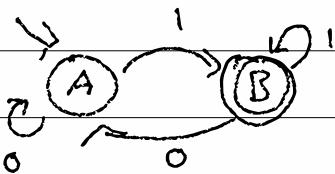
The set of strings in the lexicographic ordering

of $\Sigma \approx \Sigma^*$

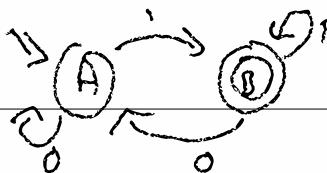
$\epsilon \in \Sigma^* \quad \text{PPCBSPPPP} \in \Sigma^*$

010111 $\in \Sigma^*$

Deterministic Finite Automata (DFA)



2-1) DFA



\circlearrowleft means loss

$$0110 = \text{No}$$

1 = Yes

\circlearrowright means No

$$0111 = \text{Yes}$$

11 = Yes

$$0010 = \text{No}$$

00 = No

transition function (edges)

$$1101 = \text{Yes}$$

$Q \times \Sigma \rightarrow Q$

$$\varepsilon = \text{No}$$

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

A	B
0	A
1	B

always finite
are the states

$\{\epsilon, A, B\}$

$\{\epsilon, 0, 1\}$

startstate
= A

accepting states
 $\{\epsilon, B\}$

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

No string DFA :

only empty string :

only the string 'J' :

'Ja' and 'Jb'

$: Q \times \Sigma \rightarrow Q$

2-3) DFA $d = (Q, \Sigma, q_0, \delta, F)$

accepts? $d \mid s : \text{DFA} \times \Sigma^* \rightarrow \text{Bool}$

accepts? $d \mid \varepsilon = \text{is } q_0 \text{ in } F:$

$d.F, \text{member}(d, q_0)$

accepts $d \mid c :: s$

$: \text{DFA} : Q : \Sigma^*$

accepts $d \mid s = \text{helper } d \mid d \cdot q_0 \mid s$

helper $d \mid q_i \mid \varepsilon = q_i \in d.F$

helper $d \mid q_i \mid c :: s = \text{helper } d \mid q_i \mid s$

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

DFA::Accepts (304 string s) {

$Q \mid q_i = \text{this} \cdot q_0;$

while (s != empty) {

$q_i = \text{this}, \delta(\text{delta}(q_i, s.\text{first}))$;

$s = s.\text{rest}$ }

return $\text{this}, F, \text{member}(q_i)$ }

↙ trace

2-3) 0110 \Rightarrow Even, Odd, Odd, Even

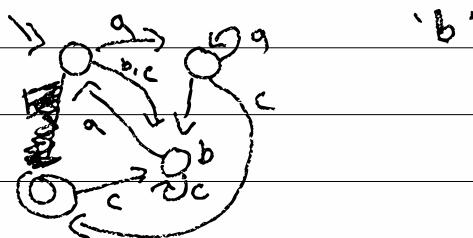
Transducers are DFAs where state writers
Moore machines

$L(d)$ = the language of DFA d
 $= \{ s \mid \text{accepts } d \text{ } s = \text{true} \}$
may be infinite

Given a DFA, return a string that would be accepted

example : DFA $\Rightarrow \Sigma^*$ or false

s.t. If example d returns s then
accepts? $d \text{ } s = \text{true}$



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

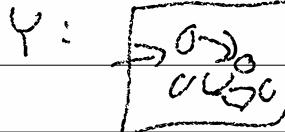
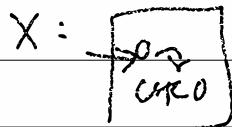
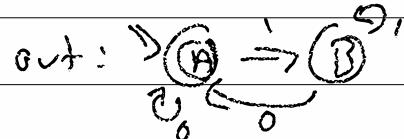
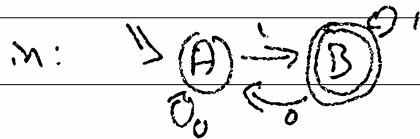
$$L(d') = \overline{L(d)} \quad (\text{ie } d' \text{ says } s \text{ yes})$$

negate: DFA \rightarrow DFA

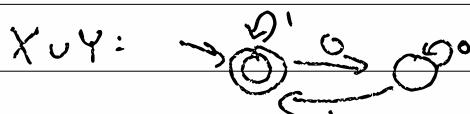
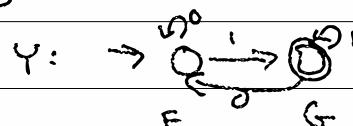
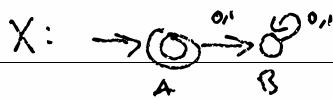
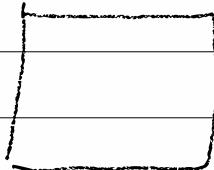
when d says no

negate (tddcs) = Evens

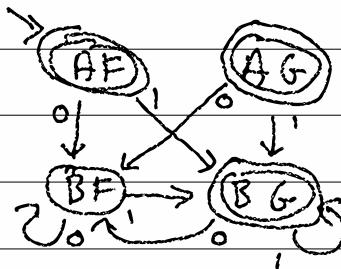
& vice versa)



$X \cup Y$:



Z:



2-5 union (x : DFA) (y : DFA) = (z : DFA)

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

$$z \cdot \Sigma = x \cdot \Sigma = y \cdot \Sigma$$

$$z \cdot g_0 = (x \cdot g_0, y \cdot g_0)$$

$$z \cdot F = \{ (g_x, g_y) \mid g_x \in x \cdot F$$

(or) $g_y \in y \cdot F \}$

$$z \cdot \delta((g_x, g_y), c)$$

$$= (x \cdot \delta(g_x, c),$$

$$y \cdot \delta(g_y, c))$$

and to make
intersect

$$X \subseteq Y \text{ (subset) iff }$$

$$\forall g \in X, g \in Y.$$

$$X = Y, \text{ iff }$$

$$X \subseteq Y$$

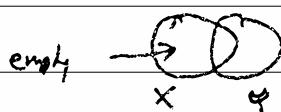
$$\text{and } Y \subseteq X$$

subset? : DFA \times DFA \rightarrow bool

subset? (\Downarrow_{DFA}) $X = \text{Yes}$

(epsilon) (Even) = Yes

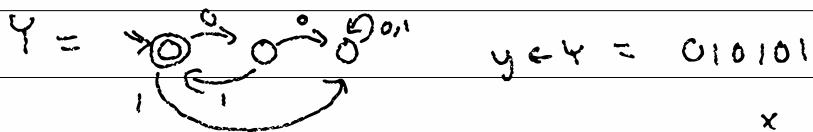
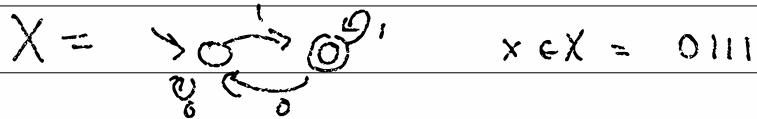
(epsilon) (odd) = No



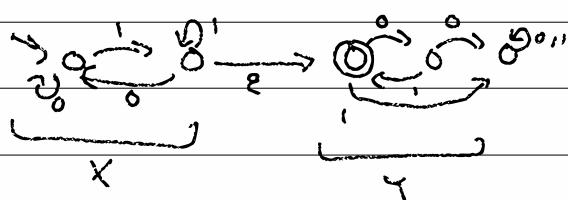
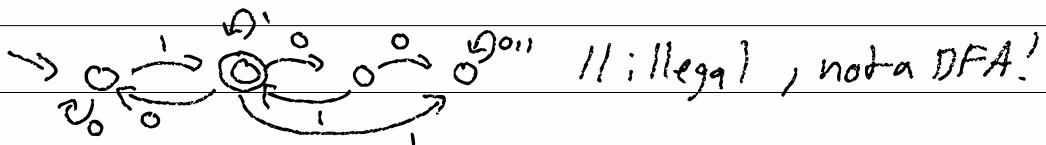
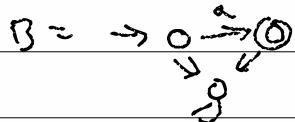
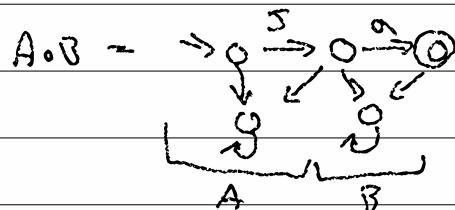
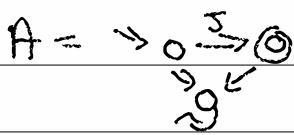
$X - Y$ must be empty

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

3-1 $z \in X^0 Y$ iff $z = xy$ where
 $x \in X$ and $y \in Y$



$$z \in X^0 Y = \overbrace{0111}^x \overbrace{010101}^y$$

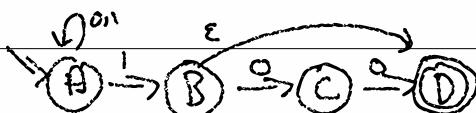


$F \xrightarrow{\quad} G$
 (skip from F to G
 at the right
 time)

3-2) NFA - non-deterministic finite automata

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

$$NFA = (Q, \Sigma, q_0, \delta: Q \times \Sigma \underset{\text{sigma events}}{\rightarrow} Q, F \subseteq Q)$$
$$\rightarrow P(Q)$$



S	A	B	C	// D
0	ΣA3	εC3	εD3	ε3
1	εA83	ε3	ε3	ε3
ε	ε3	εD3	ε3	ε3

$$(A,0)(A,1)(A,0)(A,0)$$

$x \notin L(n)$ iff

forall oracle $n \neq$ ~~for~~ (or "if")

trace = a sequence of $Q \times \Sigma \cup \epsilon$

$$(A,0) (B,1) (C,0) (D,0) \quad 0100 \in L(n)$$

$$(A,1) (A,0) (B,1) (D,\epsilon) \quad 101\epsilon = 101 \in L(n)$$

oracle interpretation : NFA \times trace \rightarrow boolean

oracle $N + =$ helper $N \cdot q_0 +$

helper $N q_i [] = q_i \in N \cdot F$

$((q_i, c) :: +') =$

is $q_i \in N \cdot \delta(q_i, c)$, then helper $N q_i +'$
o.w. "invalid trace"

3-3 / trace-tree = accept | reject
 | branch state (++, ...)

all : NFA $\times \Sigma^* \rightarrow \uparrow$
 ↑
 set

all N s = helper N N, g_0 s

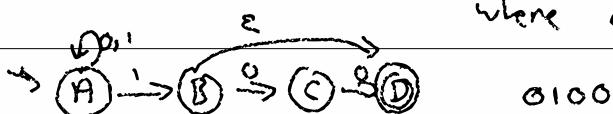
helper N g_i s =

branch g_i (case s where
 $\{ \} \rightarrow$ if $g_i \in N, F$ then
 $\{ \text{accept} \}$
 o.w.
 $\{ \text{reject} \}$)

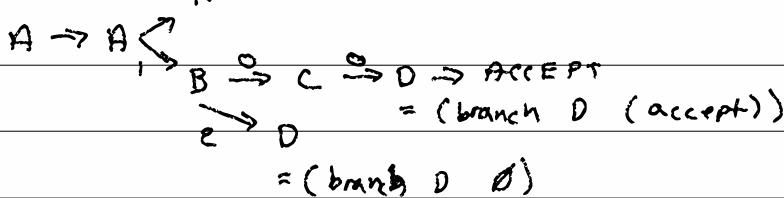
$c :: s' \rightarrow$

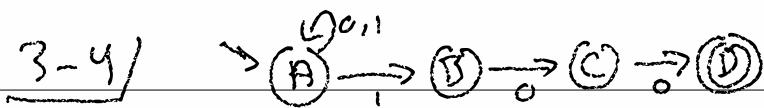
$\{ \text{++} \mid ++ = \text{helper } N g_j s'$
 where $g_j \in N, \delta(g_j, c) \}$

$\cup \{ \text{++} \mid ++ = \text{helper } N g_j s$
 where $g_j \in N, \delta(g_j, \epsilon) \text{ and } g_j \neq g_i \}$



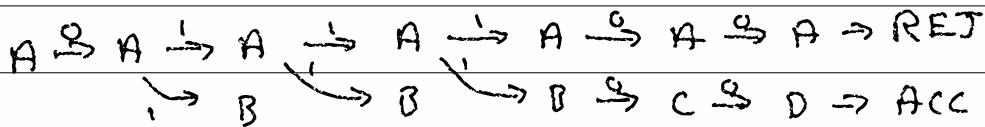
$0 \quad 1 \xrightarrow{A} \xrightarrow{B} \xrightarrow{C} \xrightarrow{D} \text{Reject}$





"ends in 100"

011100



backtrack : NFA \times String Σ^* \rightarrow Bool

backtrack N $s = \text{helper } N \text{ } N.g_0 \text{ } s$

helper N g; S =

OR case 5 with $\square \rightarrow g_i \in N, F$

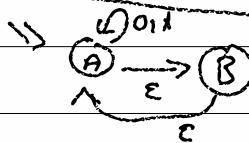
$c::s' \rightarrow OR_{g_j \in N.S(g_i, c)} helper(N.g_j.s')$

OR

$$g_j \in N_\epsilon(g_i, \epsilon)$$

helper N g; s

x=3 ; (1 || x++); x==3;



maybe DS = tree

unfold : $A \rightarrow DS(B)$

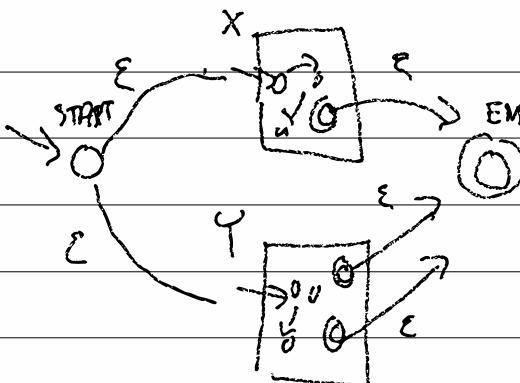
$$\text{fold} : \text{DS}(B) \rightarrow C$$

Haskell

there always exist

combined : A \Rightarrow C

4-1/



$$X = (Q_X, \Sigma, g_{0x}, \delta_X, F_X)$$

$$Y = (Q_Y, \Sigma, g_{0y}, \delta_Y, F_Y)$$

$$Z = (Q_Z, \Sigma, g_{0z}, \delta_Z, F_Z)$$

$$F_Z = \{\text{END}\}$$

$$g_{0z} = \{\text{START}\}$$

$$Q_Z = \{\text{START, END}\}$$

$$\delta_Z(g_i, c) =$$

$$\cup Q_X \times \{\emptyset\}$$

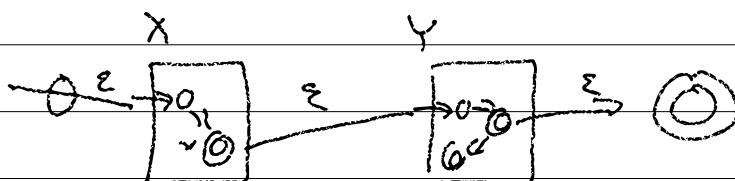
$$(\text{START}, \epsilon) = \{(g_{0x}, 0), (g_{0y}, 1)\} \quad \cup Q_Y \times \{\emptyset\}$$

$$(\text{START}, -) = \{\emptyset\}$$

$$(\text{END}, -) = \{\emptyset\}$$

$$((g_{0x}, 0), c) = \delta_X(g_{0x}, c) \times \{\emptyset\}$$

$$g_{0y}, \text{ similar} \quad \cup \text{ if } g_{0x} \in F_X \text{ and } c = \epsilon, \{\text{END}\} \text{ o.w. } \{\emptyset\}$$



4-2) Kleene-star X^*

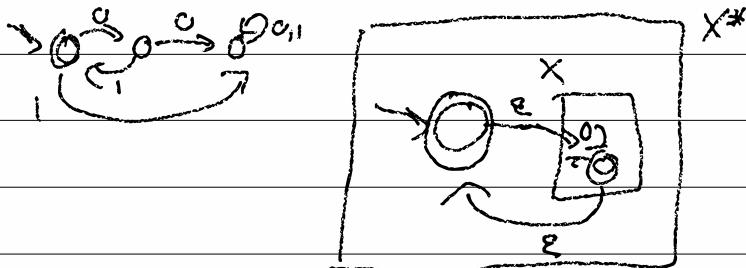
$\bar{z} \in X^*$ iff $\bar{z} = z$ OR $\bar{z} = xy$

where $x \in X$ and
 $y \in X^*$

iff $z = x_0 \dots x_n$

where $x_i \in X$

$(\{0\}^3 \{1\})^*$ = any number of 01 sequences

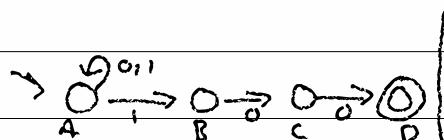


DFA's = $\cup, \cap, -$

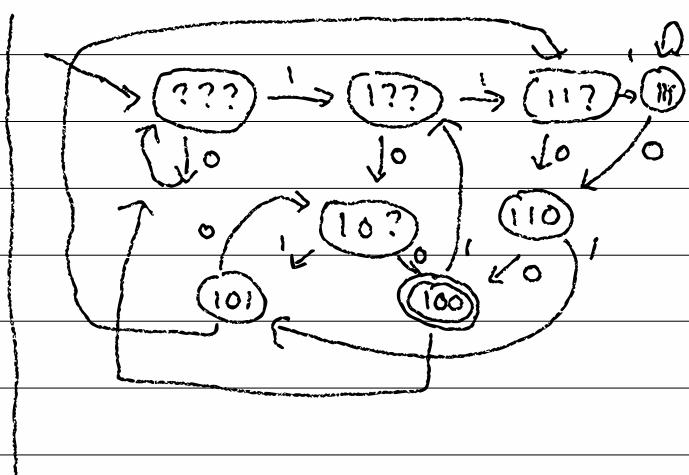
DEA \rightarrow NFA

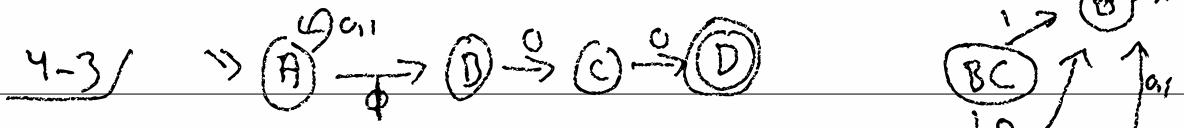
NFAs : \cup , \circ , *

wayt: NFA \Rightarrow DFA

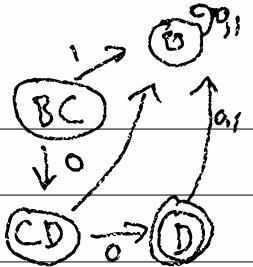


0100





01100
 $A \xrightarrow{0} A \xrightarrow{1} A \xrightarrow{0} A \xrightarrow{0} A \xrightarrow{0} A$
 $B \xrightarrow{0} B \quad C \quad D \leftarrow \checkmark$



???

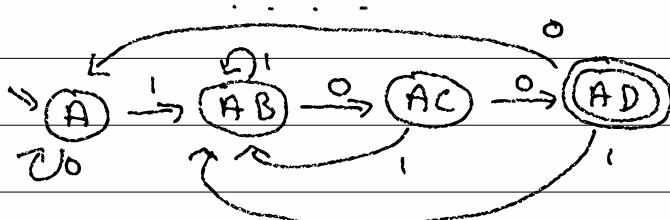
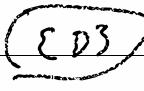
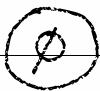
???

??

111

1102

100



01100 ✓

011001 X

1111 X

111100 ✓

NFA \Rightarrow DFA in : $(Q_N, \Sigma, \delta_N, F_N)$

out : $(Q_D, \Sigma, \delta_D, F_D)$

$$Q_D = P(Q_N) \quad \delta_D = \{ \delta_{q_n} \mid q_n \in Q_N \} \quad F_D = \{ q_n \mid q_n \in F_N \}$$

$$\delta_D(q_D, c) = \bigcup_{q_n \in q_D} \delta_N(q_n, c)$$

~~q_n ∈ q_D~~ ~~ε~~ ~~ε~~

$$E : Q_D \rightarrow Q_D = E(\ast) = \bigcup_{q_D} \bigcup_{q_n \in q_D} \bigcup_{c \in \Sigma} \delta_N(q_n, c)$$

Q_M

$\forall x \in \Sigma^*$, exec backtrace $N \ x$

 = accepts ($NFA \rightarrow DFA \ N$) $\ x$

random string

= pick a random number

(lex; n)

write the NFA manually as a VKA...

use the dfa equality checker to see
that

dfa-equal? manual ($N \Rightarrow D \ N$) = #true

regular expression $x \cup y$ regular $\rightarrow 0 \xrightarrow{\square} 1 \xrightarrow{\square} 0$
 \leftarrow Par

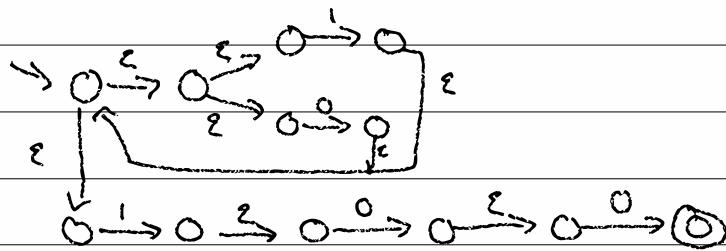
$x \cap y$ operations
 $x \circ y \rightarrow \square \rightarrow \square$

x^* $\rightarrow \square \xrightarrow{\square} \square$

$\emptyset \rightarrow \emptyset$

$\epsilon \rightarrow \emptyset$

$c \in \Sigma \rightarrow c \xrightarrow{\square}$



$$\#_c \Rightarrow \{^* \circ ' ' \circ ' c'$$

S-2/ interface RegEx;

class Empty implements RegEx; ()

Epstein RegEx ()

Char RE (char c)

Union RE (RE x, RE y)

Star RE (RE x)

Concat RE (RE x, RE y)

new Concat (new Star (new Union (new Char('1'),
new Char('0')))),

new Concat (new Char('1'),

new Concat (new Char('0'),

=

new Char('0'))))]

(1v0)*100

interface RegEx { NFA compile(); } }

Union::compile () { return nfaUnion (this.x.compile(),
this.y.compile()); }

S-3 / generate : RE $\rightarrow \Sigma^*$ s.t.

accepts? (nfa \Rightarrow dfa(compile r)) (generate r) = true

generate \emptyset = error

generate $x \circ y$ = gen x \circ gen y

gen ϵ = ϵ

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

gen c = c

gen ~~x~~ \cup y = case (flip)

heads \rightarrow gen x

tails \rightarrow gen y

printall : RE \Rightarrow void

printall r = helper r print

helper \emptyset pr = $\text{eg}(\text{void})$

h ϵ pr = pr ϵ

h c pr = pr c

h $x \circ y$ pr = h x newprint

(newprint s = h y npz

npz + = pr sat)

= h x (lambda s: h y (lambda t: pr sat))

h x^* pr = h ($\epsilon \cup x \circ x^*$) pr

h (xuy) pr = h x pr ; h y pr

numbers

$$\Sigma - y / \quad x \cdot 0 = 0 \quad x + y = y + x \quad x \cdot 1 = x$$

regex

$$x \circ \emptyset = \emptyset = \emptyset \circ x \quad \emptyset = \{\epsilon\}$$

$$x \circ \epsilon = x = \epsilon \circ x \quad \epsilon = \{\epsilon = "\prime"\}$$

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

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

$$\emptyset^* = \epsilon \quad x^* = \epsilon \cup x \circ x^*$$

$$\epsilon^* = \epsilon \quad \epsilon^* = \epsilon \cup \epsilon \circ \epsilon^* = \epsilon \cup \emptyset \circ \emptyset^*$$

$$(x^*)^* = x^* \quad \epsilon \cup \epsilon^* = \epsilon \cup \emptyset = \epsilon$$

$$x \cup z = z \text{ if } x \subseteq z$$

DFA_s \leftrightarrow NFA_s



$$\Rightarrow 0 \xrightarrow{0^{01}} 0 \xrightarrow{0} 0 \xrightarrow{0} 0 \Rightarrow (100)^* 100$$

IN

G-1/ NFA(k) \rightarrow GNFA ($2+k$)

RIP $\left[\begin{array}{l} \rightarrow \text{GNFA } (2+k-1) \\ \rightarrow \text{GNFA } (2+k-2) \\ \dots \end{array} \right]$ $\xrightarrow{\text{k times}}$

 \rightarrow GNFA (2)

OR

 \rightarrow REGNFA: $0 \xrightarrow{\delta} 0$

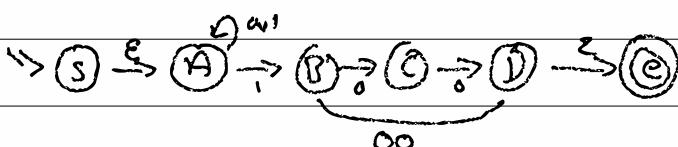
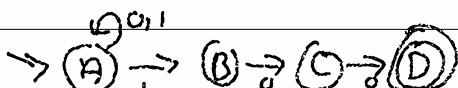
GNFA (generalized NFA)

 $0 \xrightarrow{\delta} 0$ $= (Q, \Sigma, g_a, \Delta, g_f)$ GNFA: $0 \xrightarrow{\delta} 0$ \uparrow one state, not a set $\Delta: (Q \times Q) \xrightarrow{\uparrow \downarrow} \text{Reg}$ $\delta: Q \times \Sigma \rightarrow P(Q)$ $(Q-g_f) \quad (Q-g_a)$ You can't
leave g_f

You can't

go back to g_a 

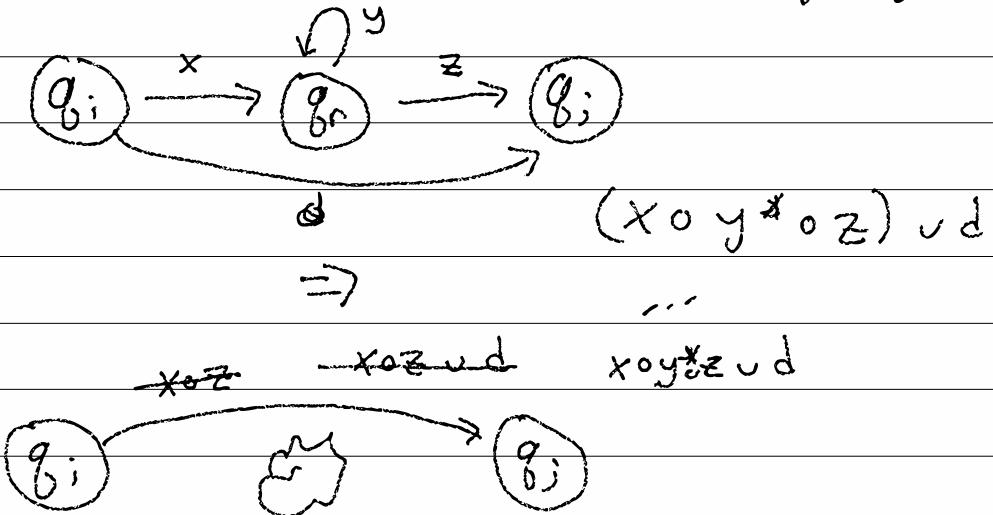
$\Delta(g_i, g_j) = r$ ~~\rightarrow~~ $x \in r$ then $g_i \xrightarrow{x} g_j$ in the NFA



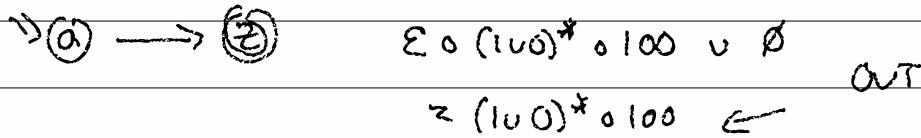
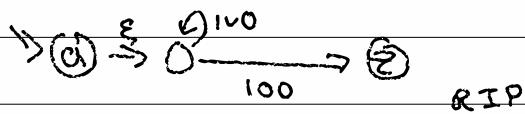
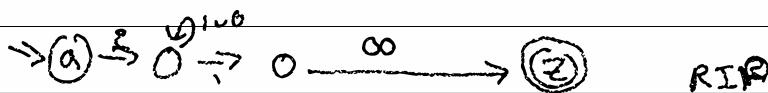
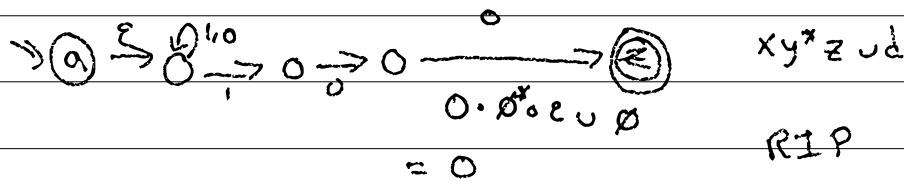
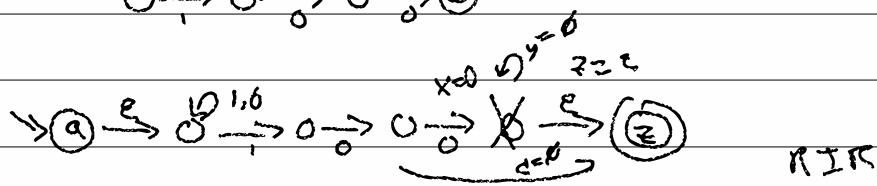
6-2/ $R_{10} : \text{GNFA } (n+1) \rightarrow \text{GNFA } (n)$

$\{q_0, q_f, q_r, q; \dots\} \rightarrow \{q_0, q_f, q; \dots\}$

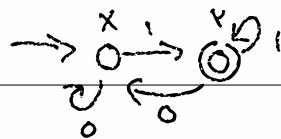
q_r is gone



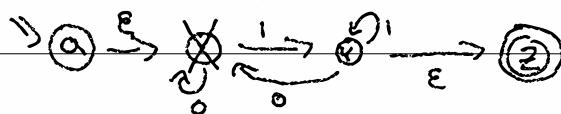
IN



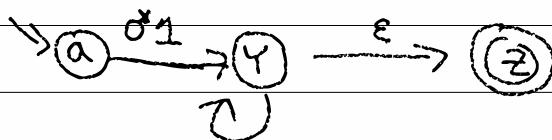
6-4)



$\Downarrow \text{IN}$

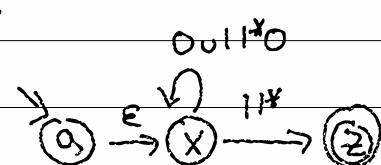


$\Downarrow \text{RIP}$



$1 \cup 00^*1$

$$0^*1 \circ (1 \cup 00^*1)^* = //$$



$\Downarrow \text{RIP}$

$(0011^*0)^*11^*$

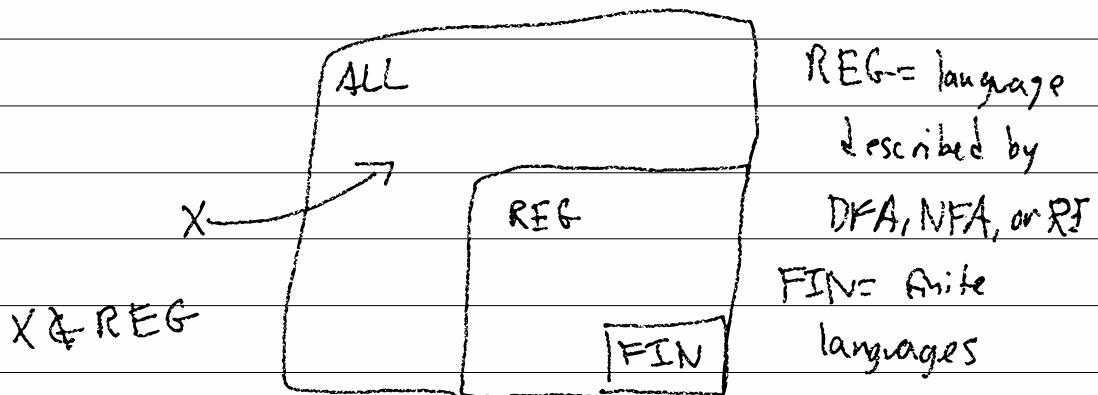
$(01)^*1$

DFA $\xleftarrow{\quad} \text{REX}$

Q.d. let $n = \text{dfa} \rightarrow \text{nex } d$

accepts? d (generate n) = true

G-5)



$$ALL = P(\Sigma^*) \quad \Sigma = \{0, 1\}$$

$$ALL = \Sigma^*, \quad \Sigma^* = \epsilon, 0, 1, 00, 01, 10, 11, \dots$$

$\{0, 1\}^*$, $\{0, 00, \epsilon, 000, 0000, \dots\}$

{all of Jay's lectures}

{JPEGs of rats}, {JPEGs of road signs}

... }
ALL?

Z-1) $\exists x \in \text{ALL} . \quad x \notin \text{REG}$

π \uparrow \uparrow
language all possible languages defined
(some problem)
languages by DFAs

option 1: $\forall y \in \text{REG} . \quad x \neq y$

option 2: $\forall y \in \text{REG} . \quad P(y)$
 $\neg P(x)$

mystery #1: What is x ? witness

#2: What is P ? property

~~the~~

\Rightarrow

proof 1: $\forall y \in \text{REG} . \quad P(y)$

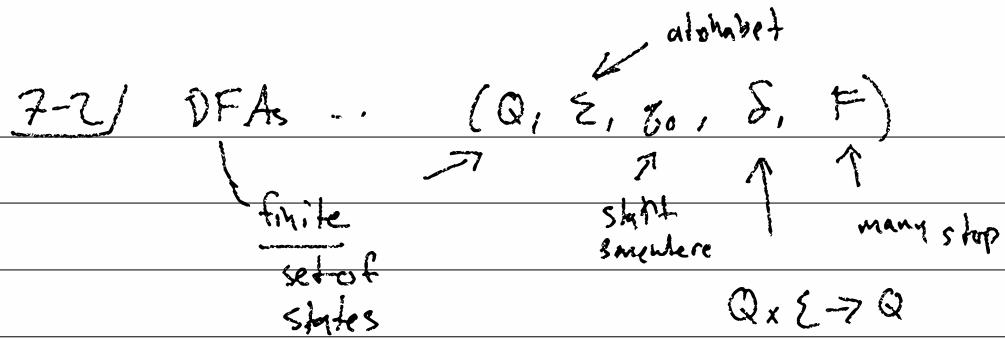
proof 2: $\neg P(x)$

\Rightarrow

$x \in \text{ALL}$, but $x \notin \text{REG}$

\Rightarrow

computers aren't omnipotent



EQ

$$\epsilon \in \text{EQ}$$

$$01 \in \text{EQ}$$

$$0011 \in \text{EQ}$$

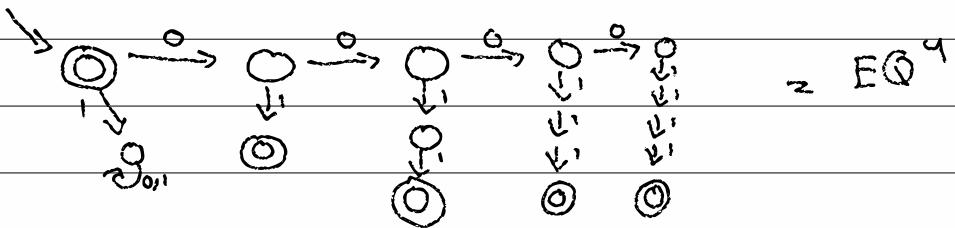
$$0 \notin \text{EQ}$$

$$010 \notin \text{EQ}$$

$$000111 \notin \text{EQ}$$

$$0110 \notin \text{EQ}$$

$$00001111 \notin \text{EQ}$$



$$|\text{EQ}^0| = 2$$

$$|\text{EQ}'| = 4 = |\text{EQ}| + 2$$

$$|\text{EQ}^3| = 11 = |\text{EQ}^2| + 4$$

$$|\text{EQ}^2| = 7 = |\text{EQ}| + 3$$

$$|\text{EQ}^n| =$$

$$|\text{EQ}^{n-1}| + n \cdot 2$$

$$= \frac{(n+1)(n+2)}{2} + 1$$

$$\forall n. 0^n 1^n \in \text{EQ}$$

$$\wedge 0^n 1^n \in \text{EQ}^m \text{ where } n \leq m$$

$$\forall m. \exists n. 0^n 1^n \notin \text{EQ}^m \quad (n = m+1)$$

7-3/ int count = 0; char c;
while (~~(c == '0')~~ c = getchar(); ~~(c == '0')~~)
 count++;
~~update(c);~~
while (c = getchar()) c == '1'
 count--);
return count == 0;

⇒

EQ^m, what is m? $m = 2^{31} - 1$

$$m = 2^{2^{31}}$$

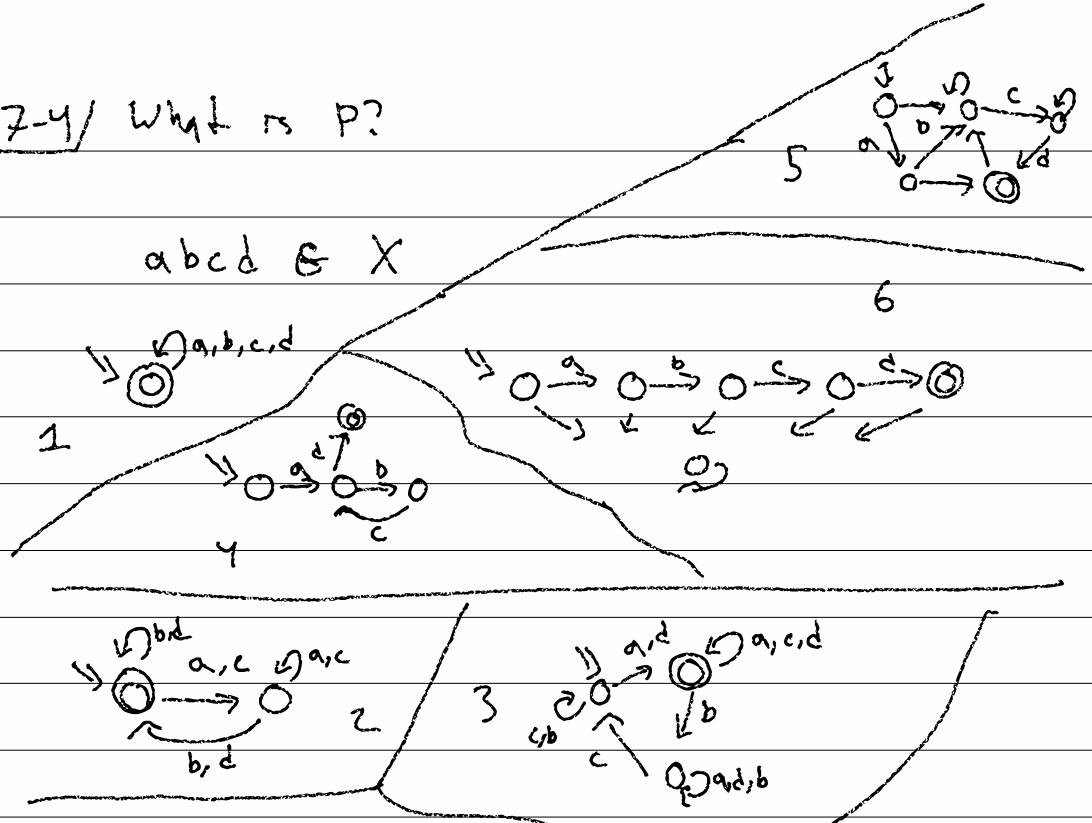
~~EQ^{2^31}~~

$0^n 1^n \in EQ$ for all n

~~x~~
=

Step 1: What is x? ✓

Z-Y / Why is P?



1: $a a a a b c d \in X$

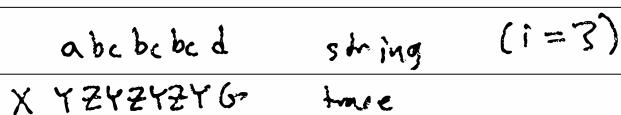
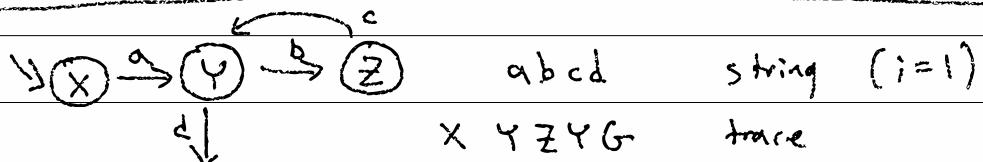
2: $a b a b a b a b c b \in X$

5: \times

3: $a b c a b c a b c d \in X$

4: $a b c b c b c d \in X$

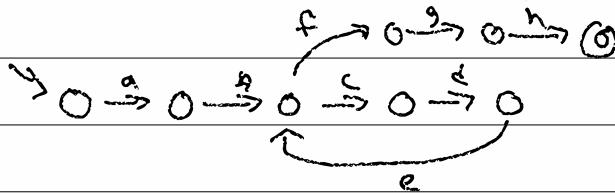
6: \times



$a(bc)^i d \in \text{DFA}$
for all $i \in [0, \infty)$

ad string ($i=0$)
 $X \quad Y \quad G$

7-5) DFAs must contain cycles!



If S is DFA, and s is long enough ($|s| \geq |Q|$),
then s must visit some state twice!

ex: $s = abcd$ s visited 4 times $|s|=4$

We express s as $x \circ y \circ z$

where x goes from q_0 to q_r

(y is not empty $\neq \epsilon$) y goes from q_r to q_r

$|xy| \leq |Q|$ z goes from q_r to $q_f \in F$

ex: $x = a$ $y = bc$ $z = d$ $q_0 = X$ $q_r = Y$ $q_f = G$

That means for all:

$x \circ y^i \circ z \in \text{DFA}$

ex: $i=0$ is $ad \in V$ $i=3$ is $abcbcbcd \in V$

7-6/ Regular Pumping Property (RPP)

RPP (A : Language) :=

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

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

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

$\wedge |y| > 0$)

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

$xy^i z \in A$.

Pumping Lemma: $\forall A \in \text{REG}$, RPP(A).

$p = |qi|$, x is the string before q_1

y is from q_1 to q_2

z is from q_2 to $q_f \in F$

Step 2: What is p ? ✓

Step 3: $\forall A \in \text{REG}$, P(A). ✓

Step 4: $\neg P(\text{EQ}) \dots$

8-1 $\neg \text{RPP}(\text{EQ})$

$\forall p \in N.$

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

$\forall (x, y, z \in S^*)$

$|y| > 0$

$|xy| < p$)

$\exists i \in N$

$xy^i z \notin A$

$\neg(A \wedge B) = \neg A \vee \neg B$

$\neg(A \vee B) = \neg A \wedge \neg B$

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

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

$\neg \text{RPP}(\text{EQ})$

given $p.$

choose $s \in \text{EQ}$ where $|s| > p$

$s = 0^p 1^p$

given x, y, z where $|y| > 0 \quad |xy| < p$

$s = xyz \quad 0^p 1^p = xyz \quad b > 0 \quad a+b+c \leq p$

$x = 0^a \quad y = 0^b \quad z = 0^c 1^p \quad a+b < p$

choose i $(i=0)$

$xy^i z \notin \text{EQ} \quad xy^i z = 0^a 0^{b+i} 0^c 1^p \notin \text{EQ}$

iff $a+bi+c \geq p$

$a+bi+c \geq a+b+c$

$b_i \geq b$

$i \neq 1$

8-2/ $s = xyz \in A$ and $xyz \in A$
then

$xy^*z \in A$
Regular expression

$xy^*z \subseteq A$

ALL \neq REG because $EQ \in ALL$

$EQ \notin REG$

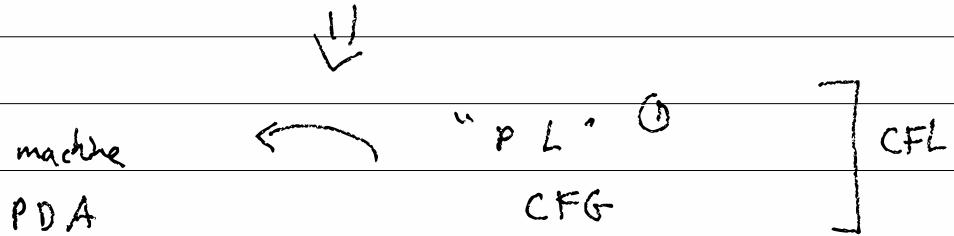
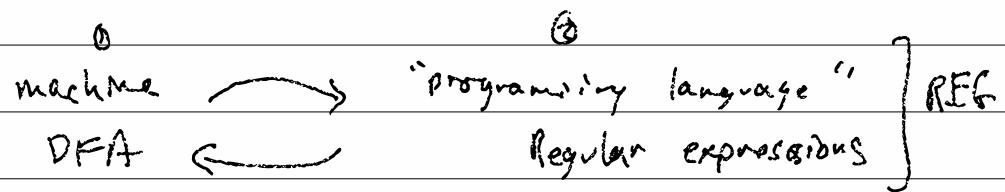
$$MEQ^0 = O^0 \circ EQ$$

$$MEQ^1 = O^1 \circ EQ$$

$$MEO^n = O^n \circ EQ$$

$$000 \circ 0011 \notin MEQ^3$$

9-1/ EQ $\ni 0^n 1^n$ for some n



CFG : context-free grammar substitution

$S \rightarrow OS1$ LHS is always a production derivation

a (FF for $0^n|n$) $V = \{S\}$ "variable" (non-hermeneutic), symbols

$S \rightarrow \epsilon \mid GS1$ RHS is a string of vars + terminals

$\Sigma = \{\epsilon, 1\}$ First var is "start symbol"

E → N | E O E

$$N \Rightarrow \emptyset \vdash \bot$$

$0 \rightarrow '+' / '-' / 'x' / ' \div '$

q2/ membership $x \in A$

defn $A = \{ \dots, \cdot, \sim \}$

generation it produces s_1, s_2, s_3, \dots

$S \rightarrow \varepsilon \quad | \quad OS1 \quad = g \quad \leftarrow$

$\overbrace{S \rightarrow OS1} \rightarrow OS11 \rightarrow 0011 \quad RL(g)$

$(S, [\circ, \cdot, \rightarrow], \{0, 1\}^G)$ = parse tree

$(S, [0,$
 $(S, []),$
 $1]),$

$$\text{PT} = \text{Var } x$$

$$(\Sigma + PT)^*$$

$\text{pt2str}(\text{V}, \text{seg}) = \text{seg2str seg}$

Seq2str [] = ε

$\text{seg2str } c \in \Sigma :: \text{seg} = c :: \text{seg2str seg}$

~~seg 2 str~~ ~~PT~~ = pt 2 str PT +
_o seg 2 str seg

CFG $g = (V, \Sigma, R, S)$

↓
↓
↓
 $v \in V$

9-3) ~~Augmenting grammar~~

$R : V \rightarrow \text{Set of } (V \cup \Sigma)^*$ $P(V \times (V \cup \Sigma)^*)$

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

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

$$(V \cup \Sigma)^* = \epsilon, 0, 1, \$, 00, 01, 0\$,$$

$$10, 11, 1\$, \$0, \$1, \$\$, \dots$$

$$V \times (V \cup \Sigma)^* = \{(S, \epsilon), (S, 0), (S, 1), \dots\}$$

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

$$\{ (S, 0), \{ (S, 0\$) \}$$

cfggen $g = \text{helper } g \circ g, S$

helper $g \circ v =$

let rules = $g, R \circ v$

rhs = random rules

return (V , map over s in rhs:

if $s \in V$, helper $g s$

o.w. s)

q-y/ all-n-deep g n = helper g n g.S

all-n-deep 0^n 1^n 2 = ε, 01, 0011

helper g n v = if n=0, don't return, o.w.

let rules = g, R v

for rhs ← rules; do

→ return (V, map s ← rhs; do

iterator if s ∈ V, helper g (n-1) s
o.w. s)

S → 0 S 1 → 00 S 1 1

REG = a language defined by some DFA

CFL - context-free languages = a lang defined by a CFF

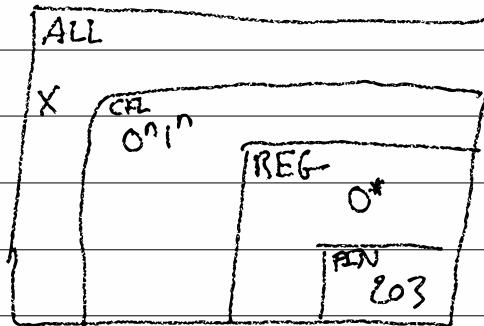
V S → S O V

S → N P | A N P | A A N P |

NP → N | PN

V → ... | Other V

Q-5)



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

$0^n * 0^n \in \text{CFL}$

$\text{REG} \subseteq \text{CFL}$

$X \subseteq Y$ iff $\forall a \in X, a \in Y$

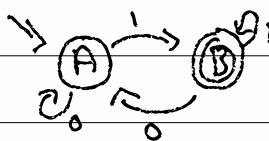
$a \in \text{REG}$ iff ~~there exists~~ $\exists d \in \text{DFA}, L(d) = a$

$a \in \text{CFL}$ iff $\exists g \in \text{CFG}, L(g) = a$

$\underbrace{\forall d \in \text{DFA}, \exists g \in \text{CFG}, L(g) = L(d)}$

$\underbrace{V \cup}_{\text{args}} \quad \underbrace{E \cup}_{\text{result}}$ is a fun

args result



①	②
$A \Rightarrow 1B$	$0A$
$B \Rightarrow \epsilon$	$1B \quad \quad 0A$

$$A \xrightarrow{1} B \xrightarrow{0} 11B \xrightarrow{1} 110A \xrightarrow{0} 1100A \xrightarrow{1} 11001B \xrightarrow{0} 11001$$

in: DFA = $(Q, \Sigma, q_0, \delta : Q \times \Sigma \rightarrow Q, F \subseteq Q)$

out: CFG = (V, Σ, R, S)

$$V = Q \quad \Sigma = \Sigma \quad S = q_0$$

If $\delta(q_i, c) = q_j$, then $R \ni q_i \rightarrow cq_j$

If $q_i \in F$, then $R \ni q_i \rightarrow \epsilon$

7-6/ dfa-accepts d

(pt2str (cfggen (dfa2cfg d))) = true

10-1/ regular operations: $\cup, \cap, \circ, *, -$

context-free ops: $\cup, \circ, *$

$$g_1 \cup g_2 = S \Rightarrow (S_1) \mid (S_2)$$

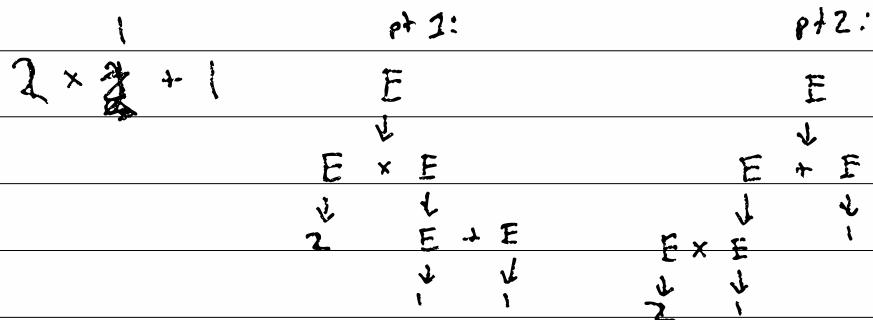
The diagram illustrates the union operation (\cup) for context-free grammars. It shows two separate boxes, each containing a set of production rules. The first box contains rules $S_1 \rightarrow \dots$, $V_1 \rightarrow \dots$, and $V_2 \rightarrow \dots$. The second box contains rules $S_2 \rightarrow \dots$, $U_1 \rightarrow \dots$, and $U_2 \rightarrow \dots$. An arrow points from the right side of the first box to the left side of the second box, indicating that the union of the two sets of rules results in a new grammar $g_1 \cup g_2$.

$$g_1 \circ g_2 = S \Rightarrow S_1 S_2$$

$$g_1 * = S \Rightarrow \epsilon \mid S, S$$

10-2 $s \in L(g)$ iff $\exists p \in g \text{ str}(p) = s$

$$E \hookrightarrow E \times E \quad | \quad E + E \quad | \quad 1 \quad | \quad 2$$



Ambiguous = $\exists s. \cancel{s \in \text{pt}_1} \vee \text{pt}_1, \text{pt}_2, \text{str}(\text{pt}_1) = s$

$\wedge \text{str}(\text{pt}_2) = s \not\rightarrow \text{str}_2$

$$\begin{array}{l} E \Rightarrow E + E \mid \cancel{E+E+E} F \\ F \Rightarrow F \times F \mid 1 \mid 2 \end{array} \quad] \text{ unambiguous}$$

ambiguous? : CFG \Rightarrow Bool

deambiguate : CFG \Rightarrow CFG s.t. amb? = #false

LL(k)

L&LR

LR

10-3/ $V \Rightarrow a b F c d G + 1 e V$ //complex

$$x+y = y+x$$

$$0+y = y = y+0 \quad \checkmark$$

$$(1+n)+y = 1 + (n+y) = (n+y)+1 = (y+n)+1 \quad \checkmark$$

$$\text{assume } n+y = y+n = y+(n+1)$$

CFG == NFA (Naam)

\Updownarrow CNF = Chomsky Normal Form

CNF == DFA

A grammar g is in CNF iff

If $r \in R$, then $r = A \rightarrow BC$ where

or $r = S \rightarrow \epsilon \quad B \in V, C \in V$

or $r = A \rightarrow a \quad \text{and } B \neq S, C \neq S, a \in \Sigma$

10-y / $S \Rightarrow ASA \mid aB$

$A \Rightarrow B \mid S$

$B \Rightarrow b \mid \epsilon$

$S' \Rightarrow S$

Add a new start sym

$S \Rightarrow ASA \mid aB$

$A \Rightarrow B \mid S$

$B \Rightarrow b \mid \epsilon$

Remove all ϵ s

$S' \Rightarrow S$

($V \rightarrow \epsilon$)

$S \Rightarrow ASA \mid aB \mid a$

$A \Rightarrow B \mid \epsilon \mid S$

$B \Rightarrow b$

$S' \Rightarrow S$

$S \Rightarrow ASA \mid SA \mid AS \mid S \mid aB \mid a$

$A \Rightarrow B \mid S$

$B \Rightarrow b$

Remove unit rules

$S' \Rightarrow ASA \mid SA \mid AS \mid aB \mid a$

($V \rightarrow U$)

$S \Rightarrow ASA \mid SA \mid AS \mid aB \mid a$

$A \Rightarrow b \mid ASA \mid SA \mid AS \mid aB \mid a$

$B \Rightarrow b$

Add intermediate vars

$S' \Rightarrow XA \mid SA \mid AS \mid 4B \mid a$

$X \Rightarrow AS$

$S \Rightarrow XA \mid SA \mid AS \mid 4B \mid a$

$Y \Rightarrow a$

$A \Rightarrow b \mid XA \mid SA \mid AS \mid 4B \mid a$

$B \Rightarrow b$

$$\frac{10-5/ \quad S \Rightarrow \epsilon \mid OS1}{S' \Rightarrow S} \text{ add } S'$$

$$\frac{S \Rightarrow \epsilon \mid OS1}{S' \Rightarrow S \mid \epsilon} \text{ removed } V \Rightarrow \epsilon$$

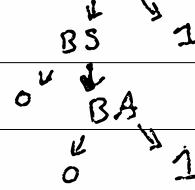
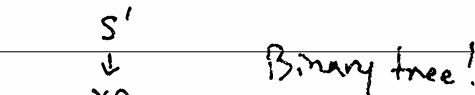
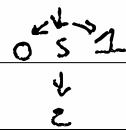
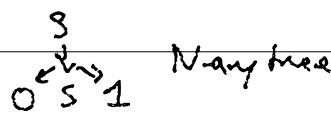
$$\frac{S \Rightarrow OS1 \mid 01}{S' \Rightarrow OS1 \mid 01 \mid \epsilon} \text{ remove } V \Rightarrow A$$

$$\frac{S \Rightarrow OS1 \mid 01}{S' \Rightarrow XA \mid BA \mid \epsilon} \text{ add mItem}$$

$$S \Rightarrow XA \mid BA$$

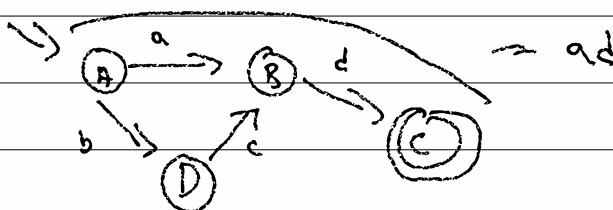
$$A \Rightarrow 1 \quad B \Rightarrow 0 \quad X \Rightarrow BS$$

$$S' \Rightarrow XA \Rightarrow BSA \Rightarrow OS1 \Rightarrow OXAA \Rightarrow OBSAA \Rightarrow OSAA \Rightarrow \\ OOBAAA \Rightarrow OOOAAA \Rightarrow OOO1AA \Rightarrow OOO11A \Rightarrow OOO111$$



CNF parse
trees are
binary!

11-1) Generating a string accepted by DFA



gsab {set of unvisited nodes} \times path to here \times here

$$\text{gsab } (\text{DFA } d) = \text{gsab } (d, Q - d, g_0) \in \Delta^Q$$

gsab Remaining Path $g_i =$

if $g_i \in d, F$ then return Path

if Remaining is empty then return FALSE

for ~~(g_{i+1})~~ in $\delta(g_i)$:

(c, g_{i+1})

if $g_{i+1} \in \text{Remaining} :$

$P = \text{gsab } (\text{Remaining} - g_{i+1}) \ (Path + c) \ g_{i+1}$

if $P \neq \text{false} :$ return P

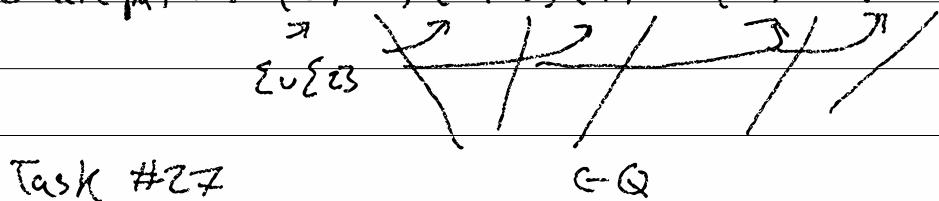
return FALSE

$$11-2 / 011_2 0 = 0110$$

$$\text{"011"} \circ \text{""} \circ \text{"0"} = \text{"0110"}$$

$$\text{NFA. } \delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$$

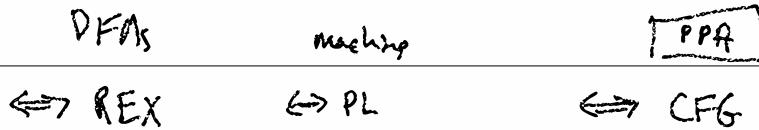
oracle path: $(0, A) (1, B) (1, C) (\epsilon, A) (0, B)$



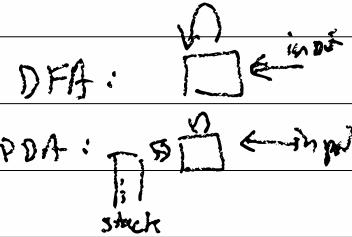
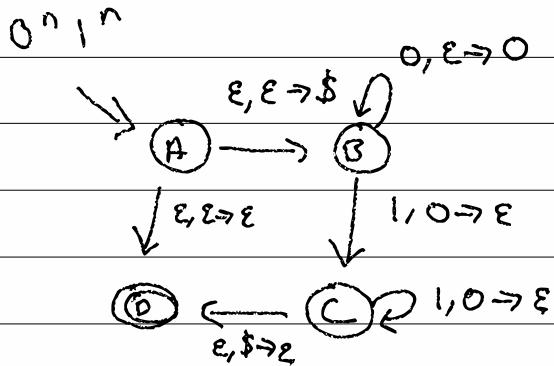
In Task #30, 0110

11-3 / Regular (REG)

Context-free (CFL)



PDA - push-down automata



$\otimes \xrightarrow{a,b \rightarrow c} \oplus$
Read a from input
Pop b from stack
Push c to stack

config = stack \times Q \times input \Rightarrow stack[a] input
 $\epsilon[A]0011 \Rightarrow \$[B]0011 \Rightarrow \$0[B]011 \Rightarrow \$00[D]11$
 $\Rightarrow \$0[c]1 \Rightarrow \$[c]\epsilon \Rightarrow \epsilon[D]\epsilon \Rightarrow \checkmark$

DKA = $(Q, \Sigma, q_0 \in Q, \delta: Q \times \Sigma \rightarrow Q, F \subseteq Q)$

NFA = $(Q, \Sigma, q_0 \in Q, \delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q), F \subseteq Q)$

PDA = $(Q, \Sigma_{\text{input}}, \Gamma_{\text{stack}}, q_0 \in Q, \delta: Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \rightarrow P(Q), F \subseteq Q)$

$$\delta: Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \rightarrow P(Q \times (\Gamma \cup \{\epsilon\}))$$

$\text{Tr}_{\mathcal{E} \in \mathcal{S}}$,

11-4/ pda-oracle P ($os : \text{List } (\Sigma \cup \{\epsilon\}, Q, \Gamma_0 \cup \{S\})$)
 pda-oracle $O^n \cap$ [$(\epsilon, \emptyset, B, \emptyset)$,
 $(0, \epsilon, B, 0)$,
 $(0, \epsilon, B, 0)$,
 $(1, 0, C, \epsilon)$,
 $(1, 0, C, \epsilon)$,
 $(\epsilon, \emptyset, D, \epsilon)$] = true

pda-oracle P $os = \text{helper } P \quad P.g_0 \quad os \in$

helper ($\text{PDA } P$) $(Q q_i)$ $(os \ os)$ ($st \ st$) =

if os is empty, net $q_i \in P, F$

let $[(C, a, q_j, b) :: os'] = os$

if $P, \delta(q_i, \epsilon, a) \ni (q_j, b)$

and $st = a \circ st'$ then

helper $P \quad q_j \quad os' \quad (b \circ st')$

O.V. FALSE

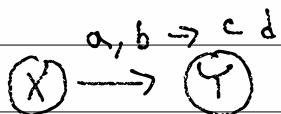
input NFA

c	$\{q_0, \dots, q_n\}$
ϵ	$\{q_1, \dots, q_n\}$

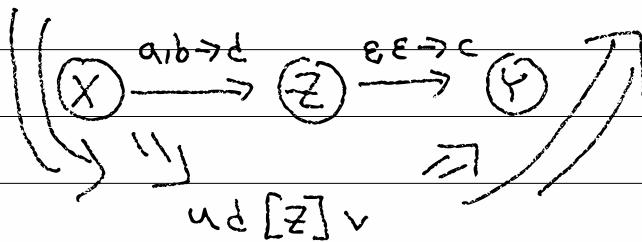
PDA stack

input	$\epsilon \dots \epsilon$	ϵ
c	$\epsilon \dots \epsilon$	$\epsilon \dots \epsilon$
ϵ	$\epsilon \dots \epsilon$	$\epsilon \dots \epsilon$

11-5/ Can a PDA push multiple things?

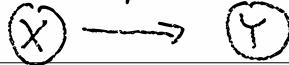


$ub[x]av \Rightarrow udc[y]v$



Can a PDA read more than 1 back?

$a, ?b \rightarrow ?c$



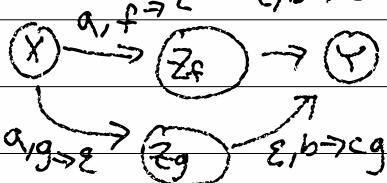
$f \in \Gamma$

$ubf[x]av$

$ubg[x]v$

$ub[z_f]v$

$ub[z_g]v$



$ucf[Y]v$

$ucg[Y]v$

$Z_i \text{ for all } i \in \Gamma$

11-6)

DFA A \rightarrow aB

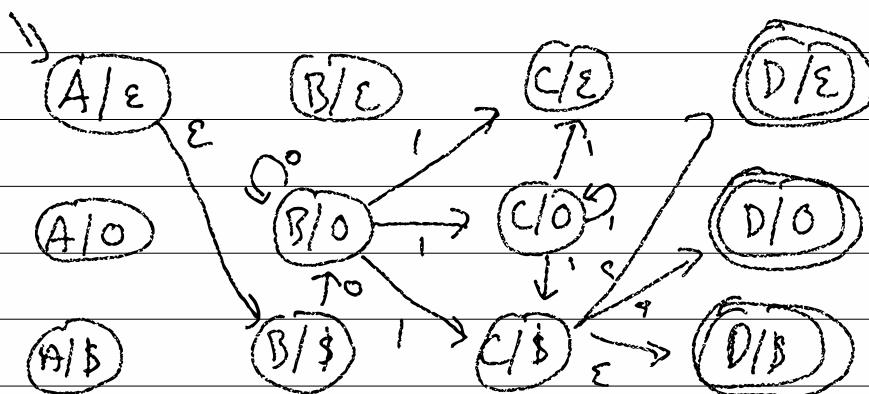
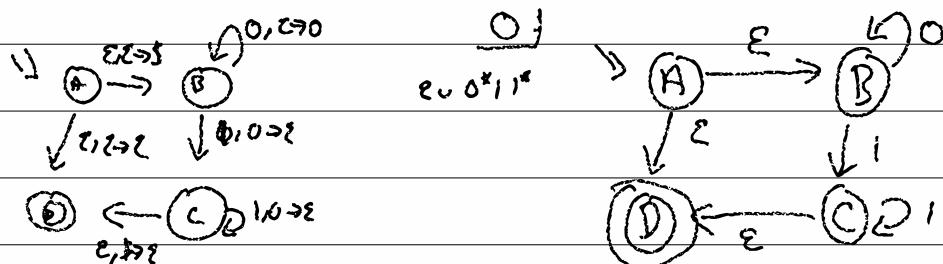
CFG A \rightarrow aBcCd

??? ABC \rightarrow aBcCd

TM AbC ϵ \rightarrow aBcD

$$Q \subseteq Q \times P^n$$

PDA to DFA (PDA P) (Nat n)



12-1) CFG \rightarrow PDA

input: CFG $g = (V, \Sigma, R, S)$

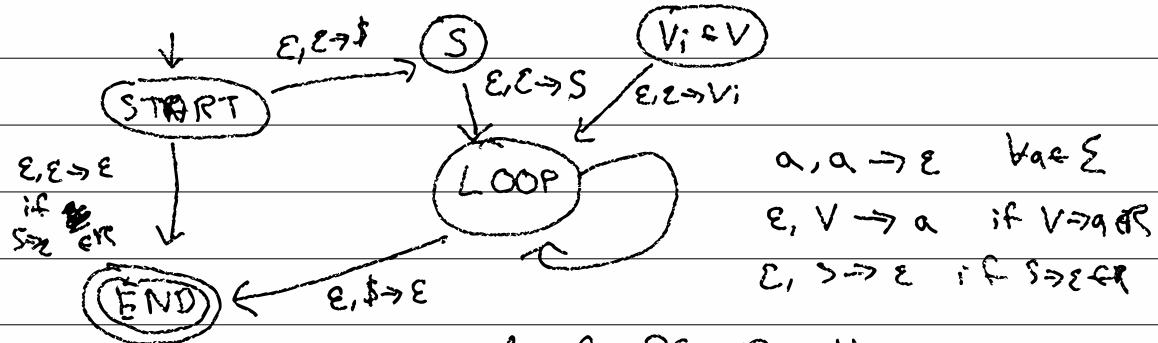
assume in CNF $R = A \Rightarrow a$
 $A \Rightarrow BC$
 $S \Rightarrow \epsilon$

output: PDA $p = (Q, \Sigma, \Gamma, q_0, \delta, F)$

$$\Gamma = V \cup \Sigma \cup \{\$\}$$

$$q_0 = \text{START} \quad F = \{\$ \in \Sigma\}$$

$$Q = \{\text{START}, \text{LOOP}, \$ \in \Sigma\} \cup V$$



if $A \Rightarrow BC \in R$, then

$$\delta(\text{LOOP}, \epsilon, A) \ni (B, C)$$

JZ-Z/ S → ε | OSI

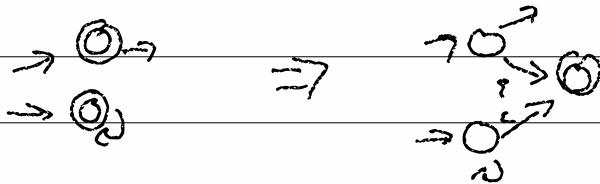
input: 000111

$\epsilon [START] 000111 \rightarrow \$ [S] 0^3 1^3 \rightarrow \$ S [LOOP] 0^3 1^3 \rightarrow$
 $\$ ISO [LOOP] 0^3 1^3 \rightarrow \$ P [LOOP] 0^2 1^3 \rightarrow \$ IIS O [L] 0^2 1^3 \rightarrow$
 $\$ IIS [L] 0^1 1^3 \rightarrow \$ III ISO [L] 0^1 1^3 \rightarrow \$ I^3 [L] 1^3 \rightarrow \$ I^3 [L] 1^3$
 $\$ I^2 [L] 1^3 \rightarrow \$ I [L] 1 \rightarrow \$ [L] \rightarrow [END] \rightarrow \checkmark$

12-3) PDA \rightarrow CFG

input : $P = (\mathbb{Q}, \Sigma, \Gamma, q_0, \delta, F)$

assume 1 : $F = \Sigma^* \cap \Gamma^*$



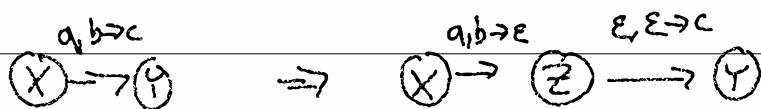
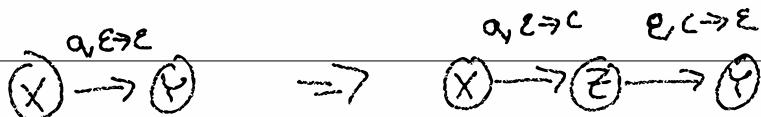
assume 2 : every transition pushes XOR pops

push : $a, \epsilon \rightarrow c$ (pushed c)

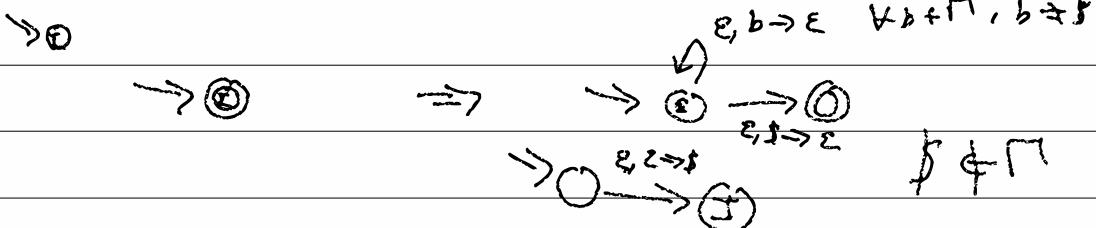
pop : $a, b \rightarrow \epsilon$ (popped b)

X ignore : $a, \epsilon \rightarrow \epsilon$ (ignore)

X replace : $a, b \rightarrow c$



assume 3 : the stack is empty on accept



12-y/ CFG $g = (V, \Sigma, R, S)$

$$V = Q \times Q \quad \Sigma = \Sigma$$

$$S = (g_0, g_f)$$

If (q_i, g_i) generates string s

$$\text{then } \varepsilon[q_i] s t \xrightarrow{*} \varepsilon[q_i] t$$

\downarrow \downarrow

If (g_0, g_f) generates string s and $u=t=\varepsilon$

$$\text{then } \varepsilon[g_0] s \xrightarrow{*} \varepsilon[g_f] \varepsilon \dots s \text{ is accepted by } P$$

$$\forall p \in Q \quad (p, p) \xrightarrow{*} \varepsilon \quad \text{path one refl}$$

$$\forall p, q, r \in Q. \quad (p, q) \xrightarrow{*} (p, r) \quad (r, q) \quad \text{paths are trans}$$

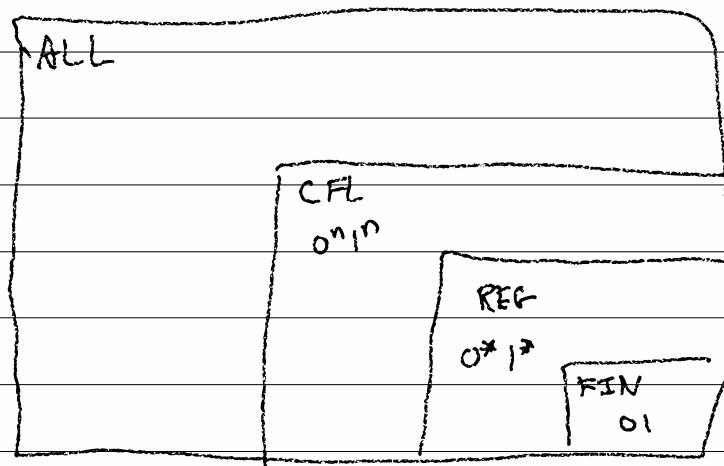
$$(r, +) \in \delta(p, a, \varepsilon)$$

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

$$(p, g) \xrightarrow{*} a \quad (r, s) \quad b$$

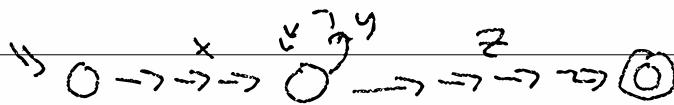
$$t \in T \quad p, q, r, s \in Q \quad a, b \in \Sigma \cup \{\varepsilon\}$$

15-1/

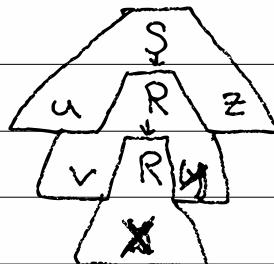


$\text{ALL} \neq \text{CFL} \leftarrow \exists x \in \text{ALL}, x \notin \text{CFL}$
 $\leftarrow \exists P, (\forall x \in \text{CFL}, P(x))$
 $\wedge (\exists y \in \text{ALL}, \neg P(y))$

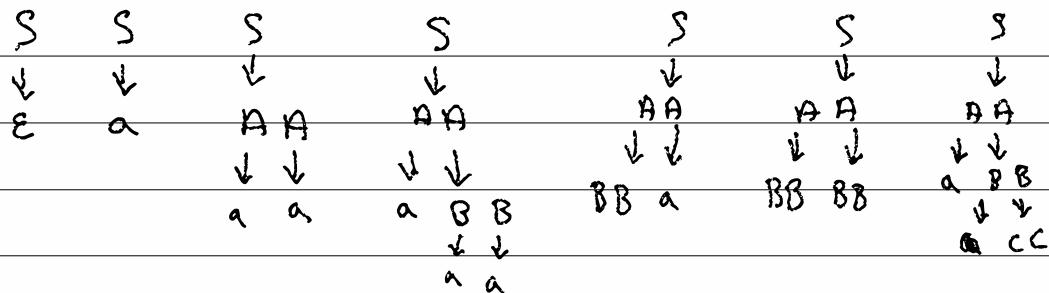
Regular PP. $s = xyz$ and $xyiz \in A$



Context PP $s = uvxyz$ and $uvi^ixyz \in A$



15-2/ Suppose F is CFG in CNF



$$S \Rightarrow AA \Rightarrow$$

How many chars

$$S \Rightarrow A^2 \Rightarrow B^4 \Rightarrow C^8 \Rightarrow D^{16}$$

are in a tree

$$V_0 \Rightarrow V_1^2 \Rightarrow V_2^4 \Rightarrow V_3^8 \Rightarrow V_4^{16} \Rightarrow a^{16} \text{ of depth } k?$$

$$[k+1, 2^k - 1]$$

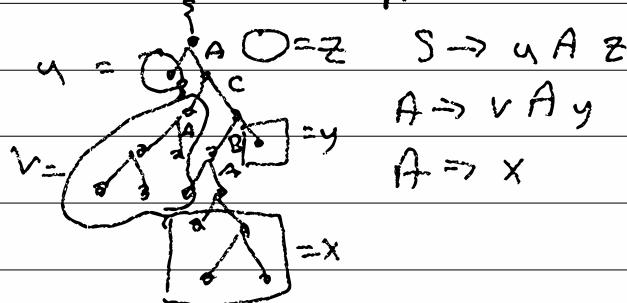
How deep is the tree of a string(accepted) with Nicks?

$$[\lg n + 1, n + 1]$$

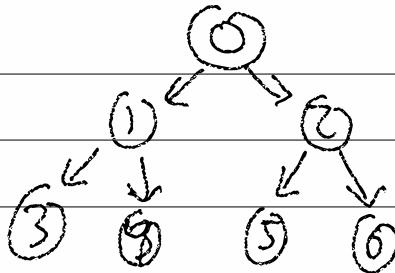
If a string is accepted/generated and has more than $2^{|V|} - 1$ chars, then ...

the tree is $|V|$ levels deep

and some variable appears on 2 levels



15-3



pre: 3140526

in: 0134256

post: 6250413

pre (L, V, R) =

visit(L) show(V) visit(R)

Context-Free Pumping Property (CFPP)

CFPP (A) =

$\exists p \in \mathbb{N} \quad - p = 2^{|V|+1}$

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

$\exists (u, v, x, y, z \in \Sigma^*) \quad |vx| \leq p$
 $\text{if } |vy| > 0$

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

$uv^ixy^iz \in A$

$uv^ixy^iz = 0^i011$

$S \Rightarrow \epsilon / 0S1$

$s = 0011$

$u = \epsilon \quad S \quad \epsilon = z$

$u = \epsilon \quad z = \epsilon$

$v = \epsilon \quad S(\epsilon) = y$

$v = \epsilon \quad y = 01$

$\epsilon \quad S(01) = x$

$x = 01$

$$\underline{15-y} \quad E \Rightarrow 0 \mid 1 \mid E+E \mid E \times E$$

$$1 + 1 \times 1 \quad u = 0 \quad E \quad 0 = z$$

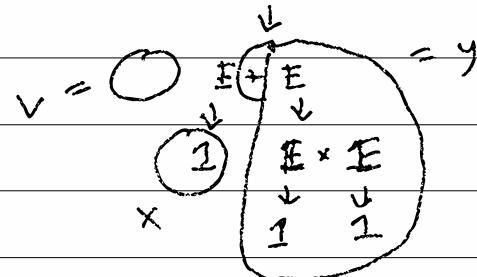
$$u = e$$

$$v = e$$

$$x = 1$$

$$y = + 1 \times 1$$

$$z = e$$



~~$$uv^0 \times y^0 z = 1$$~~

$$uv^2 \times y^2 z = 1 + ((1 \times 1) + (1 \times 1))$$

16-1) $\neg CFPP(A) :=$

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

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

$\forall (x_1, v, x, y, z \in \Sigma^*) \quad | \quad$

$|v_{xy}| \leq p$
 $\text{AP } |v_y| > 0$)

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

$uv^ix_yz \notin A$

$B = 0^n 1^n 0^n$

$\Sigma \in B$

$010 \in B$

$001100 \in B$

$00110 \in B$

$01100 \in B$

$00100 \in B$

\dots

init: see 0, push 0, goto init

see \emptyset , ~~pop 0~~^{replace pop 0, push 1}, ~~push 0~~^{push 0}, goto mid

mid: see 1, ~~push 0~~^{copy stack, no 0}, goto mid

[I] 001100

see 0, pop 0, goto end

00[I] 01100

end: see 0, pop 0, goto end

0000[I] 1100

done \Rightarrow yes

000[M]100

[I] 011 \rightarrow 00[I] 11 \rightarrow 0[n] 1

00[M]00

0[E]0

$\rightarrow [M] \rightarrow \checkmark$

[E] $\rightarrow \checkmark$

$C = 0^n 1^m 0^k \quad | \quad j+k=2n$

16-2/ TCFPP ($0^n 1^n 0^n$)

given: p

choose: $s \in B \wedge |s| \geq p$

$$s = 0^p 1^p 0^p$$

given: u, v, x, y, z st. $|vxy| \leq p \wedge |vy| > 0$

case 1: $\begin{matrix} 0^p & 1^p & 0^p \\ u & vxy & z \end{matrix}$ (only left Os)

case 2: $\begin{matrix} 0^p & 1^p & 0^p \\ u & v & xyz \end{matrix}$ (in between left Os & 1s)

case 3: $\begin{matrix} 0^p & 1^p & 0^p \\ u & vx & yz \end{matrix}$ (only 1s)

case 4: $\begin{matrix} 0^p & 1^p & 0^p \\ u & vxz & y \end{matrix}$ (in between 1s & right Os)

case 5: $\begin{matrix} 0^p 1^p & 0^p \\ u & vxz \end{matrix}$ (only right Os)

case I (3, 5): $vxy =$ only left Os

$$u = 0^a \quad vxy = 0^b \quad z = 0^c 1^p 0^p$$

$$\boxed{\begin{array}{l} a+b+c=p \\ b \leq p \\ b \leq p \end{array}} \quad \begin{array}{l} b = \hat{v} + \hat{x} + \hat{y} \\ v = 0^{\hat{v}} \quad x = 0^{\hat{x}} \quad y = 0^{\hat{y}} \end{array} \quad uv:xy:z \in B \Leftrightarrow \text{iff}$$

$$0^a 0^{\hat{v}} 0^{\hat{x}} 0^{\hat{y}} 0^p 0^c 1^p 0^p \in B \text{ iff}$$

$$a + \hat{v} + \hat{x} + \hat{y} + c = p \quad ((-1)\hat{v} + (-1)\hat{y}) = 0$$

$$(-1)(\hat{v} + \hat{y}) = 0 \quad -1 = 0 \quad i = 1$$

✓

(6-3) / case 2 (4) : vxy is LO, 1

$$u = 0^{\hat{a}} \quad vxy = 0^a 1^b \quad z = 1^{\hat{c}} 0^p$$

$$\hat{a} + a = p \quad \hat{c} + b = p$$

case 2,1 : $v = 0^{\hat{v}}$ $x = 0^c 1^d$ $y = 1^{\hat{y}}$

$$a = \hat{v} + c \quad b = \hat{y} + d$$

$$uvixyz = 0^{\hat{a}} 0^{\hat{v}} 0^c 1^d 1^{\hat{y}} 1^{\hat{z}} 0^p \in S$$

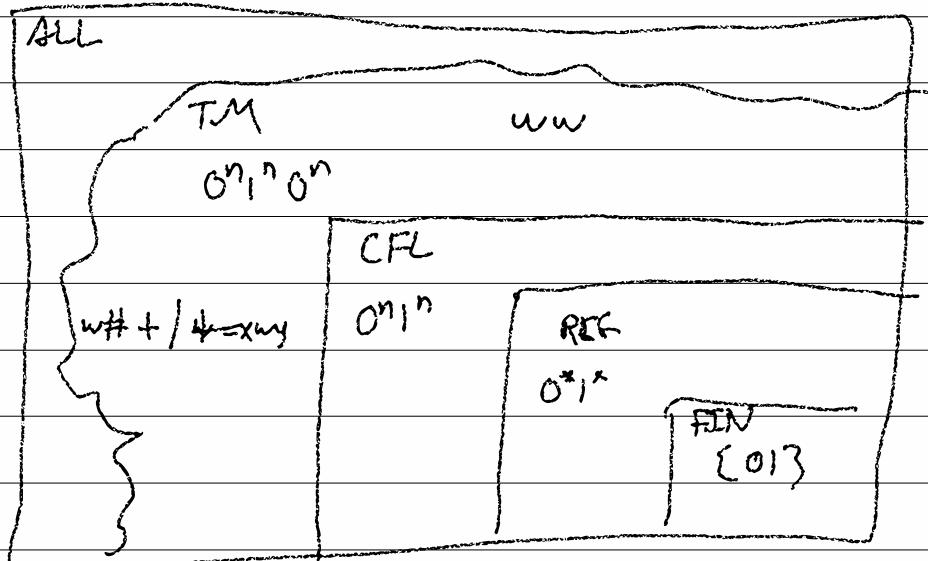
$$\text{iff } \hat{a} + \hat{v} + c = d + \hat{y} + \hat{z} = p$$

$$\hat{v} + c - a = 0 = \hat{y} + d - b \quad \hat{v} \geq 0$$

$$\hat{v}(i-1) = 0 = \hat{y}(i-1) \quad \hat{y} \geq 0$$

case 2,1,1 : $\hat{v} > 0 \quad \hat{v} + \hat{y} > 0$

$$i-1 = 0 \quad i = 1$$



(7-1) $O^n 1^n 0^n$ & CFL

$w \# w$ & CFL

$w \# w$ & CFL

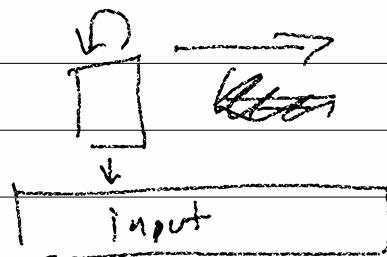
Turing Machines (TM)

-Turing Test -

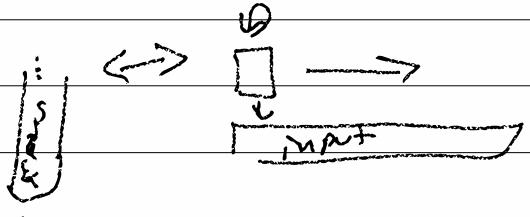
-ENIGMA

Alan Turing

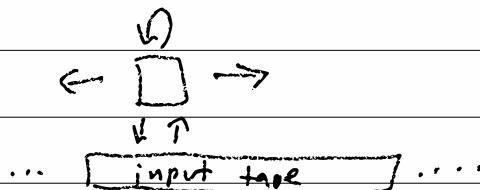
DFA



PDA



TM



17-2/ DFA: $\delta: Q \times \Sigma \rightarrow Q$

input-state input output
sym state

PDA : $\delta: Q \times \Sigma_e \times \Gamma_e \rightarrow P(Q \times \Gamma_e)$

optional $\xrightarrow{\text{input}}$ optional $\xrightarrow{\text{stack}}$ π nonlet $\xrightarrow{\text{stack}}$
input stack pop push

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

tape read $\xrightarrow{\text{input}}$ tape write $\xrightarrow{\text{stack}}$ π switch

$(Q, \Sigma, \Gamma, q_0, \delta, q_A, q_R)$

$q_0, q_A, q_R \in Q$ q_0 is the start

$\Sigma \subseteq \Gamma$ q_A is the ACCEPT

$w \in \Gamma$ q_R is the REJECT

The tape is infinitely long and starts as

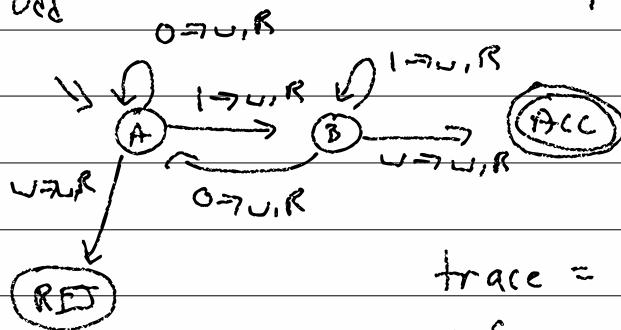
$\dots w$ input $w \dots$

$\xrightarrow{\text{start}}$ here

$$\Sigma = \{a, b\}$$

$$\Gamma = \{0, 1, \omega\}$$

17-3) Odd



trace = seq of config
 config = $\Gamma^* [Q] \Gamma^*$

run on 01101 ... $\cup [A] 01101 \cup \dots$

... $\cup [A] 1101 \cup \dots$

$[B] 101 \rightarrow [B] 01 \rightarrow [A] 1 \rightarrow$
 $[B] \sim \rightarrow [ACC] \rightarrow \checkmark$

If $\delta = (Q, \Sigma, q_0, \delta: Q \times \Sigma \rightarrow Q \times F)$

then $\tau = (Q \cup \{\text{ACC}, \text{REJ}\}, \Sigma, \Sigma \cup \{\omega\}, q_0, \delta', \text{ACC}, \text{REJ})$

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

$\delta(q_i, \omega) = \text{ACC if } q_i \in F$

REJ if $q_i \notin F$

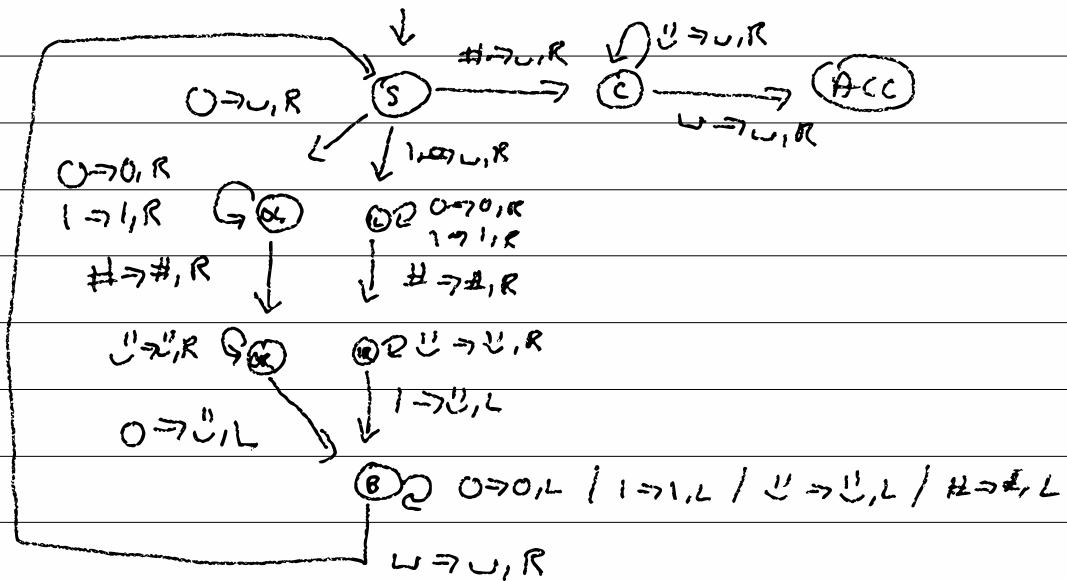
Tape =

$\perp \leftarrow \boxed{A} \leftarrow \boxed{B} \leftarrow \boxed{T} \leftarrow \boxed{C} \rightarrow \boxed{D} \rightarrow \perp$

$$\Sigma = \{0, 1, 3\}$$

$$\Pi = (\Pi_1, U, \cup)$$

17-4) $w \# w$



$$[S] 010\#010 \rightarrow [0L] 10\#010 \xrightarrow{?} 10[0L]\#010 \rightarrow 10\# [0R] 010 \xrightarrow{?} 10\# [B] \#U10 \rightarrow [B] \#U10 \#U10 \rightarrow [S] 10\# U10 \xrightarrow{?} [S] 0\#UU0$$

$$[S] \#UUU \rightarrow [C] UUU \rightarrow [C] \cup \rightarrow ACC \checkmark$$

$$2|w|^2 + 2|w| = O(|w|^2)$$

[7-5] simulate : TM \times input \rightarrow tape

simulate + s = h + ([], +.go, s)

$h + \overset{c}{\underset{=}{n}}(\text{before}, g_i, \text{after}) = \text{cons}$ cn

(cn) case after of

[] \rightarrow (\cup , [])

c: after' \rightarrow (c, after')

(g_i, c', d) = +.f(g_i, c)

case d of

L \rightarrow case before of

[] \rightarrow h + ([], g_j, \cup :c' +')

b: before' \rightarrow h + (before', g_j, b:c':q)

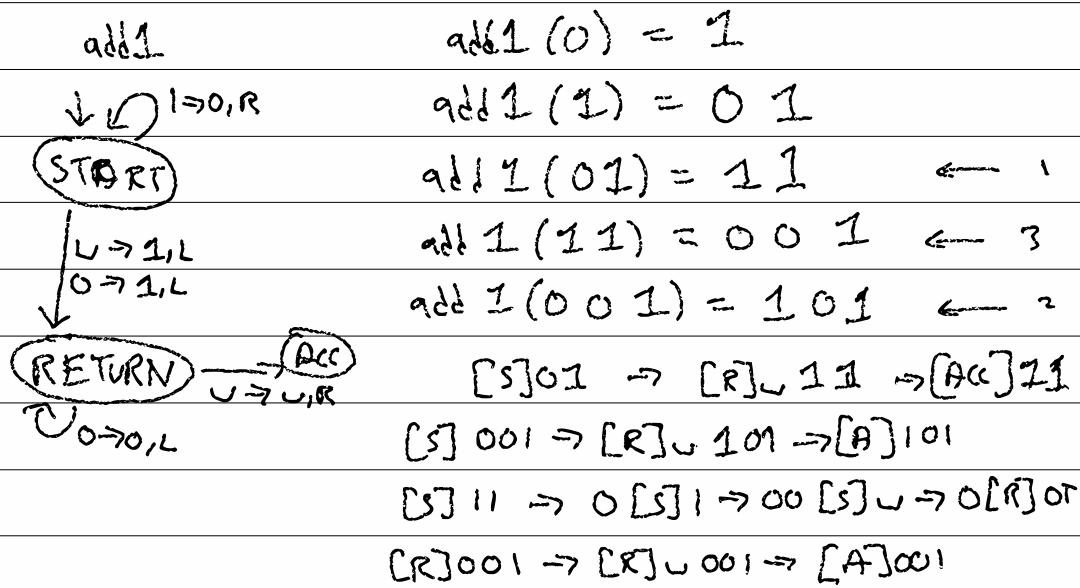
R $\xrightarrow{h+}$ (c': before, g_j, after')

h + (before, ~~g_j~~cc, after) = (1B + T)

REJ = (1B + F)

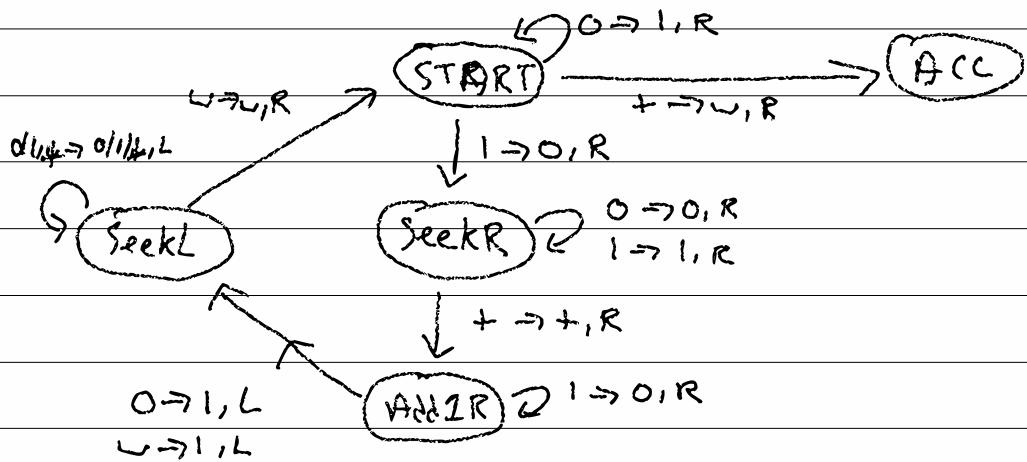
17-6) $x \in L(f)$ iff $[q_0]x \Rightarrow^* u[q_a]v$
 acceptance

$y = f(x)$ iff $[q_0]x \Rightarrow^* u[q_a]y$
 Computable function



$$1B-1) \quad X + Y \quad 0+Y \Rightarrow Y$$

$$(1+X)+Y \Rightarrow X + (1+Y)$$



$$2+1 \Rightarrow 01+1$$

$[S]01+1 \Rightarrow 1[S]1+1 \Rightarrow 10[SR]+1 \Rightarrow 10+[AR]1 \Rightarrow$
 $10+0[AR]\omega \Rightarrow 10+[SL]01 \Rightarrow [ST]10+01 \Rightarrow$
 $0[SR]0+01 \Rightarrow 00+[AR]01 \Rightarrow 00[SL]+11 \Rightarrow$
 $[ST]00+11 \Rightarrow 1[ST]0+11 \Rightarrow 11[ST]+11 \Rightarrow 11\omega[ACC]11$

18-3 DFAs defined Regular Languages
REC

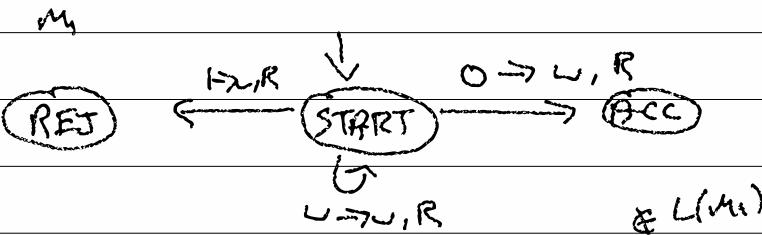
CFGs \equiv Context-Free Langs
CFL

TM $=:$ Turing-recognizable
 Σ_1

$A \in \Sigma_1$ iff $\exists M \text{ TM}, L(M) = A$

When a TM runs on input x ,

- 1) ACCEPT $[q_0]x \Rightarrow^* u[q_a]v$
- 2) REJECT $[q_0]x \Rightarrow^* u[q_r]v$
- 3) DIVERGE/
LOOP $\forall q_i, u, v. [q_0]x \Rightarrow^* u[q_i]v \rightarrow$
 $\exists q_j, f_j. u[q_i]v \Rightarrow f_j[q_j]g_j$.
 and $g_j \neq q_a$ or q_r



$0 \epsilon^* \in L(M_1)$ $1 \epsilon^* \in L(M_1)$ M_1 rejects
 $\epsilon^{*\in L(M_1)}$ M_1 diverges

$x \in L(M_1)$ iff $[q_0]x \Rightarrow^* u[q_a]v$

18-3

TMs



recognizers

\supseteq

deciders

||

||

may diverge

Never diverge

$\forall x \in \Sigma^*, M(x) = ACC$

$\forall x \in \Sigma^*, M(x) = ACC$

$\vee M(x) = REJECT$

or $M(x) = REJECT$

$\vee M(x)$ diverge

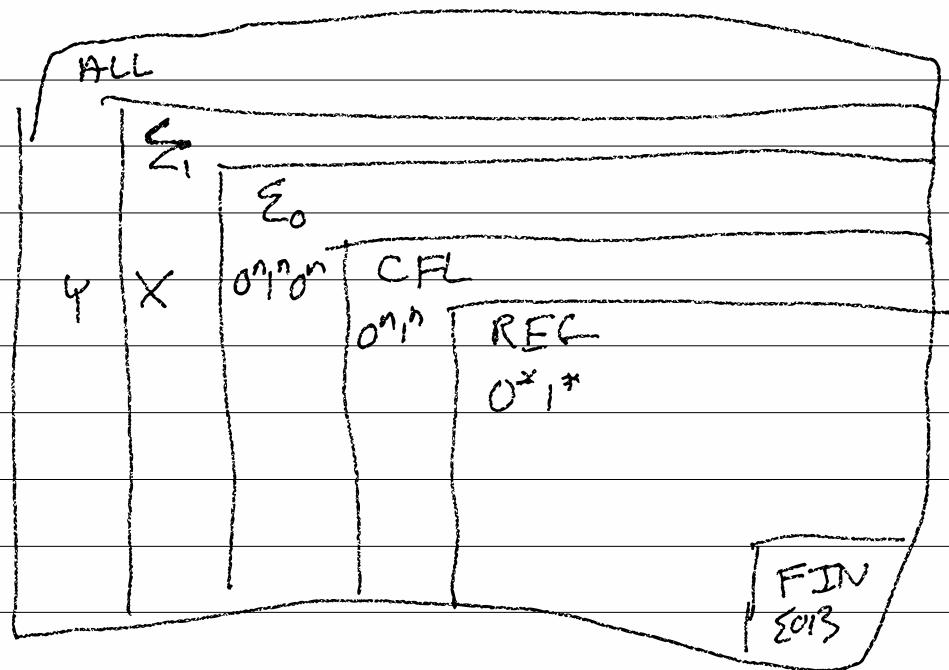
$\Sigma_1 =$ Turing-recognizable $A \in \Sigma_1$ iff

$\Sigma_0 =$ Turing-decidable $\exists f \in \text{recog. } L(f) = A$

$A \in \Sigma_0$ iff

$\exists f \in \text{deciders. } L(f) = A$

18-4 /



class DFA < State >

Function < State, Bool > Q

Function < Pair < State, char >, State > D

~~State = pair (char, function)~~

↳ DFA(x, y) Union $\{ \text{DFA } <x> \times, \text{DFA } <y> y \}$

... $Q = g \Rightarrow x.Q(q, \text{fst}) \wedge y.Q(q, \text{snd})$

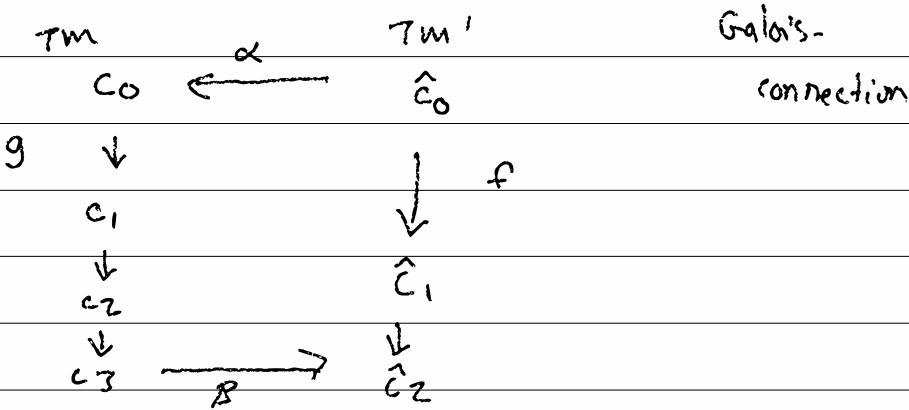
$g_0 = \text{newPair}(x, g_0, y, g_0)$

19-1 DFAs \longleftrightarrow NFAs \longleftrightarrow REs

compile : NFA \rightarrow DFA

decompile : DFA \rightarrow NFA

$\forall R \in \text{REs}, \exists N \in \text{NFA}, L(R) = L(N)$



19-3) Stay-Put TM

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

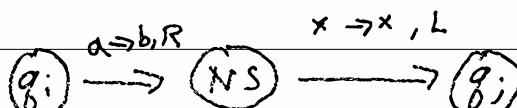
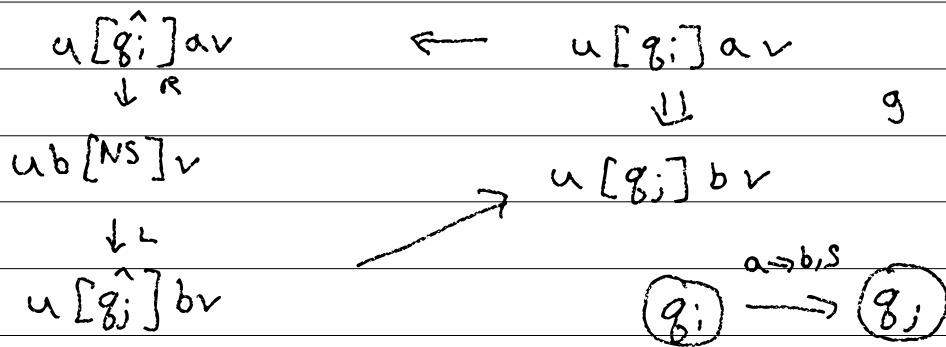
SP : $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$

$$\underline{\delta(q_i, a) = (q_i, b, L)} \quad \underline{\delta(q_i, a) = (q_j, b, R)}$$

$$u[q_i]av \Rightarrow u[q_j]c bv \quad u[q_i]av \Rightarrow ub[q_j]cv$$

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

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



$$\Gamma = \{0, 1, \omega\}$$

$$\begin{array}{l}
 0 \rightarrow 0, L \\
 1 \rightarrow 1, L \\
 \omega \rightarrow \omega, L
 \end{array}$$

19-3 / Multi-tape TM

$$S: Q \times \Gamma \times \Gamma \rightarrow Q \times (\Gamma \times \{\text{L}, \text{R}, \text{S}\}) \\ \times (\Gamma \times \{\text{L}, \text{R}, \text{S}\})$$

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

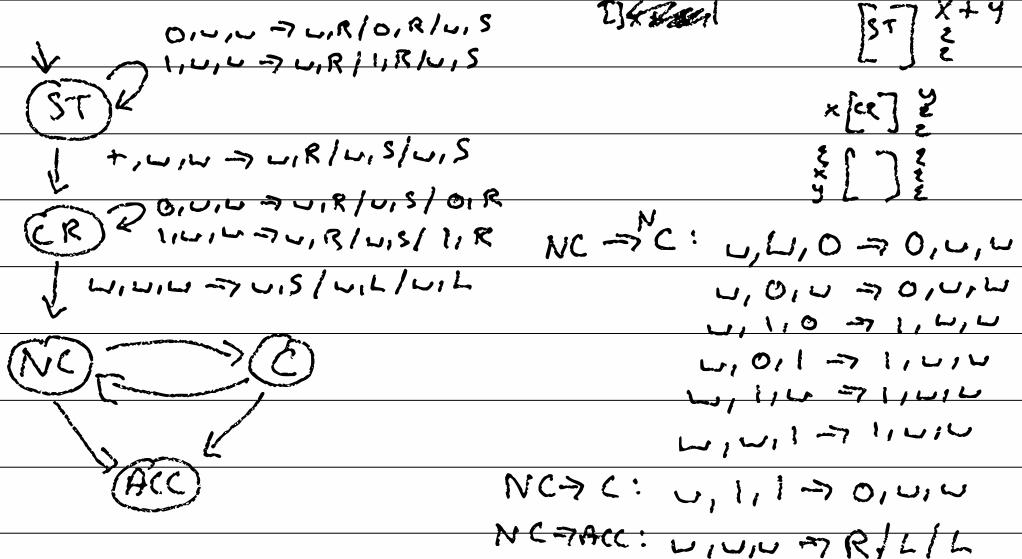
$$\underline{S(q_i, a, x) = S(q_i, b, L, y, R)}$$

$$f \begin{bmatrix} q_i \\ x \end{bmatrix} \xrightarrow{av} f \begin{bmatrix} q_i \\ y \end{bmatrix} \xrightarrow{zg} \begin{bmatrix} u \\ v \end{bmatrix} \xrightarrow{cbv}$$

binary addition MTM

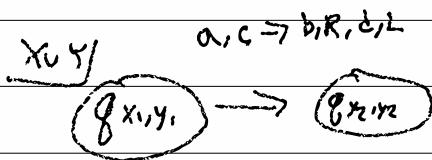
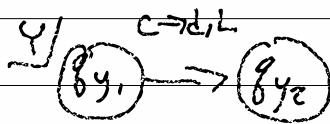
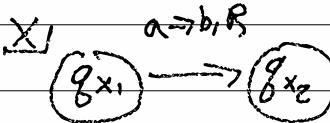
$$10 + 01 = 11$$

$$\text{input: } (01)^* + (01)^* \quad \text{output: add}$$



19-y/ Σ_1 and Σ_0 are closed under U and A

$\text{Union}_0(X, Y) :=$ copy input to tape 1
 move back to start of both
 simulate X and Y at same time
 if one reaches ACC, we ACC
 if both reach REJ, REJ



$$TM \xleftarrow{\alpha} MTM$$

$$\downarrow \quad \quad \quad \downarrow \\ \xrightarrow{\beta} MTM$$

$$\alpha : \underset{x}{\cup} [q_i]^{av} \xrightarrow{\beta} [q_i]^{av \# x^b y} : B$$

$$q_i \rightarrow q_i / L$$

$$\xrightarrow{\gamma} q_i / L^a / q_i$$

$$q_i \xleftarrow{\text{seeky}} q_j / q_i$$

$$u [q_i]^{av \# x^b y}$$

$$u[q_i/a]^v \# x^b y$$

$$u^{av \# x} [q_i/a]^b y$$

30-1] Σ_1 and Σ_0 are based under 0 , $*$

$xoy \in X_0 Y$ iff $x \in X$ and $y \in Y$

|s| # of places to divide - non-deterministically
choose which

Concat(X, Y) =

loop: choose between #1. copy char to tape 1;
goto loop
#2: stop

stop: simulate of X and Y (X sees tape 1)
check both finish (Y sees tape 0)

Non-det TM

20-2) $\delta: Q \times \Gamma \rightarrow (Q \times \Gamma \times \{L, R\}) + (Q \times Q)$

forking:

$$u[g_i]av \xrightarrow{\text{if}} u[g_L]av \quad u[g_R]av \xrightarrow{\text{if}}$$

config: seq (det-config)

$$\epsilon \Rightarrow \text{REJ}$$

$$\delta(g_i, a) = (g_i, b)$$

$$u[g_a]v; \dots \Rightarrow \text{ACC} \quad \text{where and}$$

$$u[g_i]av; \text{REST} \Rightarrow \text{REST}; u'[g_i]v' \quad u', v' = \text{top}(u, a, v, b, i)$$

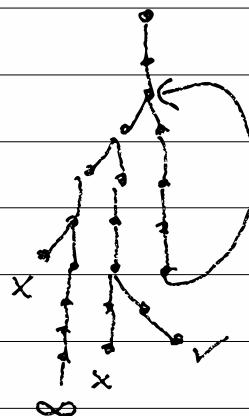
$$u[g_i]av; \text{REST} \Rightarrow \text{REST}; u[g_L]av; u[g_R]av \\ \text{iff } \delta(g_i, a) = (g_L, g_R)$$

$$u[g_r]v; \text{REST} \Rightarrow \text{REST}$$

$$"u[g_i]v; x[g_j]y" \xrightarrow{\alpha} \quad \Gamma' = \Gamma \cup Q \cup \{j\}, \dot{u}$$

$[st] \ u[g_i]v \ j \ x[g_j]y$

20-3/ Backtracking nondet TM



breadth-first-search

$$\text{rop} = (0 \cup 1)^*$$

consume 1 length of rope



lookat | exor-and
of rope

c, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110,

MTM: tape 0 = input

tape 1: current rope

tape 2: current simulation

composition

$$\begin{array}{ccc} z_0-y & f(x) & g(y) \\ & \swarrow \downarrow \searrow & \\ & g(f(x)) & \end{array}$$

PL for TM:

$e =$ By-hand

	e ∪ e		e ∩ e
	e ∘ e		e*
	e(e)		

enumerator = . . .

normal = $(Q, \Sigma, \Gamma, q_0, \delta, q_a, q_r)$

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

$\xrightarrow{q_0}$
 $\{q_a, q_r\}$

enum = $(Q, \Sigma, \Gamma, q_0, \delta, q_p)$

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

$\xrightarrow{q_p}$

If $\varepsilon[q_0]z \Rightarrow^* u[q_p]v$, then $v \in L(\text{enum})$

enum \rightarrow TM :

TM \rightarrow enum !