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

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

which statements are true?

"All birds have wings"

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

Defining the set of true stmts

Making a decision procedure

Generating a list

A statement is a string of characters from an alphabet

some finite set

Σ

A finite set is one where you can write down all the elements! $S = \{$Pikachu, Charmander, Squirtle, Bulbasaur$\} = \{C, P, B, S\}$

A string of Σ is a finite sequence of Σs

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

1-3/ A language is ~~on~~ ~~the~~ set of strings

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

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

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

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

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

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

⤷ complement or negation of Y

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

$PP \circ BC = PPBC$

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

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

$PPC \in$

lexicographic ordering of Σ is a sequence

$lo(\{0,1\}) = \underbrace{\varepsilon}, \underbrace{0, 1}, \underbrace{00, 10, 01, 11},$

$\underbrace{000, 001, 010, 011, 100, 101, 110, 111},$

$\ldots$

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

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

lexi 0 = ε   lexi 1 = 0   lexi 2 = 1

lexi 3 = 00

lexi n =   sz = size of Σ

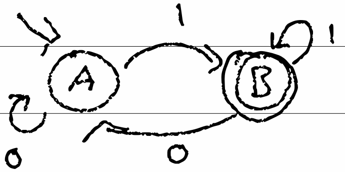
```
if n < $sz^0$ then ret ε                     of len 1
    $(n - sz^0) < sz^1$ then convert $(n-sz^0)$ into a string
  $(n-sz^0) - sz^1 < sz^2$        convert of len 2
```

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

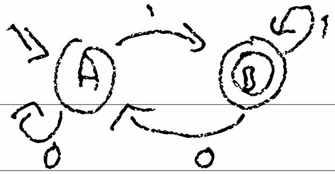
$\varepsilon \in \Sigma^*$ PPCBSPPPP $\in \Sigma^*$

010111 $\in \Sigma^*$

Determinisitic Finite Automata (DFA)



2-1) DFA



◎ means Yess

○ means No

0110 = No

0111 = Yes

0010 = No

1101 = Yes

ε = No

1 = Yes

11 = Yes

00 = No

transition function (edges)

$Q \times \Sigma \to Q$

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

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

Σ: $\{0,1\}$

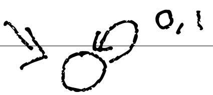
q_0: startastate $= A$

F: accepting states $\{B\}$

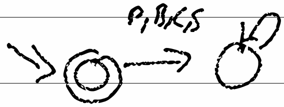
	A	B
0	A	A
1	B	B

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

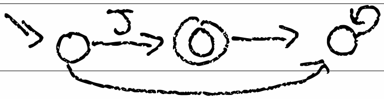
No string DFA :



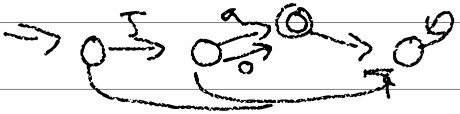
only empty string :



only the string 'J' :



'Ja' and 'Jo'



$: Q \times \Sigma \to Q$

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

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

accepts? d ε = is q_0 in F ?

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

accepts d $(c :: s)$

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

accepts d s = helper d $d.q_0$ s

helper d q_i ε = $q_i \in d.F$

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

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

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

↙ trace

2-3) 0110 ⇒ Even, Odd, Odd, Even

Transducers are DFAs where trace matters

Moore machines

L(d) = the language of DFA d

= { s | accepts d s = true }

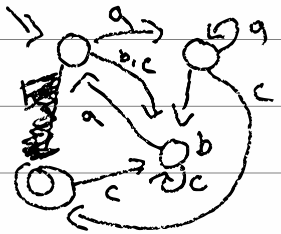
may be infinite

Given a DFA, return a string that would be accepted

example : DFA ⇒ Σ^* or false

s.t. if example d returns s then

accepts? d s = true



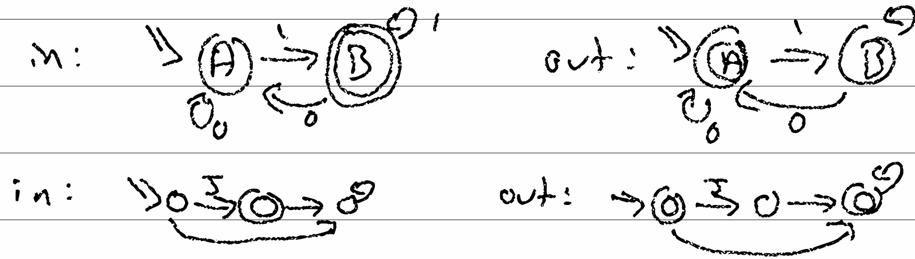
'b'

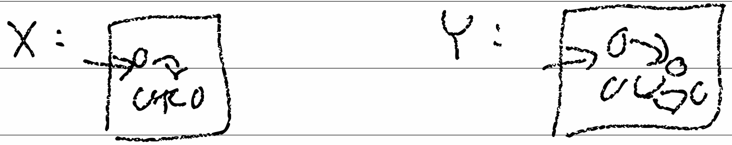
2-4/ Suppose that d is a DFA, construct d' where

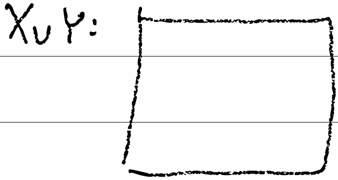
$L(d') = \overline{L(d)}$ (i.e. d' says yes when d says no & vice versa)

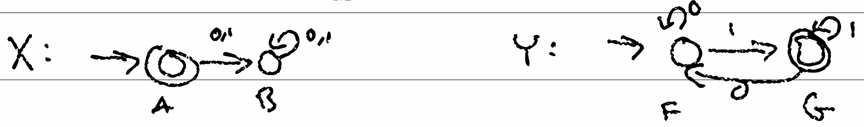
negate : DFA → DFA

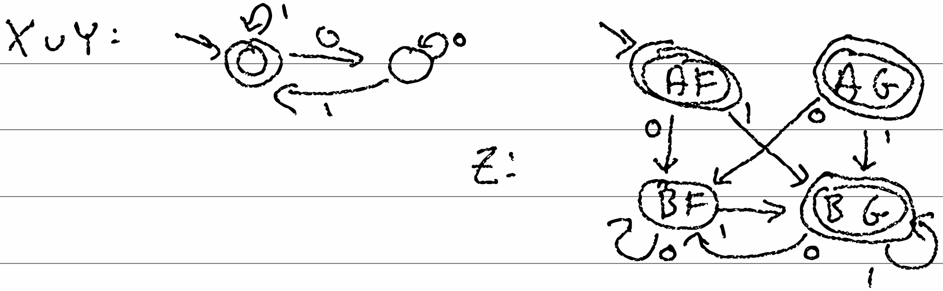
negate (Odds) = Evens











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

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

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

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

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

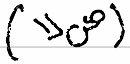
and to make intersect

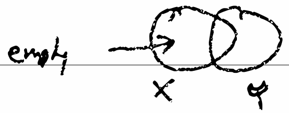
$z.\delta((q_x, q_y), c)$
$= (x.\delta(q_x, c), y.\delta(q_y, c))$

$X \subseteq Y$ (subset) iff
$\forall q \in X.\ q \in Y.$

$X = Y$ iff
$X \subseteq Y$
and $Y \subseteq X$

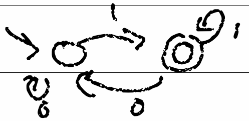
Subset? : DFA × DFA → bool

Subset? () X = Yes
(epsilon) (Evens) = Yes
(epsilon) (odd) = No

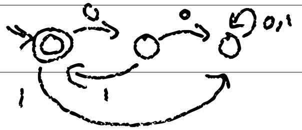


$X - Y$ must be empty
$X \cap \overline{Y}$ is empty

3-1 / $z \in X \circ Y$ iff $z = x \circ y$ where $x \in X$ and $y \in Y$

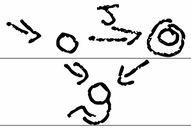
$X =$ 

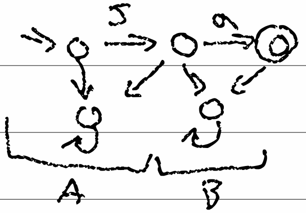
$x \in X = 0111$

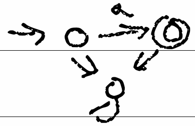
$Y =$ 

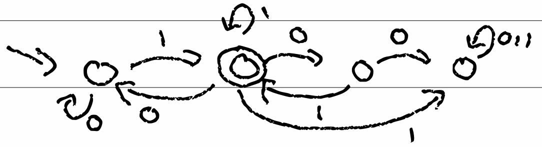
$y \in Y = 010101$

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

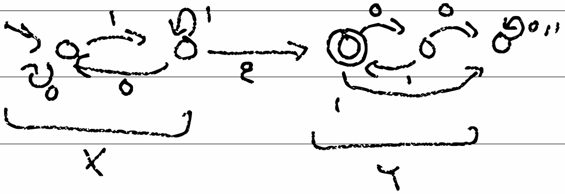
$A =$ 

$A \circ B =$ 

$B =$ 



// illegal, not a DFA!

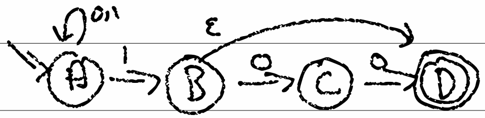


$F \xrightarrow{\epsilon} G$
(skip from F to G at the right time)

3-2/ NFA - non-deterministic finite automata

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

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



δ	A	B	C	D
0	{A}	{C}	{D}	{}
1	{A,B}	{}	{}	{}
ε	{}	{D}	{}	{}

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

$x \notin L(n)$ iff

∀t. oracle n t = ff

(or "; f")

trace = a sequence of $Q \times \Sigma$ or ε

(A,0) (B,1) (C,0) (D,0) 0100 ∈ L(n)

(A,1) (A,0) (B,1) (D,ε) 101ε = 101 ∈ L(n)

oracle interpretation : NFA × trace → boolean

oracle N t = helper N N.q_0 t

helper N q_i [] = q_i ∈ N.F

((q_i, c) :: t') =

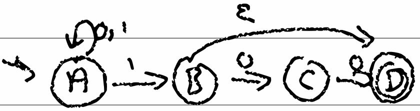
is q_j ∈ N.δ(q_i, c), then helper N q_j t'

o.w. "invalid trace"

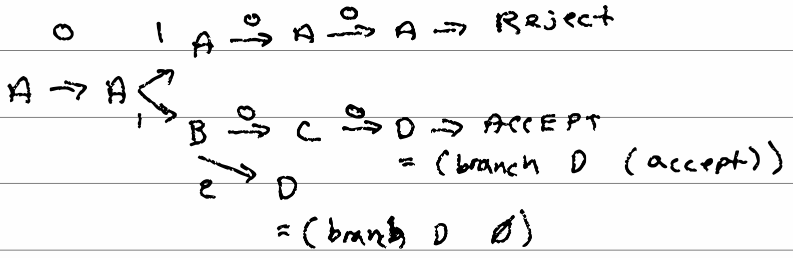
3-3/ trace-tree = accept | reject
| branch state (tt, ...)

↑ set

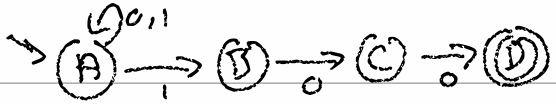
```
all : NFA × $\Sigma^*$ → tt
all N s = helper N N.$q_0$ S
helper N $q_i$ s =
    branch $q_i$ ( case s where
        [] → if $q_i \in$ N.F then
                 { accept }
             o.w
                 { reject }
        c :: s' →
             { tt | tt = helper N $q_j$ s'
                 where $q_j \in$ N.$\delta(q_i, c)$ } )
    ∪ { tt | tt = helper N $q_j$ s
          where $q_j \in$ N.$\delta(q_i, \varepsilon)$ and $q_i \neq q_j$ }
```



0100

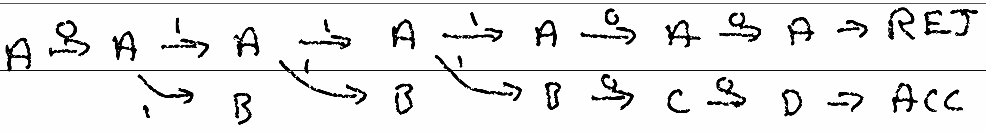


3-4/



"ends in 100"

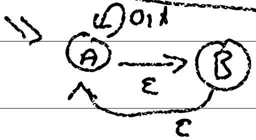
011100



```
backtrack : NFA x String $\Sigma^*$ → Bool
backtrack N s = helper N N.$q_0$ s
helper N $q_i$ s =
    OR (case s with [] → $q_i \in$ N.F
            c::s' → $\mathrm{OR}_{q_j \in N.\delta(q_i,c)}$ helper N $q_j$ s' )

    $\mathrm{OR}_{q_j \in N.\delta(q_i,\varepsilon)}$ helper N $q_j$ s
```

x=3 ; ($\underline{1}$ || x++) ; x==3,



maybe DS = tree

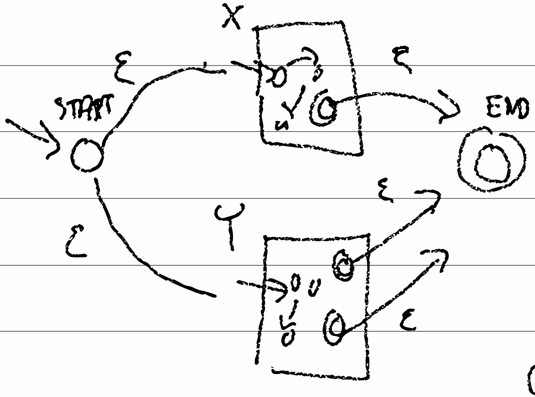
unfold : A → DS(B)

fold : DS(B) → C   Haskell

there always exist

combined : A → C

4-1



$X = (Q_x, \Sigma, q_{0x}, \delta_x, F_x)$

$Y = (Q_y, \Sigma, q_{0y}, \delta_y, F_y)$

$Z = (Q_z, \Sigma, q_{0z}, \delta_z, F_z)$

$F_z = \{END\}$

$q_{0z} = START$

$Q_z = \{START, END\}$

$\cup \; Q_x \times \{0\}$

$\cup \; Q_y \times \{1\}$

$\delta_z(q_i, c) =$

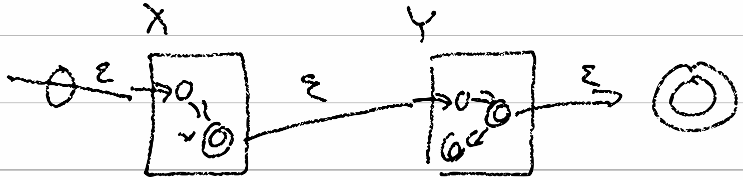
$(START, \varepsilon) = \{(q_{0x}, 0) \; (q_{0y}, 1)\}$

$(START, \_) = \{\}$

$(END, \_) = \{\}$

$((q_x, 0), c) = \delta_x(q_x, c) \times \{0\}$

$q_y, 1$ similar $\quad \cup$ if $q_x \in F_x$ and $c = \varepsilon$, $\{END\}$ o.w. $\{\}$

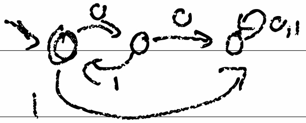


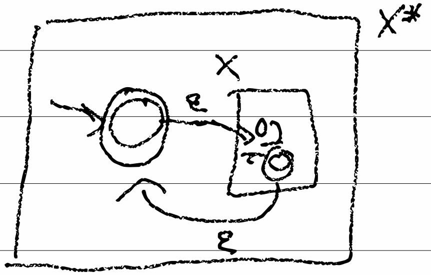
4-2/ Kleene-star X^*

$z \in X^*$ iff $z = \varepsilon$ OR $z = xy$
where $x \in X$ and
$y \in X^*$

iff $z = x_0 \dots x_n$
where $x_i \in X$

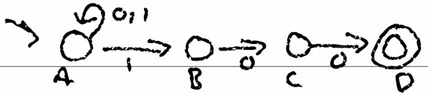
$(\{0\} \circ \{1\})^*$ = any number of 01 sequences



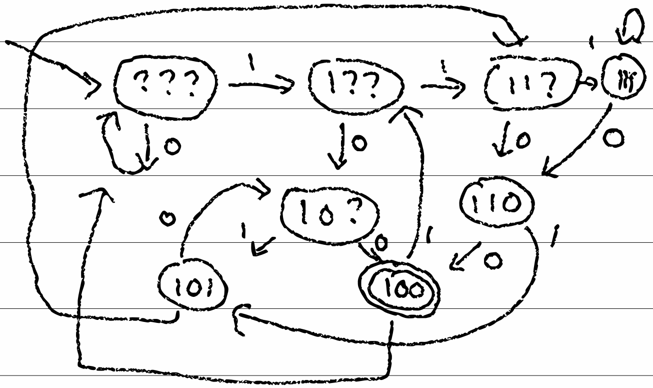


DFAs : ∪, ∩, $\overline{}$ DFA → NFA

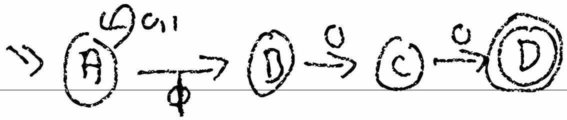
NFAs : ∪, ∘, * want : NFA → DFA

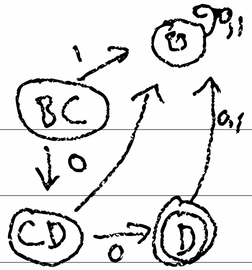


01100



4-3





01100

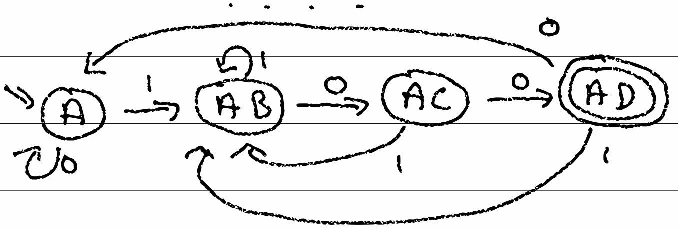
$A \xrightarrow{0} A \xrightarrow{1} A \xrightarrow{1} A \xrightarrow{0} A \xrightarrow{0} A$

B ~~[illegible]~~ B   C   D ← ✓

???   ???   1??   11?   110?   100

∅   {A}   {B}   {C}   {D}

. . . . .



01100	✓
011001	X
1111	✗
111100	✓

NFA ⇒ DFA   in : $(Q_N, \Sigma, q_{0n}, \delta_N, F_n)$

out : $(Q_D, \Sigma, q_{0D}, \delta_D, F_D)$

$Q_D = P(Q_N)$ $q_{0D} = E(\{q_{0N}\})$ $F_D = \{q_d \mid F_n \cap q_d \neq \emptyset\}$

$$\delta_D(q_d, c) = E\Big(\bigcup_{q_n \in q_d} \delta_N(q_n, c)\Big)$$

$$E : Q_D \Rightarrow Q_D \quad = \quad E(\underset{q_d}{\cancel{x}}) = \underset{q_d}{\cancel{x}} \cup E\Big(\bigcup_{q_n \in q_d} \delta_N(q_n, \varepsilon)\Big)$$

4-4/

$\underbrace{\forall x \in \Sigma^*}_{\text{randomstring}}$. ~~acce~~ backtrack N x

= accepts (NFA→DFA N) x

= pick a random number

(lexi n)

write the NFA manually as a DFA...

use the dfA equality checker to see that

dfa-equal? manual (N→D N) = #true

Then

5-1/ regular expression := $x \cup y$ regular operations

$x \cap y$

x,y r.e $x \circ y$

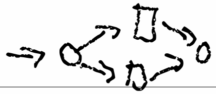
$\overline{x}$

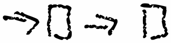
x^*

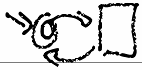
$\emptyset$

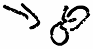
ε

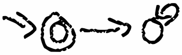
$c \in \Sigma$

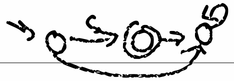




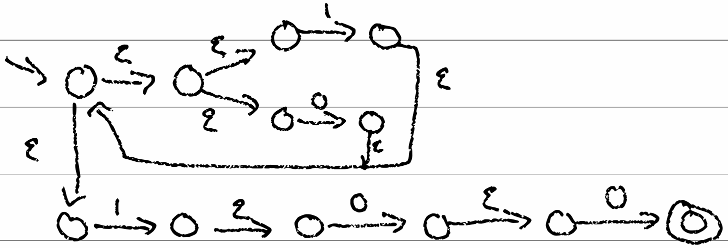




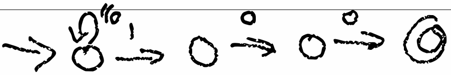




$('1' \cup '0')^* \circ '1' \circ '0' \circ '0' = (1 \cup 0)^* 100$



=



$*.c \Rightarrow \Sigma^* \circ '.' \circ 'c'$

5-2/ interface RegEx;

```
class  Empty   implents  RegEx;    ()
       Epsilon           RegEx     ()
                                   char
       Char              RE        (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')))]
```

$(1 \cup 0)^* 100$

```
interface RegEx { NFA compile(); }

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

5-3 / generate : RE → Σ^* s.t.

accepts? (nfa->dfa (compile r)) (generate r) = true

generate $\emptyset$ = error

gen ε = ε

gen c = c

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

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

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

heads → gen x

tails → gen y

printall : RE → void

printall r = helper r print

helper $\emptyset$ pr = ~~er~~ (void)

h ε pr = pr ε

h c pr = pr c

h $x \circ y$ pr = h x new-print

(new-print s = h y np2

np2 t = pr $s \circ t$)

= h x (lambda s: h y (lambda t: pr $s \circ t$))

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

h $(x \cup y)$ pr = h x pr ; h y pr

5-4

numbers

$x \cdot 0 = 0$ $\qquad x + y = y + x$ $\qquad x \cdot 1 = x$

regex

$x \circ \emptyset = \emptyset = \emptyset \circ x$ $\qquad \emptyset = \{\}$

$x \circ \varepsilon = x = \varepsilon \circ x$ $\qquad \varepsilon = \{\varepsilon = ""\}$

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

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

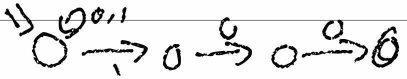
$\emptyset^* = \varepsilon$ $\qquad x^* = \varepsilon \cup x \circ x^*$

$\varepsilon^* = \varepsilon$ $\qquad \varepsilon^* = \varepsilon \cup \varepsilon \circ \varepsilon^*$ $\qquad = \varepsilon \cup \emptyset \circ \emptyset^*$

$(x^*)^* = x^*$ $\qquad \varepsilon \cup \varepsilon^*$ $\qquad = \varepsilon \cup \emptyset = \varepsilon$

$x \cup z = z$ if $x \leq z$

$$\begin{tikzcd}
\text{DFAs} \arrow[rr, leftrightarrow] \arrow[dr, bend right] & & \text{NFAs} \\
& & \text{REX} \arrow[u] \arrow[ull, dashed]
\end{tikzcd}$$

1) 

$\Longrightarrow (1 \cup 0)^* 100$

G-1/

$\mathrm{NFA}(k) \xrightarrow{\text{IN}} \mathrm{GNFA}\ (2+k)$

RIP $\left\{ \begin{array}{l} \rightarrow \mathrm{GNFA}\ (2+k-1) \\ \rightarrow \mathrm{GNFA}\ (2+k-2) \\ \quad \cdots\cdots \end{array} \right]$ — k times

$\rightarrow \mathrm{GNFA}\ (2)$

$\xrightarrow{\text{OUT}} \mathrm{REX}$

NFA: $\circ \xrightarrow{c} \circ$

$\circ \xrightarrow{\varepsilon} \circ$

GNFA: $\circ \xrightarrow{r} \circ$

GNFA (generalized NFA)

$= (Q, \Sigma, q_a, \Delta, q_f)$

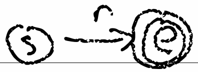
↑one state, not a set

$\Delta : (\underbrace{Q}_{(Q-q_f)} \times \underbrace{Q}_{(Q-q_a)}) \rightarrow \mathrm{Rex}$

You can't leave qf

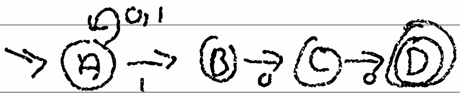
You can't go back to qa

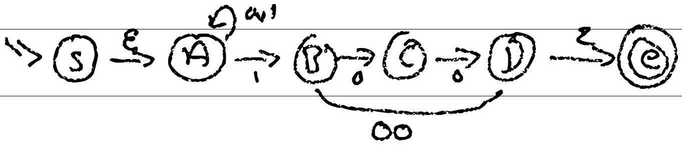
$\delta : Q \times \Sigma \rightarrow P(Q)$



$\Delta(q_i, q_j) = r$ ~~[illegible]~~ if $x \in r$ then $q_i \xrightarrow{x} q_j$

in the NFA

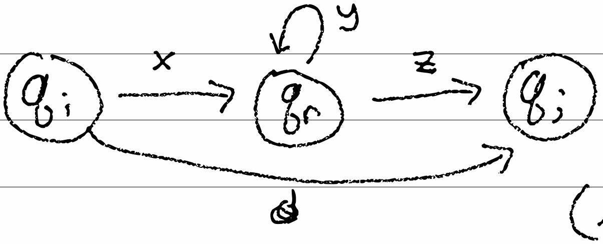




6-2/ Rip : GNFA(n+1) ⇒ GNFA(n)

$\{q_0, q_f, q_r, q_i \ldots\} \rightarrow \{q_0, q_f, q_i \ldots\}$

q_r is gone



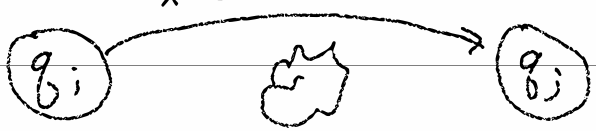
$(x \circ y^* \circ z) \cup d$

⟹

...

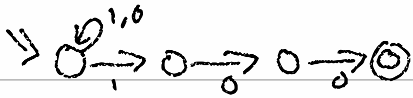
~~$x \circ z \cup d$~~ $x \circ y^* \circ z \cup d$

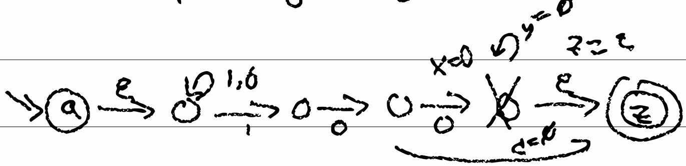
~~$x \circ z$~~



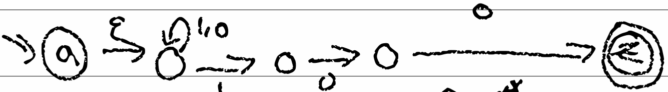
G-3/

IN





RIR

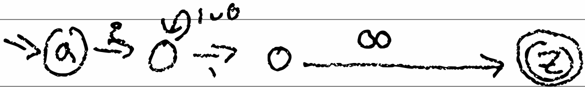


$xy^* z \cup d$

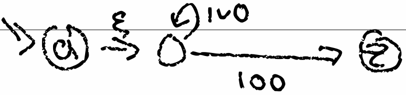
$0 \cdot \emptyset^* \circ \varepsilon \cup \emptyset$

$= 0$

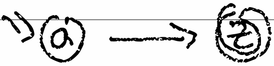
RIP



RIP



RIP

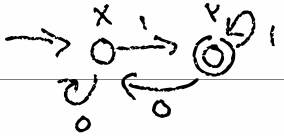


$\varepsilon \circ (1 \cup 0)^* \circ 100 \cup \emptyset$

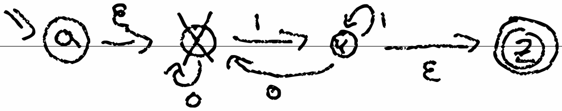
OUT

$= (1 \cup 0)^* \circ 100$ ←

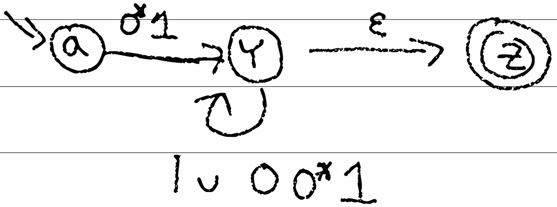
6-4

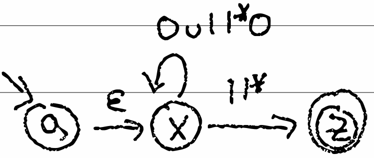


⇓ IN



⇓ RIP





⇓ RIP

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

⇓ RIP

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

//

||

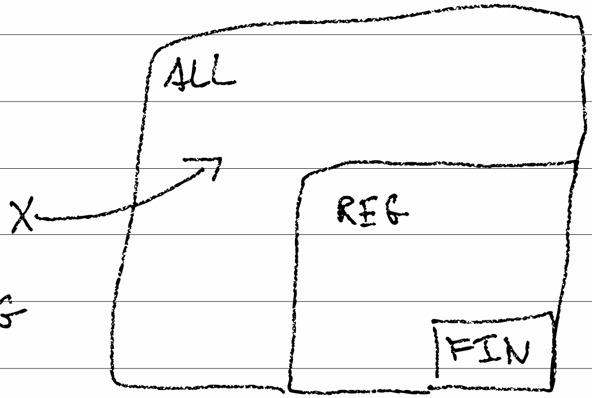
$(0 \cup 1)^* 1$

DFA ⇄ REX

∀d. let r = dfa→rex d

accepts? d (generate r) = true

G-5)



$X \notin REG$

REG = language described by DFA, NFA, or RE

FIN = finite languages

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

$ALL = \{ \{\},$ $\Sigma^* = \varepsilon, 0, 1, 00, 01, 10, 11, \ldots$

$\{0, 1\}, \{0, 00, \varepsilon, 000, 0000, \ldots\}$

{all of Jay's lectures}

{JPEGs of cats}, {JPEGs of road signs}

... }

REG = ALL?

2-1/ $$\exists \underset{\substack{\nearrow \\ \text{language} \\ \text{(some problem)}}}{X} \in \underset{\substack{\uparrow \\ \text{all possible} \\ \text{languages}}}{ALL}.\quad X \notin \underset{\substack{\uparrow \\ \text{languages defined} \\ \text{by DFAs}}}{REG}$$

option 1: $\forall Y \in REG.\ X \neq Y$

option 2: $\forall Y \in REG.\ P(Y)$

$\neg P(X)$

mystery #1: What is X? witness

#2: what is P? property

$\Longrightarrow$

proof 1: $\forall Y \in REG.\ P(Y)$

proof 2: $\neg P(X)$

$\Longrightarrow$

$X \in ALL$, but $X \notin REG$

$\Longrightarrow$

computers aren't omnipotent

alphabet

7-2/ DFAs .. $(Q, \Sigma, q_0, \delta, F)$

finite set of states

start somewhere

many stop

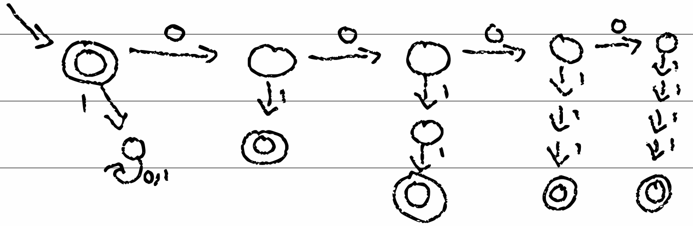
$Q \times \Sigma \to Q$

EQ

$\varepsilon \in EQ$ $\quad 01 \in EQ$ $\quad 0011 \in EQ$

$0 \notin EQ$ $\quad 010 \notin EG$ $\quad 000111 \in EQ$

$0110 \notin EQ$ $\quad 0000\,1111 \in EQ$



$= EQ^4$

$|EQ^0| = 2$ $\quad |EG^1| = 4 = |EQ^0| + 2$ $\quad |EG^3| = 11 = |EG^2| + 4$

$|EQ^2| = 7 = |EG^1| + 3$ $\quad |FQ^n| = |EQ^{n-1}| + n+1$

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

$\forall n.\ 0^n 1^n \in EQ$

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

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

7-3/

```
int count = 0;   char c;
while ( c = getc(); c=='0')
    count ++;
ungetc(c);
while ( c = getc(); c == '1')
    count --;;
return count == 0;
```

$\Longrightarrow$

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

$m = 2^{2^{33}}$

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

$=$

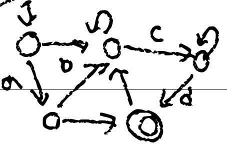
X

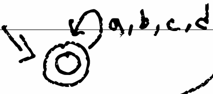
step 1: what is X? ✓

7-4/ What is P?

abcd ∈ X

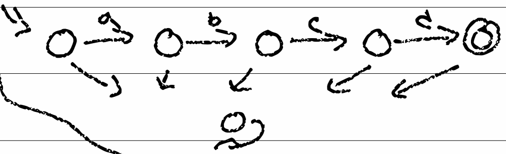
5

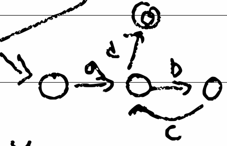




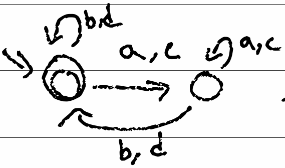
1

6



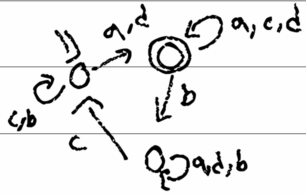


4



2

3



1: aaaabcd ∈X

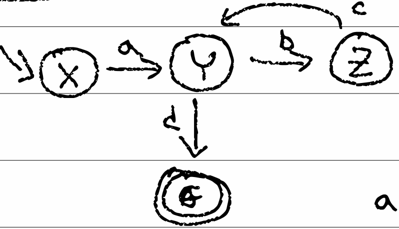
3: abcabcabcd ∈X

2: ababababcd ∈X

4: abcbcbcd ∈X

5: x

6: x



abcd string (i=1)

X Y Z Y G trace

abcbcbcd string (i=3)

X YZYZYZYG trace

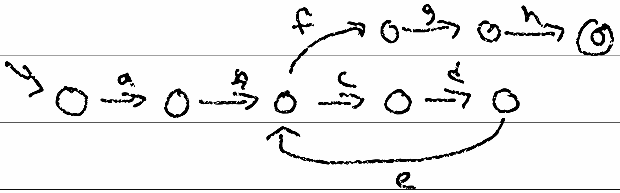
$a(bc)^i d \in$ DFA

for all $i \in [0, \infty)$

ad string (i=0)

X YG

7-5/ DFAs must contain cycles!



If $s \in$ DFA, and s is longenough ($|s| \geq |Q|$), then s must visit some state $^{=q_r}$ twice!

ex: $s = abcd$ s visited Y twice $|Q| = 4$

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

where x goes from q_0 to q_r

(y isn't empty $\neq \varepsilon$) 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 i

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

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

7-6/ Regular Pumping Property (RPP)

$$
\begin{aligned}
&RPP(A : \text{Language}) :=\\
&\quad \exists p \in N.\\
&\quad\quad \forall (s \in A \mid \ |s| \geq p)\\
&\quad\quad\quad \exists (x, y, z \in \Sigma^* \mid \ |xy| \leq p\\
&\quad\quad\quad\quad\quad\quad\quad\quad \wedge\ |y| > 0)\\
&\quad\quad\quad\quad \forall i \in N.\\
&\quad\quad\quad\quad\quad xy^i z \in A.
\end{aligned}
$$

Pumping Lemma: $\forall A \in REG.\ RPP(A).$

$p = |Q|$, x is the string before q_r

y is from q_r to q_r

z is from q_r to $q_f \in F$

Step 2: What is P? ✓

Step 3: $\forall A \in REG.\ P(A).$ ✓

Step 4: $\neg P(EQ)$ ...

8-1) $\neg RPP(EQ)$ $\qquad$ $\neg(A \wedge B) = \neg A \vee \neg B$

$= \forall p \in N.$ $\qquad$ $\neg(A \vee B) = \neg A \wedge \neg B$

$\exists(s \in A \mid |s| > p)$ $\qquad$ $\neg \forall x, P(x) = \exists x, \neg P(x)$

$\forall(x, y, z \in \Sigma^* \mid$ $\qquad$ $\neg \exists x, P(x) = \forall x, \neg P(x)$

$|y| > 0$

$|xy| < p)$ $\qquad$ $\forall$ = you don't choose it

$\exists\ i \in N$ $\qquad$ $\exists$ = you DO choose it

$xy^i z \notin A$

$\neg RPP(EQ)$

given p.

choose $s \in EQ$ where $|s| > p$

$s = 0^p 1^p$

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

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

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

choose i $\quad$ $\boxed{i = 0}$

$xy^i z \notin EQ$ $\qquad$ $xy^i z = 0^a 0^{b \cdot i} 0^c 1^p \notin EQ$

iff $\quad a + bi + c \neq p$

$a + bi + c \neq a + b + c$

$bi \neq b$

$i \neq 1$

8-2/ $s = \overset{\in A}{xyz}$ and $xy^iz \in A$

then

$xy^*z \in A$

regular expression

$xy^*z \subseteq A$

$ALL \not\subseteq REG$ because $EQ \in ALL$

$EQ \notin REG$

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

$MEQ^1 = 0^1 \circ EQ$

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

$000 \circ 0011 \in MEQ^3$

9-1/ EQ ⇒ $0^n 1^n$ for some n

$$\begin{tikzcd}
\text{① machine DFA} \arrow[r, bend left] & \text{② "programming language" Regular expressions} \arrow[l, bend left] \arrow[d, Rightarrow] & \text{REG} \\
\text{machine PDA} & \text{"PL" ① CFG} \arrow[l, bend right] & \text{CFL}
\end{tikzcd}$$

CFG : context-free grammar substitution

$S \to \varepsilon$ RHS → RHS – rule

$\underbrace{S \to 0S1}_{\text{a CFG for } 0^n1^n}$ LHS is always a production derivation

V={S} "variable" (non-terminals)/symbols

$S \to \varepsilon \mid 0S1$ RHS is a string of vars + terminals

$\Sigma = \{0, 1\}$ First var is "start symbol"

E → N | E O E

N → '0' | '1' | '2'

O → '+' | '-' | 'x' | '÷'

q2/ membership $x \in A$

defn $A = \{ \ldots \quad . \quad \sim \}$

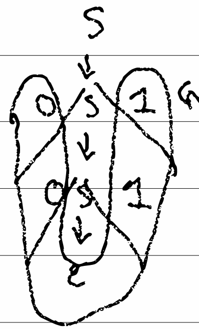
generation A produces $s_1, s_2, s_3, \ldots$

$S \Rightarrow \varepsilon \mid 0 S 1 \quad = g$

$S \rightarrow 0S1 \rightarrow 00S11 \rightarrow 0011 \in L(g)$

$(S, [0,$
$\quad (S, [0,$
$\qquad (S, []),$
$\qquad 1]),$
$\quad 1])$

=



= parse tree

$PT = Var \times (\Sigma + PT)^*$

```
pt2str   (V, seg)      =  seg2str seg
seg2str  []            =  ε
seg2str  c∈Σ :: seg    =  c :: seg2str seg
seg2str  PT::sq        =  pt2str PT ++ seg2str seq
```

9-3/ ~~cfggen g~~

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

$\swarrow$ some set $\quad\downarrow$ alphabet $\quad\downarrow\quad\updownarrow \in V$

$R : V \Rightarrow$ Set of $(V \cup \Sigma)^*$ $\qquad P(V \times (V \cup \Sigma)^*)$

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

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

$(V \cup \Sigma)^* = \varepsilon, 0, 1, S, 00, 01, 0S,$
$10, 11, 1S, S0, S1, SS, \ldots$

$V \times (V \cup \Sigma)^* = (S, \varepsilon) \ (S, 0) \ (S, 1) \ \ldots$

$P(\nearrow) = \{\} \ \{(S, \varepsilon)\}, \ \{(S, \varepsilon), (S, 0)\}$

$\{(S, \varepsilon), (S, 0S1)\}$

```
cfggen g = helper g g.S
helper g v =
    let rules = g.R v
        rhs = random rules
    return ( v, map over s in rhs:
                    if s ∈ V, helper g s
                    o.w.      s )
```

9-4/ all_n_deep g n = helper g n g.S

all_n_deep 0^n1^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 )
```

$S \Rightarrow 0S1 \Rightarrow 00S11$

REG = a language defined by some DFA

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

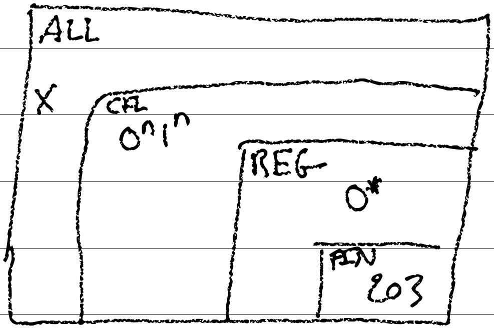
$VS \rightarrow S\ O\ V$

$S \rightarrow NP \mid ANP \mid A\ A\ Np \mid$

$Np \rightarrow N \mid PN$

$V \rightarrow \ldots \mid Adver\ V$

9-5



$0^n1^n \notin REG$

$0^n1^n \in CFL$

$REG \subsetneq CFL$

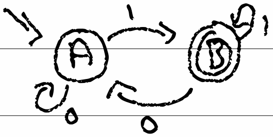
$X \subset Y$ iff $\forall a \in X.\ a \in Y$

$a \in REG$ iff ~~Regex~~ $\exists d \in DFA.\ L(d) = a$

$a \in CFL$ iff $\exists g \in CFG.\ L(g) = a$

$$\underbrace{\forall d \in DFA.}_{\forall \underbrace{}_{args}} \underbrace{\exists g \in CFG.}_{\exists \underbrace{}_{result}} L(g) = L(d)$$

is a fun



$A \Rightarrow \overset{\textcircled{1}}{1B} \mid \overset{\textcircled{2}}{0A}$

$B \Rightarrow \overset{\textcircled{3}}{\varepsilon} \mid \overset{\textcircled{4}}{1B} \mid \overset{\textcircled{5}}{0A}$

$A \overset{1}{\Rightarrow} 1B \overset{4}{\Rightarrow} 11B \overset{5}{\Rightarrow} 110A \overset{2}{\Rightarrow} 1100A \overset{1}{\Rightarrow} 11001B \overset{3}{\Rightarrow} 11001$

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

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

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

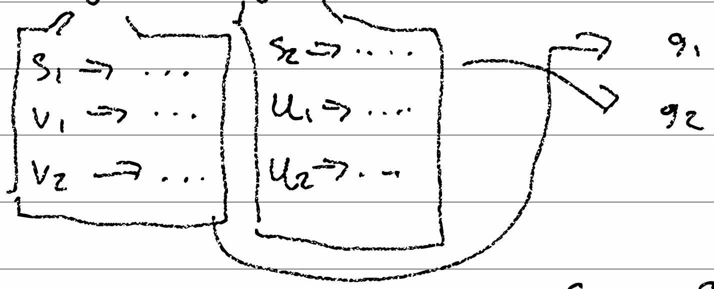
If $\delta(q_i, c) = q_j$, then $R \ni q_i \Rightarrow c q_j$

If $q_i \in F$, then $R \ni q_i \Rightarrow \varepsilon$

9-6/ dfa-accepts d
(pt2str (cfggen (dfa2cfg d))) = true

10-1/ regular operations: ∪, ∩, ∘, *, ‾
context-free ops: ∪, ∘, *

$g_1 \cup g_2 \quad = \quad S \to (0, S_1) \mid (1, S_2)$



$g_1 \circ g_2 \quad = \quad S \to S_1 S_2$

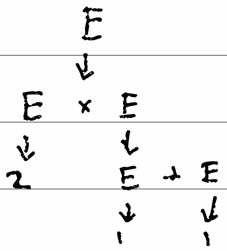
$g_1{}^* \quad = \quad S \to \varepsilon \mid S_1 S$

10-2/ $s \in L(g)$ iff $\exists pt \in g\ str(pt) = s$

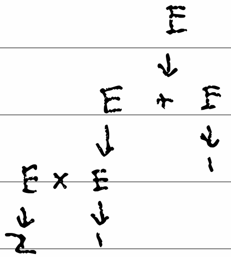
$$E \to E \times E \mid E + E \mid 1 \mid 2$$

$2 \times 1 + 1$

pt 1:



pt 2:



Ambiguous = $\exists s.\ \forall pt_1, pt_2,\ str(pt_1) = s$
$\wedge\ str(pt_2) = s$ ~~$\Rightarrow pt_1 = pt_2$~~

$E \to E + E \mid F$
$F \to F \times F \mid 1 \mid 2$]— unambiguous

ambiguous? : CFG → Bool
deambiguate : CFG → CFG s.t. amb? = #false

LL(k) LALR LR

10-3/ $V \to a\ b\ F\ c\ d\ G\ H\ e\ V$ //complex

$x + y = y + x$

$0 + y = y = y + 0$ ✓

$(1+n)+y = 1 + (n+y) = (n+y)+1 = (y+n)+1$ ✓

assume $n+y = y+n$ $\qquad = y + (n+1)$

$$\begin{tikzcd} \text{CFG} \arrow[r, phantom, "=="] \arrow[d, Leftrightarrow] & \text{NFA} \arrow[d, Leftrightarrow] \\ \text{CNF} \arrow[r, phantom, "=="] & \text{DFA} \end{tikzcd}$$

(Noam)

CNF = Chomsky Normal Form

A grammar g is in CNF iff

If $r \in R$, then $r = A \to BC$ where $B \in V, C \in V$ and $B \neq S$, $C \neq S$

or $r = S \to \varepsilon$

or $r = A \to a$

$a \in \Sigma$

10-4/ $S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \varepsilon$

$S' \to S$ Add a new start sym

$S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \varepsilon$

Remove all εs ($V \to \varepsilon$)

$S' \to S$

$S \to ASA \mid aB \mid a$

$A \to B \mid \varepsilon \mid S$

$B \to b$

$S' \to S$

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

$A \to B \mid S$

$B \to b$

Remove unit rules ($V \to u$)

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

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

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

$B \to b$

Add intermediate vars

$S' \to XA \mid SA \mid AS \mid YB \mid a$

$X \to AS$ $S \to XA \mid SA \mid AS \mid YB \mid a$

$Y \to a$ $A \to b \mid XA \mid SA \mid AS \mid YB \mid a$

$B \to b$

10-5/ $S \to \varepsilon \mid 0S1$ add S'

$S' \to S$

$S \to \varepsilon \mid 0S1$ removed $V \to \varepsilon$

$S' \to S \mid \varepsilon$

$S \to 0S1 \mid 01$ remove $V \to A$

$S' \to 0S1 \mid 01 \mid \varepsilon$

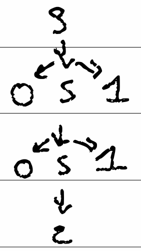
$S \to 0S1 \mid 01$ add them

$S' \to XA \mid BA \mid \varepsilon$

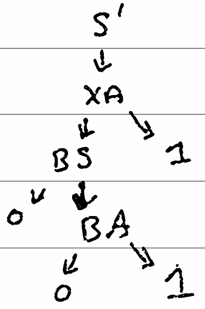
$S \to XA \mid BA$

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

$S' \to XA \to BSA \to 0SA \to 0XAA \to 0BSAA \to 00SAA \to$
$00BAAA \to 000AAA \to 0001AA \to 00011A \to 000111$



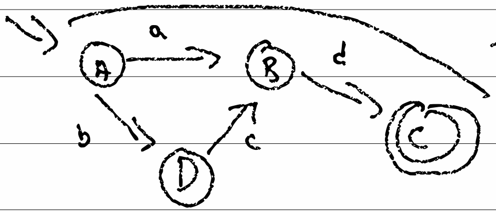
N-ary tree



Binary tree!

CNF parse trees are binary!

11-1/ Generating a string accepted by DFA



→ ad

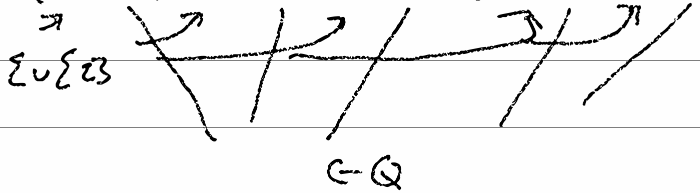
```
gsabh {set of unvisited nodes} x path to here x here
    gsab (DFA d) = gsabh (d.Q - d.q_0) ε d.q_0
  gsabh Remaing Path q_i =
      if q_i ∈ d.F then return Path
      if Remaining is empty then return FALSE
      for (~~q_i...~~) in δ(q_i) :
          (c, q_j)
         if q_j ∈ Remaining :
            p = gsabh (Remaining - q_j) (Path ++ c) q_j
            if p ~~is true~~ false : return p
      return FALSE
```

11-2/ $011\varepsilon0 = 0110$

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

NFA. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to P(Q)$

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



Task #27

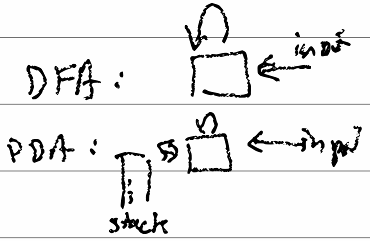
In Task #30, 0110

11-3 | Regular (REG) Context-free (CFL)

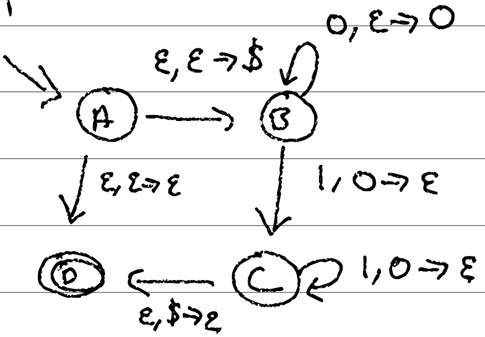
DFAs machine $\boxed{\text{PPA}}$

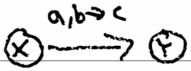
⟺ REX ⟺ PL ⟺ CFG

PDA - push-down automata



$0^n 1^n$





Read a from input

Pop b from stack

Push c to stack

config = stack × Q × input ⟹ stack[Q] input

ε [A] 0011 ⟹ $ [B] 0011 ⟹ $0 [B] 011 ⟹ $00 [B] 11

⟹ $0 [C] 1 ⟹ $ [C] ε ⟹ ε [D] ε ⟹ ✓

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

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

$PDA = (Q, \Sigma \text{ input}, \Gamma \text{ stack}, q_0 \in Q,$ // $, F \subseteq Q)$

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$

$\to P(Q \times (\Gamma \cup \{\varepsilon\}))$

```
11-4/  pda-oracle P (os : List ($\Sigma \cup \{\varepsilon\}$, $\Gamma \cup \{\varepsilon\}$, Q, $\Gamma \cup \{\varepsilon\}$))
    pda-oracle $0^n 1^n$ [ ($\varepsilon$, $\varepsilon$, B, \$),
                           (0, $\varepsilon$, B, 0),
                           (0, $\varepsilon$, B, 0),
                           (1, 0, C, $\varepsilon$),
                           (1, 0, C, $\varepsilon$),
                           ($\varepsilon$, \$, D, $\varepsilon$)]      = true

  pda-oracle P os = helper P P.q0 os $\varepsilon$
  helper (PDA p) (Q $q_i$) (os os) (st st) =
     if os is empty, ret $q_i \in$ P.F
     let [(c, a, $q_j$, b) :: os'] = os
     if P.$\delta$($q_i$, c, a) $\ni$ ($q_j$, b)
      and st = a $\circ$ st'   then
        helper P $q_j$ os' (b $\circ$ st')
    o.w. FALSE
```

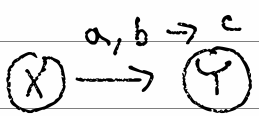
NFA

input	
c	$\{q_0, \dots, q_n\}$
ε	$\{q_2, \dots, q_4\}$

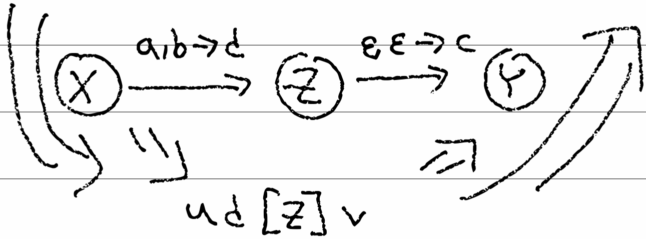
PDA

	stack	
input	a	ε
c	$\{\dots\}$	$\{\dots\}$
ε	$\{\dots\}$	$\{\dots\}$

11-5/ Can a PDA push multiple things?

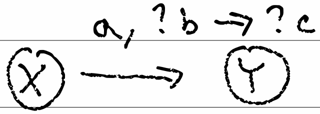


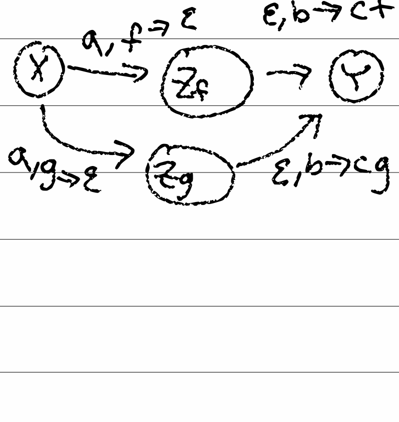
$ub[X]av \Rightarrow udc[Y]v$



Can a PDA read more than 1 back?

f ∈ Γ





$ubf[X]av$ $\quad$ $ubg[X]a$

$ub[Z_f]v$ $\quad$ $ub[Z_g]v$

$ucf[Y]v$ $\quad$ $ucg[Y]v$

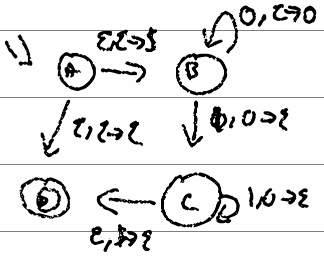
Z_i forall $i \in \Gamma$

11-6

DFA	A	→	a B
CFG	A	→	a B c C d
???	ABC	→	a B c C d
TM	AbCd	→	a B c D

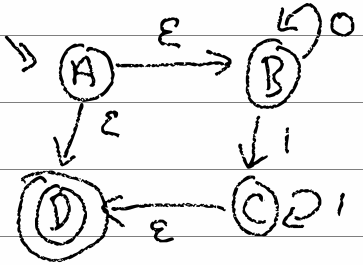
$= Q' = Q \times \Gamma^n$

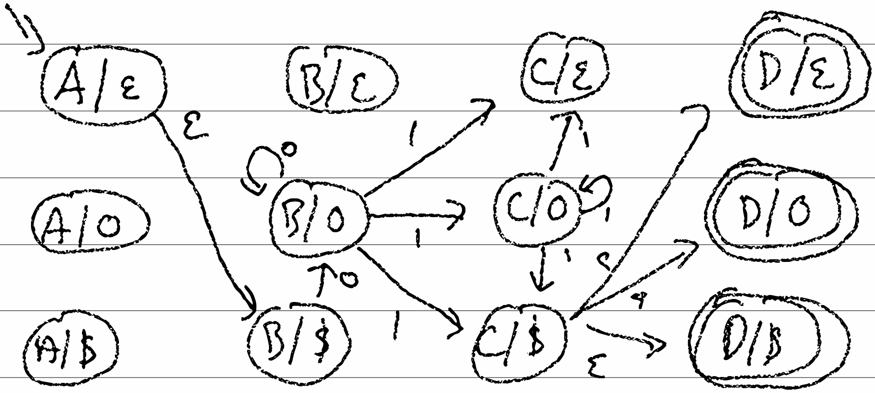
PDA to DFA (~~P~~PDA p) (Nat n)



$\underline{0}\}$

$\varepsilon \cup 0^*11^*$





12-1/ CFG → PDA

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

assume in CNF $R =$ $A \to a$

$A \to BC$

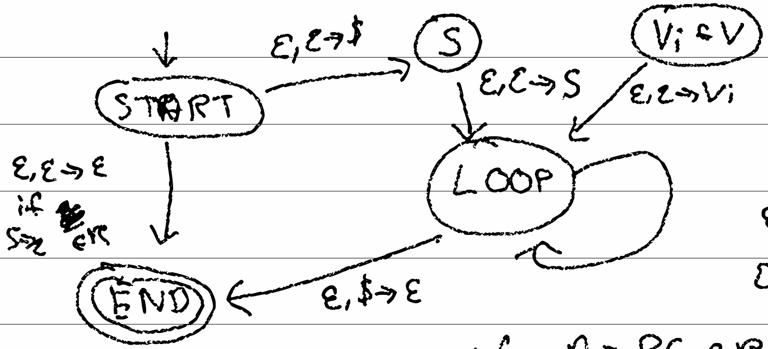
$S \to \varepsilon$

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

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

$q_0 =$ START $F = \{$END$\}$

$Q = \{$START, LOOP, END$\} \cup V$



START
$\varepsilon, \varepsilon \to \$$
S
$\varepsilon, \varepsilon \to S$
$V_i \in V$
$\varepsilon, \varepsilon \to V_i$
LOOP
$\varepsilon, \varepsilon \to \varepsilon$
if $S \to \varepsilon \in R$
END
$\varepsilon, \$ \to \varepsilon$

$a, a \to \varepsilon$ $\forall a \in \Sigma$

$\varepsilon, V \to a$ if $V \to a \in R$

$\varepsilon, S \to \varepsilon$ if $S \to \varepsilon \in R$

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

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

12-2/ $S \to \varepsilon \mid 0S1$

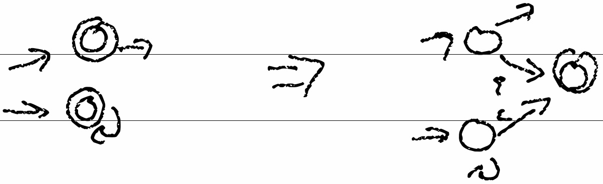
input : 000111

ε [START] 000111 → \$ [S] $0^3 1^3$ → \$ S [LOOP] $0^3 1^3$ →
\$ 1 S 0 [LOOP] $0^3 1^3$ → \$ \$ [LOOP] $0^2 1^3$ → \$ 1 1 S 0 [L] $0^2 1^3$ →
\$ 1 1 S [L] $0 1^3$ → \$ 1 1 1 S 0 [L] $0 1^3$ → \$ 1^3 [L] 1^3 → \$ 1^3 [L] 1^3
\$ 1^2 [L] 1^2 → \$ 1 [L] 1 → \$ [L] → [END] → ✓

12-3 / PDA → CFG

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

assume 1 : $F = \{q_f\}$



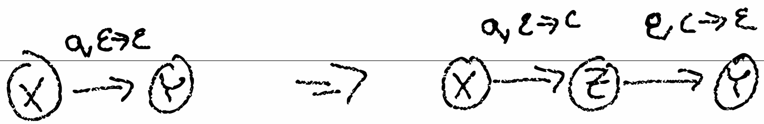
assume 2 : every transitions pushes XOR pops

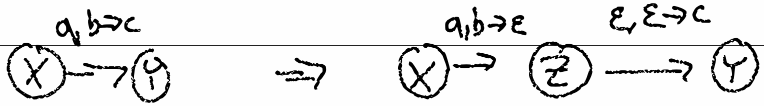
push : $a, \varepsilon \to c$ (pushed c)

pop : $a, b \to \varepsilon$ (poped b)

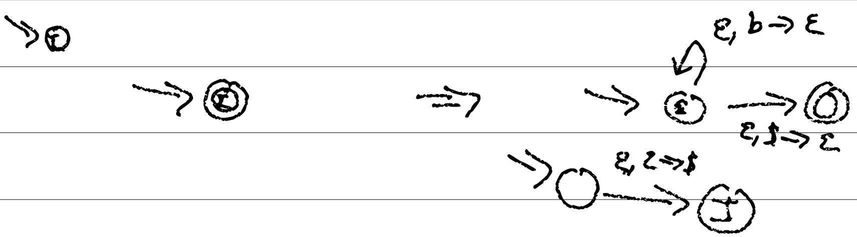
✗ ignore : $a, \varepsilon \to \varepsilon$ (ignore)

✗ replace : $a, b \to c$





assume 3 : the stack is empty on accept



∀b∈Γ, b≠$

$ ∉ Γ

12-4/ CFG $g = (V, \Sigma, R, S)$

$V = Q \times Q$ $\qquad \Sigma = \Sigma$

$S = (q_0, q_f)$

If (q_i, q_j) generates string s

then $\underset{u}{\varepsilon}[q_i]\,st \Rightarrow^* \underset{u}{\varepsilon}[q_j]\,t$

If (q_0, q_f) generates string s and $u = t = \varepsilon$

then $\varepsilon[q_0]\,s \Rightarrow^* \varepsilon[q_f]\,\varepsilon$ ... s is accepted by P

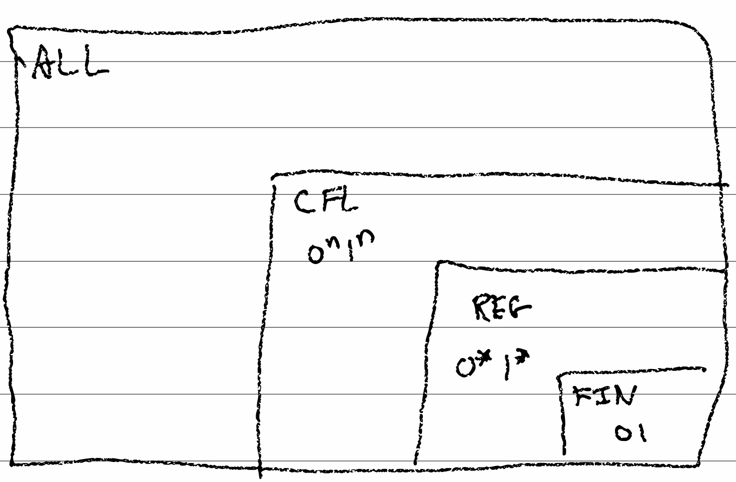
$\forall p \in Q \quad (p,p) \to \varepsilon$ paths are refl

$\forall p,q,r \in Q. \quad (p,q) \to (p,r)\,(r,q)$ paths are trans

$$\begin{array}{c} (r, t) \in \delta(p, a, \varepsilon) \\ (q, \varepsilon) \in \delta(s, b, t) \\ \hline (p,q) \to a\ (r,s)\ b \end{array}$$

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

15-1

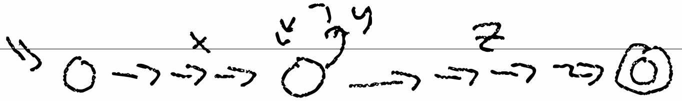


$\text{ALL} \neq \text{CFL} \longleftarrow \exists x \in \text{ALL}.\ x \notin \text{CFL}.$

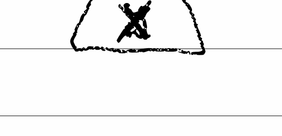
$\longleftarrow \exists P.\ (\forall x \in \text{CFL}.\ P(x))$

$\wedge\ (\exists y \in \text{ALL}.\ \neg P(y))$

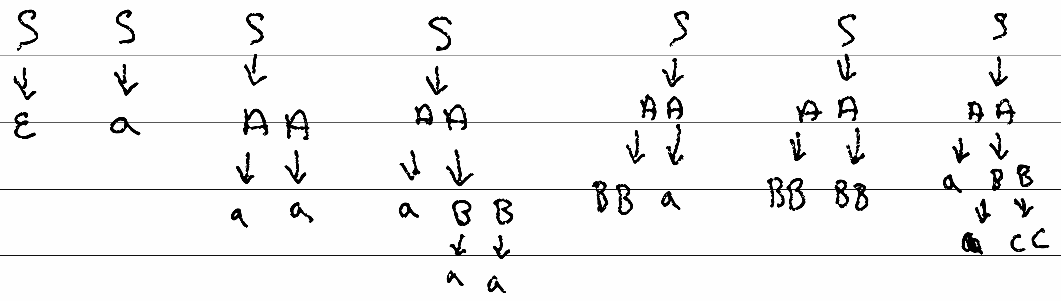
Regular PP. $s = xyz$ and $xy^iz \in A$



Context PP $s = uvxyz$ and $uv^ixy^iz \in A$



15-2/ Suppose G is CFG in CNF



$S \Rightarrow AA \Rightarrow$

How many chars are in a tree of depth k?

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

$V_0 \rightarrow V_1^2 \rightarrow V_2^4 \rightarrow V_3^8 \rightarrow V_4^{16} \rightarrow a^{16}$

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

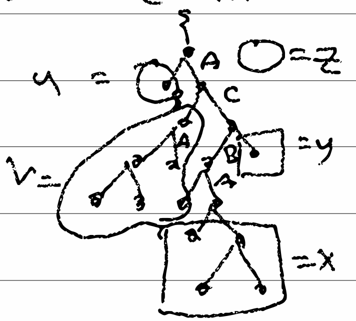
How deep is the tree of a string(accepted) with N chars?

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

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

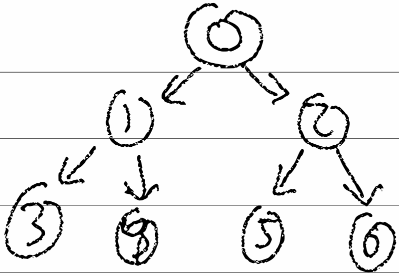


$S \Rightarrow uAz$

$A \Rightarrow vAy$

$A \Rightarrow x$

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) =

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

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

$\exists (u,v,x,y,z \in \Sigma^* \mid |vxy| \leq p$ && $|vy| > 0)$

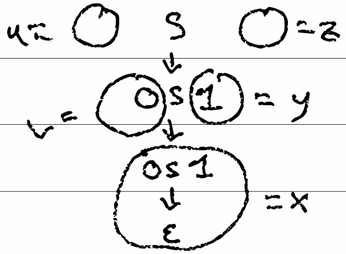
$\forall i \in \mathbb{N}.$

$u v^i x y^i z \in A$

$uv^ixy^iz = 0^i 0 1 1^i$

$S \Rightarrow \varepsilon \mid 0S1$

$s = 0011$



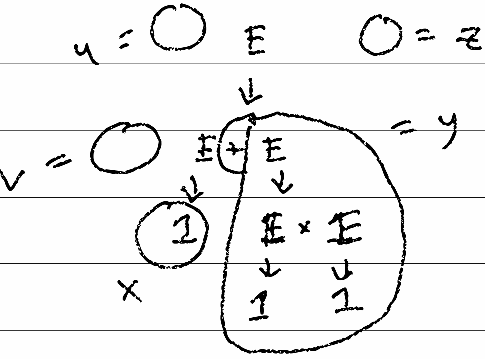
$u = \varepsilon$ $z = \varepsilon$

$v = 0$ $y = 1$

$x = 01$

15-4/ $E \rightarrow 0 \mid 1 \mid E + E \mid E \times E$

$\underline{1} + \underline{1} \times \underline{1}$



$u = \varepsilon$

$v = \varepsilon$

$x = 1$

$y = + 1 \times 1$

$z = \varepsilon$

$uv^0xy^0z = 1$

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

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

$\forall p \in N$

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

$\forall (u, v, x, y, z \in \Sigma^* \mid |vxy| \leq p$

~~&~~ $|vy| > 0)$

$\exists i \in N.$

$u v^i x y^i z \notin A$

$B = 0^n 1^n 0^n$ $\varepsilon \in B$ $010 \in B$ $001100 \in B$

$00110 \notin B$ $01100 \notin B$ $00100 \notin B$

00

```
init: see 0, push 0 , goto init
      see 1, ~~replace~~ pop 0, push , [illegible], goto mid
mid:  see 1, ~~pop 0, push~~ , pop 0, [illegible], goto mid
      see 0, pop 0, goto end
end:  see 0, pop 0 , goto end
      done → yes
```

00∤0

[I] 001100

00 [I] 01100

0000 [I] 1100

000 [M] 100

00 [M] 00

0 [E] 0

[E] → ✓

[I] 011 → 00 [I] 11 → 0 [M] 1 → [M] → ✓.

$C = 0^n 1^j 0^k \mid j + k = 2n$

16-2 / $\neg$CFPP $(0^n 1^n 0^n)$

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: $a\ \overset{0^p}{vxy}\ \overset{1^p\ 0^p}{z}$ (only left 0s)

case 2: $u\ \overset{0^p}{v}\ \overset{1^p}{x}\ y\ \overset{0^p}{z}$ (in between left 0s & 1s)

case 3: $\overset{0^p}{u}\ \overset{1^p}{vxy}\ z^{0^p}$ (only 1s)

case 4: $\underset{u}{0^p}\ \underbrace{1^p\ 0^p}_{vxy\quad z}$ (in betwen 1s & right 0s)

case 5: $\overset{0^p 1^p}{u}\ \overset{0^p}{vxy}\ z$ (only right 0s)

case 1 (3, 5): vxy = only left 0s

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

$\boxed{a+b+c = p \quad b \leq p \quad \hat{v} + \hat{y} > 0}$

$b = \hat{v} + \hat{x} + \hat{y}$ $v = 0^{\hat{v}}$ $x = 0^{\hat{x}}$ $y = 0^{\hat{y}}$

$uv^i x y^i z \in B \iff$ iff

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

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

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

✓

16-3/ case 2 (4) : vxy is L0,1

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

$\hat{u} + a = p$ $\hat{z} + b = p$

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

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

$u v^i x y^i z = 0^{\hat{u}} 0^{\hat{v} i} 0^c 1^d 1^{\hat{y} i} 1^{\hat{z}} 0^p \in B$

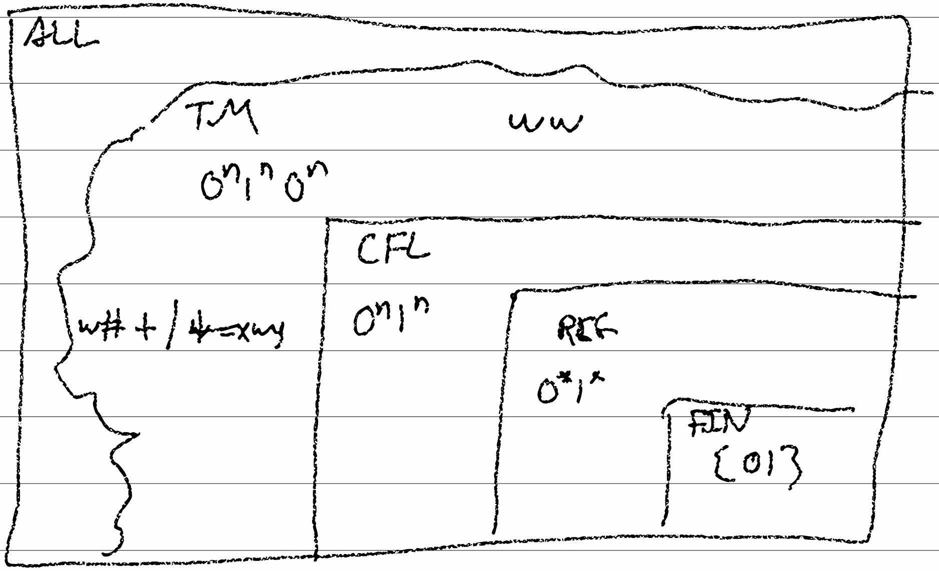
iff $\hat{u} + \hat{v} i + c = d + \hat{y} i + \hat{z} = p$

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

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

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

$i - 1 = 0$ $i = 1$



[7-1] $0^n 1^n 0^n \notin$ CFL

$w \# w \notin$ CFL

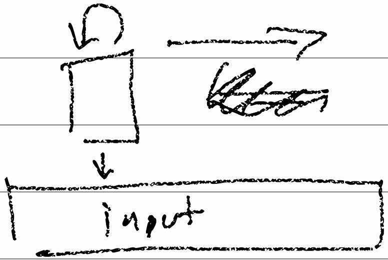
$w \# w^R \in$ CFL

Turing Machines (TM)   Alan Turing

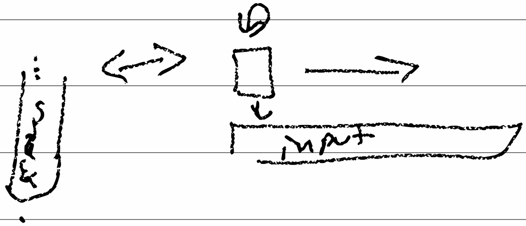
- Turing Test -

- ENIGMA

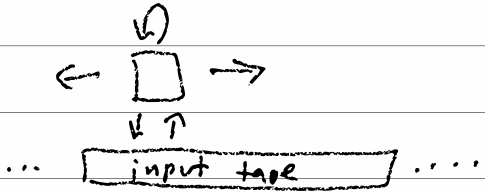
DFA



PDA



TM



17-2/ DFA: $\delta : \underset{\text{input state}}{Q} \times \underset{\text{input sym}}{\Sigma} \to \underset{\text{output state}}{Q}$

PDA: $\delta : Q \times \underset{\text{optional input}}{\Sigma_\varepsilon} \times \underset{\text{optional stack pop}}{\Gamma_\varepsilon} \to \underset{\text{nondet}}{P}(Q \times \underset{\text{optional stack push}}{\Gamma_\varepsilon})$

TM: $\delta : Q \times \underset{\text{tape read}}{\Gamma} \to Q \times \underset{\text{tape write}}{\Gamma} \times \underset{\text{direction}}{\{L, R\}}$

$(Q, \overset{\text{input}}{\Sigma}, \overset{\text{tape}}{\Gamma}, q_0, \delta, q_a, q_r)$

$q_0, q_a, q_r \in Q$ $\quad$ q_0 is the start

$\Sigma \subseteq \Gamma$ $\quad$ q_a is the ACCEPT

$\sqcup \in \Gamma$ $\quad$ q_r is the REJECT

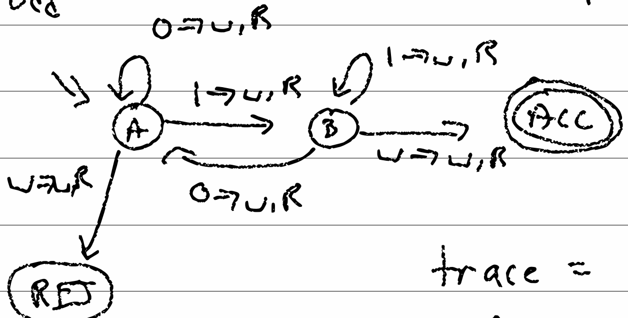
The tape is infinitely long and starts as

$\ldots \underset{\text{starts here}}{\sqcup} \text{ input } \sqcup \ldots.$

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

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

17-3/ Odd



trace = seq of config

config = $\Gamma^* \, [Q] \, \Gamma^*$

run on 01101 $\quad \ldots \sqcup\sqcup [A] 01101 \sqcup\sqcup \ldots$

$\ldots \sqcup\sqcup\sqcup [A] 1101 \sqcup \ldots$

$[B] 101 \rightarrow [B] 01 \rightarrow [A] 1 \rightarrow$

$[B] \sqcup \rightarrow [Acc] \rightarrow$ ✓

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

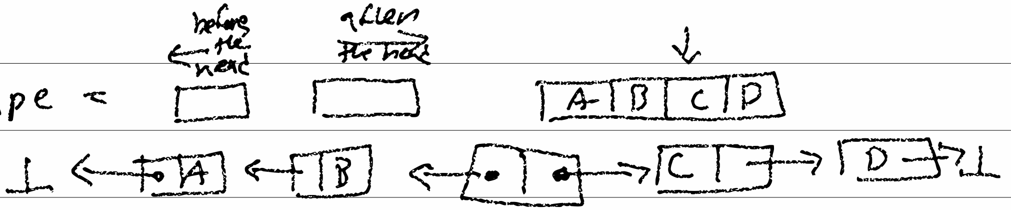
then $t = (Q \cup \{ACC, REJ\}, \Sigma, \Sigma \cup \{\sqcup\}, q_0, \delta', ACC, REJ)$

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

$\delta(q_i, \sqcup) = ACC$ if $q_i \in F$

$\qquad\qquad\quad REJ$ if $q_i \notin F$

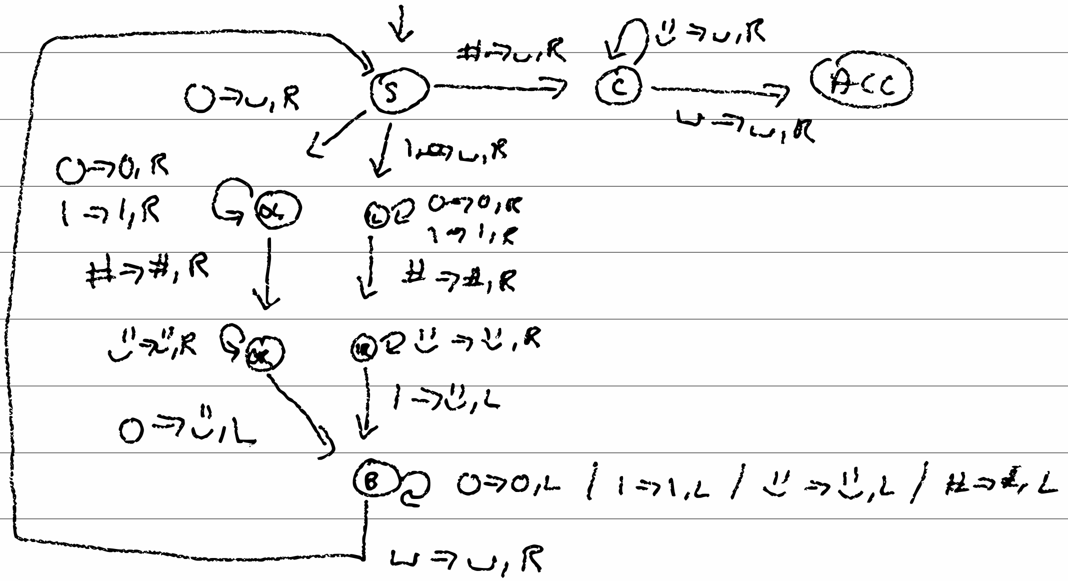
Tape =



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

17-4/ $w\#w$

$\Gamma = \{0, 1, \ddot{\sqcup}, \sqcup\}$



S
$\# \Rightarrow \sqcup, R$
C
$\ddot{\sqcup} \Rightarrow \sqcup, R$
ACC
$\sqcup \Rightarrow \sqcup, R$
$0 \Rightarrow \sqcup, R$
$1 \Rightarrow \sqcup, R$
$0 \Rightarrow 0, R$
$1 \Rightarrow 1, R$
0L
1L
$0 \Rightarrow 0, R$
$1 \Rightarrow 1, R$
$\# \Rightarrow \#, R$
$\# \Rightarrow \#, R$
$\ddot{\sqcup} \Rightarrow \ddot{\sqcup}, R$
0R
1R
$\ddot{\sqcup} \Rightarrow \ddot{\sqcup}, R$
$1 \Rightarrow \ddot{\sqcup}, L$
$0 \Rightarrow \ddot{\sqcup}, L$
B
$0 \Rightarrow 0, L$ / $1 \Rightarrow 1, L$ / $\ddot{\sqcup} \Rightarrow \ddot{\sqcup}, L$ / $\# \Rightarrow \#, L$
$\sqcup \Rightarrow \sqcup, R$

[S]010#010 $\Rightarrow$ [0L]10#010 $\Rightarrow^{2}$ 10[0L]#010 $\Rightarrow$ 10#[0R]010 ~~[illegible]~~

10[B]#$\ddot{\sqcup}$10 $\Rightarrow$ [B]$\sqcup$10#$\ddot{\sqcup}$10 $\Rightarrow$ [S]10#$\ddot{\sqcup}$10 $\Rightarrow''$ [S]0#$\ddot{\sqcup}\ddot{\sqcup}$0

[S]#$\ddot{\sqcup}\ddot{\sqcup}\ddot{\sqcup}$ $\Rightarrow$ [C]$\ddot{\sqcup}\ddot{\sqcup}\ddot{\sqcup}$ $\Rightarrow^{3}$ [C]$\sqcup$ $\Rightarrow$ ACC✓

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

17-5/ simulate : TM x input → trace

```
        simulate t s = h t ([], t.q0, s)
  h t n^{c=}(before, qi, after) = cons cn
(c,after')= case after of
              [] ⇒ (⊔, [])
              c:after' ⇒ (c, after')
     (qj, c', d) = t.δ(qi, c)
     case d of
          L ⇒ case before of
                         [] ⇒ h t ([], qj, ⊔:c':q')
                   b:before' ⇒ h t (before', qj, b:c':q')
     R ⇒ h t (c':before, qj, after')

  h t (before, qACC, after) = (list T)
                REJ                = (list F)
```

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

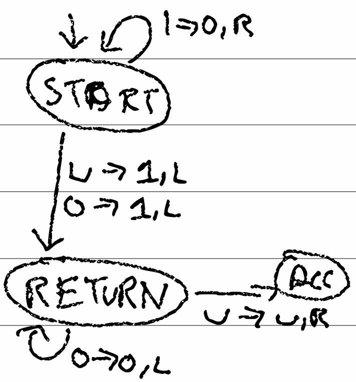
acceptance

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

Computable function

---

add1



add1(0) = 1

add1(1) = 0 1

add1(0 1) = 1 1 ← 1

add1(1 1) = 0 0 1 ← 3

add1(0 0 1) = 1 0 1 ← 2

$[S]01 \rightarrow [R]\sqcup 11 \rightarrow [ACC]11$

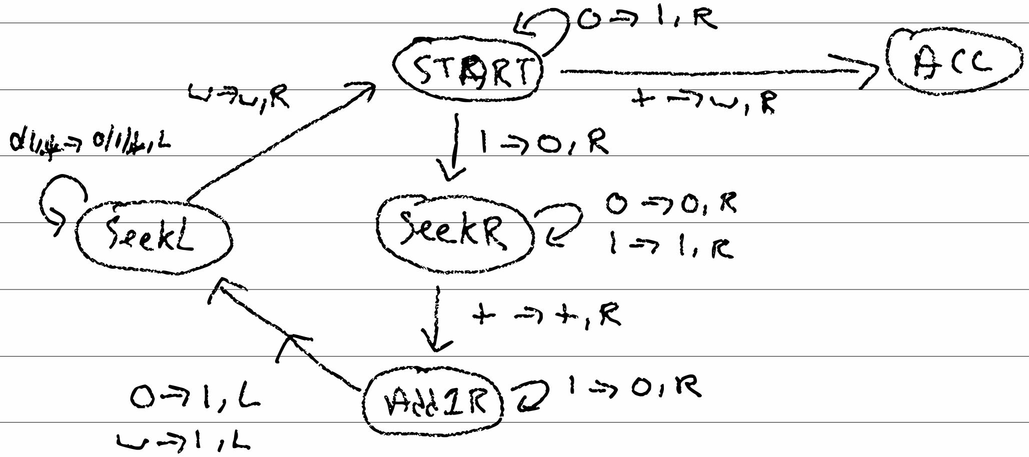
$[S]001 \rightarrow [R]\sqcup 101 \rightarrow [A]101$

$[S]11 \rightarrow 0[S]1 \rightarrow 00[S]\sqcup \rightarrow 0[R]01$

$[R]001 \rightarrow [R]\sqcup 001 \rightarrow [A]001$

18-1) X + Y   0+Y ⇒ Y

(1+X)+Y ⇒ X + (1+Y)



2+1 ⇒ 01 + 1

[S]01+1 → 1[S]1+1 → 10[SR]+1 → 10+[AR]1 →

10+0[AR]␣ → 10+[SL]01 → [ST]10+01 ⇒

0[SR]0+01 → 00+[AR]01 → 00[SL]+11 ⇒

[ST]00+11 ⇒ 1[ST]0+11 → 11[ST]+11 ⇒ 11␣[ACC]11

18-2/ DFAs defined Regular Languages

REG

CFGs =: Context-Free Langs

CFL

TM =: Turing-recognizable

Σ_1

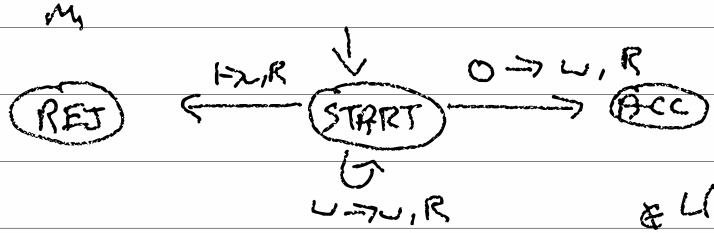
$A \in \Sigma_1$ iff $\exists t \in TM.\ L(t) = A$

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 $\forall q_i, u, v.\ [q_0]x \Rightarrow^* u[q_i]v \rightarrow \exists q_j, f, g.\ u[q_i]v \Rightarrow f[q_j]g.$ and $q_j \neq q_a$ or q_r



$0\Sigma^* \in L(M_1)$ $1\Sigma^*$ $\notin L(M_1)$ M_1 rejects

ε $\notin L(M_1)$ 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

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

$\lor\ M(x)=REJ$

$\lor\ M(x)$ diverge

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

or $M(x)=REJ$

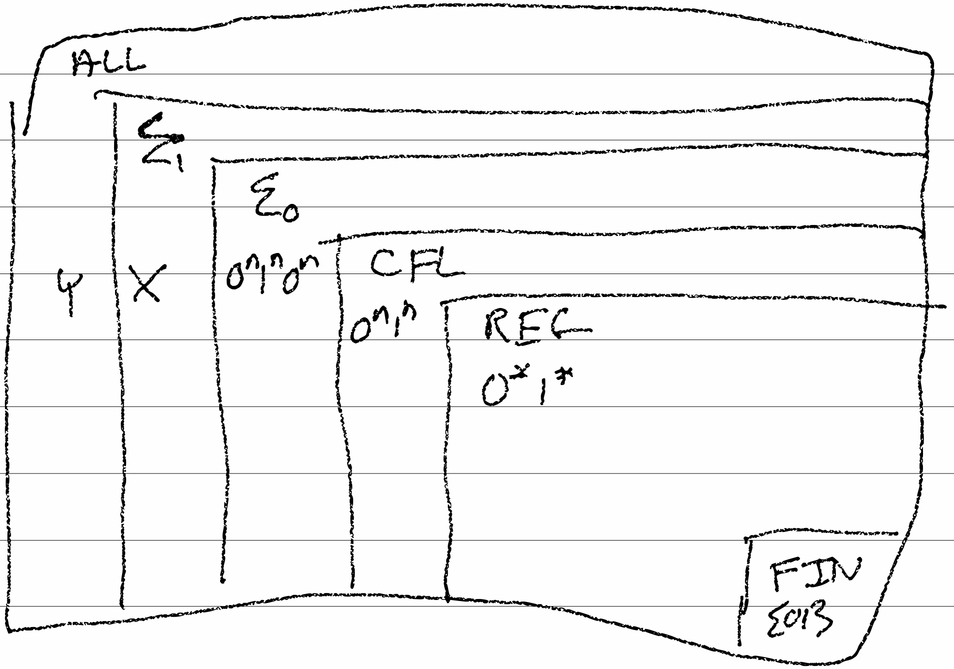
Σ_1 = Turing-recognizable $A \in \Sigma_1$ iff

Σ_0 = Turing-decidable $\exists t \in$ recog. $L(t)=A$

$A \in \Sigma_0$ iff

$\exists t \in$ deciders. $L(t)=A$

18-4



```
class DFA < State >
        Function < State , Bool > Q
        Function < Pair < State, char >, State > D
```

~~State ? Pair < Char, State >~~

```
DFA<Pair<X,Y>> Union ( DFA <X> x , DFA <Y> y )
        ...    Q =  q => x.Q(q.fst) && y.Q(q.snd)
           q0 = new Pair ( x.q0 , y.q0 )
```

19-1/ DFAs ⟷ NFAs ⟷ REX

compile : NFA → DFA

decompile : DFA → NFA

$\forall R \in REX.\ \exists N \in NFA.\ L(R) = L(N)$

$$\begin{tikzcd}
TM & & TM' \\
c_0 \arrow[d, "g"'] & & \hat{c}_0 \arrow[ll, "\alpha"'] \arrow[dd, "f"] \\
c_1 \arrow[d] & & \\
c_2 \arrow[d] & & \hat{c}_1 \arrow[d] \\
c_3 \arrow[rr, "\beta"'] & & \hat{c}_2
\end{tikzcd}$$

Galois-connection

19-2) Stay-Put TM

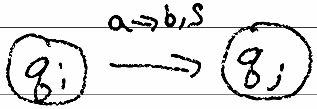
Normal : $\delta : Q \times \Gamma \Rightarrow Q \times \Gamma \times \{L, R\}$

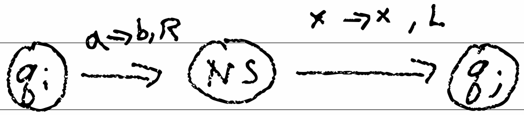
SP : $\delta : Q \times \Gamma \Rightarrow Q \times \Gamma \times \{L, R, S\}$

$$\frac{\delta(q_i, a) = (q_j, b, L)}{uc[q_i]av \Rightarrow u[q_j]cbv} \qquad \frac{\delta(q_i, a) = (q_j, b, R)}{u[q_i]acv \Rightarrow ub[q_j]cv}$$

$$\frac{\delta(q_i, a) = (q_j, b, S)}{u[q_i]av \Rightarrow u[q_j]bv}$$

$$\begin{tikzcd}
u[\hat{q}_i]av \arrow[d, "R"] & u[q_i]av \arrow[l] \arrow[d, Rightarrow, "g"] \\
ub[NS]v \arrow[d, "L"] & u[q_j]bv \\
u[\hat{q}_j]bv \arrow[ur] &
\end{tikzcd}$$





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

0 ⇒ 0, L
1 ⇒ 1, L
$\sqcup \Rightarrow \sqcup$, L

19-3/ Multi-tape TM

$$\delta : Q \times \Gamma \times \Gamma \to Q \times (\Gamma \times \{L,R,S\}) \times (\Gamma \times \{L,R,S\})$$

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

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

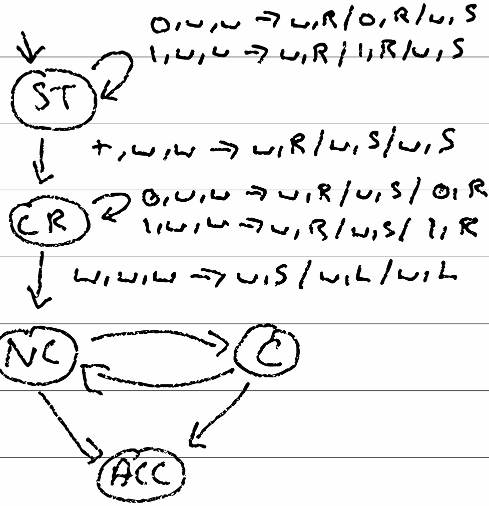
$$\begin{array}{c} uc \\ f \end{array}\left[\, q_i \,\right]\begin{array}{c} av \\ xg \end{array} \Rightarrow \begin{array}{c} u \\ f \;\; y \end{array}\left[\, q_j \,\right]\begin{array}{c} cbv \\ zg \end{array}$$

binary addition MTM

10 + 01 = 11

input : $(0 \cup 1)^* + (0 \cup 1)^*$

output : add



$$\left[ST\right]\begin{array}{l} x+y \\ ⊔ \\ ⊔ \end{array}$$

$$x\left[CR\right]\begin{array}{l} y \\ ⊔ \\ ⊔ \end{array}$$

$$\begin{array}{r} ⊔ \\ x \\ y \end{array}\left[\quad\right]\begin{array}{l} ⊔ \\ ⊔ \\ ⊔ \end{array}$$

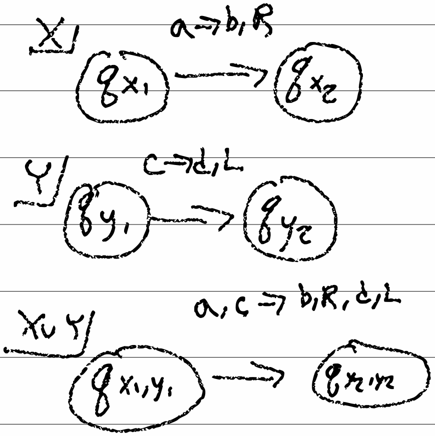
$NC \xrightarrow{N} C$: ⊔,⊔,0 → 0,⊔,⊔
⊔,0,⊔ → 0,⊔,⊔
⊔,1,0 → 1,⊔,⊔
⊔,0,1 → 1,⊔,⊔
⊔,1,⊔ → 1,⊔,⊔
⊔,⊔,1 → 1,⊔,⊔

NC → C : ⊔,1,1 → 0,⊔,⊔

NC → ACC : ⊔,⊔,⊔ → R/L/L

19-4/ Σ_1 and Σ_0 are closed under $\cup$ and $\cap$

$\text{Union}_0(X, Y) :=$ 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



$$\begin{tikzcd} TM \arrow[d, squiggly] & MTMs \arrow[l, "\alpha"'] \arrow[d] \\ {} \arrow[r] & MTM \end{tikzcd}$$

$\alpha : \overset{u}{x}\,[q_i]\,\overset{av}{by} \xrightarrow{\curvearrowleft} [q_i]\,u\dot{a}v\#x\dot{b}y : B$

$q_i \rightarrow q_i/L \Rightarrow q_i/L^{\Delta}/q_i \rightarrow \text{seek beg}/q_j \rightarrow q_j$

$u[q_i]\dot{a}v\#x\dot{b}y$
$\downarrow$
$u\dot{a}[q_i/a]v\#x\dot{b}y$
$u\dot{a}v\#x[q_i/a]\dot{b}y$

30-1 Σ_1 and Σ_0 are closed under $\circ$, $*$

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

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

```
Concate(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:

$$\begin{tikzcd}
 & u[q_i]av \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
u[q_L]av & & u[q_R]av
\end{tikzcd}$$

config : seq (det-config)

$\varepsilon \Rightarrow REJ$

$u[q_a]v ; \dots \Rightarrow ACC$

$u[q_i]av ; REST \Rightarrow REST ; u'[q_j]v'$ where $\delta(q_i, a) = (q_j, b, d)$ and $u', v' = tape(u, a, v, b, d)$

$u[q_i]av ; REST \Rightarrow REST ; u[q_L]av ; u[q_R]av$

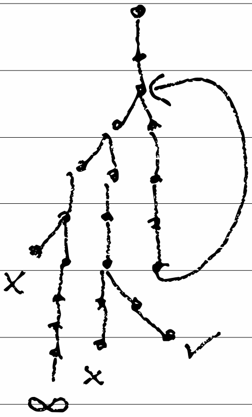
iff $\delta(q_i, a) = (q_L, q_R)$

$u[q_r]v ; REST \Rightarrow REST$

"$u[q_i]v ; x[q_j]y$" $\xrightarrow{\alpha}$ $\Gamma' = \overset{d}{\Gamma} \cup Q \cup \{;\}$ $, \omega$

$[ST]\ u q_i v ; x q_j y$

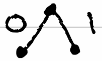
20-3/ Back-tracking non-det TM



breadth-first-search

rop = (0 ∪ 1)*

consume 1 length of rope



look at lexico-order of rope

ε, 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

20-4/

$$\begin{tikzcd}[column sep=small]
f(x) \arrow[dr, no head] & & g(y) \arrow[dl, no head] \\
& g(f(x)) &
\end{tikzcd}$$

PL for TM:

e = By-hand

| $e \cup e$ | $e \cap e$

| $e \circ e$ | e^*

| $e(e)$

---

enumerator = ...

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

$\delta : \underset{\overset{\nearrow}{Q - \{q_a, q_r\}}}{Q} \times \Gamma \to Q \times \Gamma \times \{L, R\}$

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

$\delta : \underset{\overset{\nearrow}{\in Q_p}}{Q} \times \Gamma \to Q \times \Gamma \times \{L, R\}$

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

enum ⇒ TM :

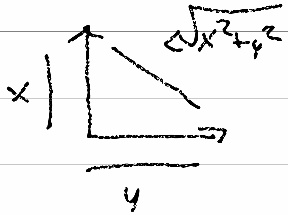
TM ⇒ enum :

21-1)

Church-Turing Thesis

"Algorithm" = $\left\{ \begin{array}{l} \text{Turing Machine} \\ \text{Lambda Calculus} \end{array} \right\}$

"Add" +



$x + 0 = x$

$x + y = y + x$

$A \notin \Sigma_1 \quad \wedge \quad CTT$

$\Rightarrow A$ is unsolvable

$A \notin \Sigma_0 \quad \wedge \quad CTT$

$\Rightarrow A$ is undecidable

$A \notin P \quad (\exists d \in DFA.\ |d.Q| \leq |inpt|)$

$\Rightarrow A$ is un/infeasible / intractable

21-2) 1900 - World Congress of Mathematics

David Hilbert ~~[illegible]~~ 15 problems
proposed

for [1900, 2000]

Polynomial Root Problem

Given a polynomial, what integers for the variable, equal to 0?

$$ax^2 + bx + c = 0 \quad \text{iff} \quad x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

A polynomial over n variables ($x_0 \dots x_n$)

↗ is defined by co-efficients $a_{0 \dots n}^{0 \dots m}$

of degree m

$$4x^3z^2 + 5x^4zy + 1 + 2y^2z^2$$

$\{x, y, z\}$, 4, $a_{3,0,2} = 4$ $a_{4,1,1} = 5$

$a_{0,0,0} = 1$ $a_{0,2,2} = 2$

Imagine poly of 1 var, but any degree

If $\exists x.\ p(x) = 0$, then $x \in [-L, +L]$

$L = k \cdot \frac{c_{max}}{c_i}$ where k = degree

c_i = coefficient of last degree

c_{max} = large coefficient

21-2) $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$2x^2 - 4x - 2 = 0$ $+4 \pm \sqrt{16 + 16}$

$\overset{\nearrow}{a}$ $\overset{\nearrow}{b}$ $\overset{\nearrow}{c}$

~~$-7 + 21x - 7x^2 - 1x^3 + 2x^4$~~

$4x^3 - 2x^2 + x - 7$ $k = 3$ $3 \cdot \frac{7}{4} = 5\frac{1}{4}$

$\pm k \cdot \frac{c_{max}}{c_1}$ $c_1 = 4$

$c_{max} = 7$

$[-5, 5]$ Matijasevič's Theorem

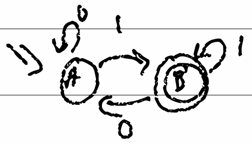
w is accepted by D

21-3/ $A_{DFA} = \{ \langle D, w \rangle \mid$ D is an "encoding" of a DFA and $w \in \Sigma^* \}$

DFA $d = (Q, \Sigma, q_0, \delta, F)$ $\Sigma = \{0, 1\}$

$1^n 0$ where $n = |Q|$

x in lg n bits where x is idx of q_0



n-bits where 0 mean $q_i \notin F$

1 mean $q_i \in F$

$|Q| = 110$ 11000

$\delta =$

	Q			A	B
Σ	lg n	lg n	0	A	A
	lg n	lg n	1	B	B

$q_0 = 0$ = 0011

$F = 01$

$\delta = 0011$

1100010011 01101 $\in A_{DFA}$

$A_{DFA} \in \Sigma_0$ (decidable)

A.

22-1/ $A_{TM} = \{ <M, w> \mid$ where M is TM-encoding
w is in Σ^*
and $w \in L(M) \}$

Turing Omnibus

A machine that solves $A_{TM} = \mathcal{U}$

$\mathcal{U} \in \Sigma_1$ "The Halting Problem"

Assume that $L(H) = A_{TM}$ and $H \in \Sigma_0$

$H(<M, w>) =$ accept if M accepts w
reject if M does not accept w
~~NO LOOP~~

D = " On input $<M>$, where M is a TM,
1. Run H on $<M, <M>>$
2. Output opposite of H. "

$D(<M>) =$ accept is M does not accept $<M>$
reject if M accepts $<M>$

Run D on $<D> =$ accept if D does not acc $<D>$
reject if D accepts $<D>$

what answer is returned? → LOOP

$\Rightarrow H \notin \Sigma_0$

22-2/ $A_{TM} \in \Sigma_1$ but $A_{TM} \notin \Sigma_0$

$\Rightarrow \Sigma_1 \neq \Sigma_0$

solvable ≠ decidable

recognizers ≠ deciders

$A \in \Sigma_0 \Rightarrow A \in \Sigma_1$ and $\bar{A} \in \Sigma_1$

$\exists m$ ↗ m ↗ $\bar{A}(w)$ = run $A(w)$, output opposite

$A \in \Sigma_0 \Rightarrow \bar{A} \in \Sigma_0$ $A \in \Sigma_0 \Rightarrow A \in \Sigma_1$

$A \in \Sigma_1$ and $\bar{A} \in \Sigma_1 \Rightarrow A \in \Sigma_0$

$\exists x$ $\exists y$ $\Rightarrow$ $Z(w)$ = run X on w

interleave (run Y on w

if X acc, we acc

if Y acc, we reject

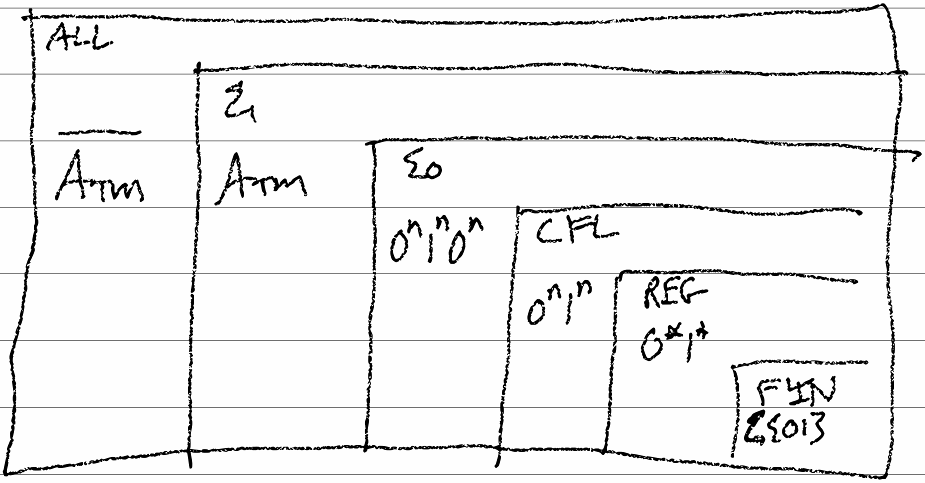
$$A_{TM} \notin \Sigma_0 = \neg(A_{TM} \in \Sigma_0)$$

$$= \neg(A_{TM} \in \Sigma_1 \wedge \overline{A_{TM}} \in \Sigma_1)$$

$$= \neg A_{TM} \in \Sigma_1 \vee \neg \overline{A_{TM}} \in \Sigma_1$$

$$= \underbrace{A_{TM} \notin \Sigma_1}_{\text{FALSE}} \vee \underbrace{\overline{A_{TM}} \notin \Sigma_1}_{\text{TRUE}}$$

22-3

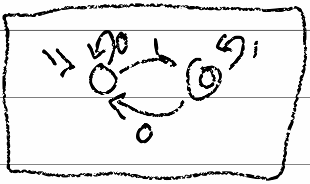

ALL
$\overline{A_{TM}}$
Σ_1
A_{TM}
Σ_0
$0^n1^n0^n$
CFL
0^n1^n
REG
0^*1^*
FIN
$\varepsilon\{01\}$

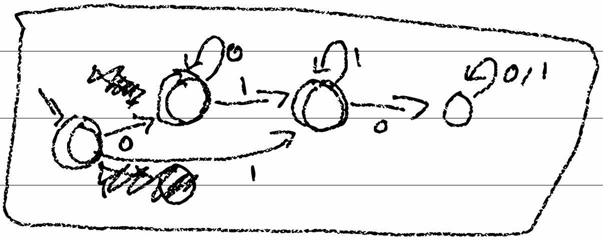
22-4) When are two sets the same size?

$|\{a, b, c\}| = 3$

$3 = 3 \Rightarrow$ √

$|\{C, B, S\}| = 3$





A set X is the same size as a set Y iff $\exists f : X \to Y$, where f is one-to-one and onto

one-to-one : $\forall a, b.\ f(a) = f(b) \Longrightarrow a = b$

onto : $\forall b.\ \exists a.\ f(a) = b$

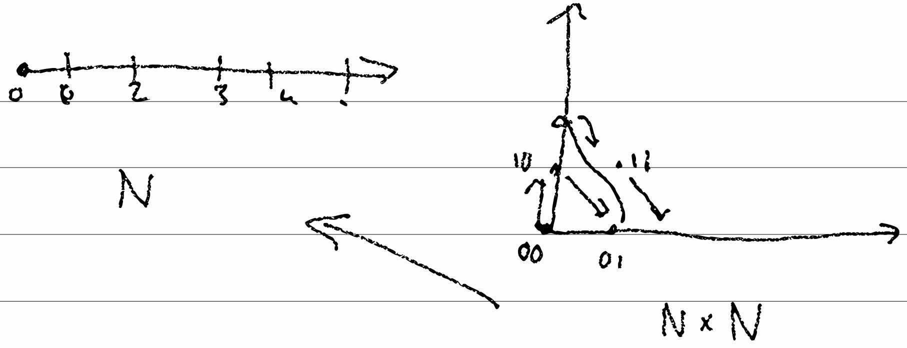
The natural numbers are the same size as the even numbers

$N = 0, 1, 2, 3, 4, 5, 6, \ldots$

$f(x) = 2x$

$EVEN = 0, 2, 4, 6, 8, 10, 12, \ldots$

22-3)



$f(x, y) = 0.5(x+y)(x+y+1) + y$

Cantor-pairing function

$f(0,0) = 0$ $f(0,1) = 2$ $f(1,0) = 1$

$f(1,1) = 4$ $f(0,2) = 5$ $f(2,0) = 3$

$N \approxeq N \times N = N^2$

$N \approxeq (N \times N) \times N$

$N \approxeq N^k \quad \forall k$ $TM = (Q, \Sigma, \Gamma, q_0, \delta, q_a, q_r)$

$|TM| < N^{10}$

$N \approxeq \Sigma^*$ (lexi) lexi : $N \Rightarrow \Sigma^*$

$A \approxeq N$:= "countable"

22-4/ IBS = infinite binary sequence

$\in \mathbb{N} \to \{0,1\}$

0000000 .... $\leftarrow$ IBS

fun (i) { return 0; }

$f = g$ iff

$\forall x.\ f(x) = g(x)$

01010101 ... $\leftarrow$ IBS

fun (i) { i % 2)

111 1 1 00... $\leftarrow$ IBS = fun(i) { return i<5; }

$\mathbb{N} \not\approx$ IBS =

$\neg (\exists f \in \mathbb{N} \to \text{IBS}.$ f is oto $\wedge$ f is onto$)$ =

$\forall f \in \mathbb{N} \to \text{IBS}.$ f isn't oto $\vee$ f isn't onto $\Leftarrow$

$\forall f \in \mathbb{N} \to \text{IBS}.$ f isn't onto =

$\neg (\forall b \in \text{IBS}.\ \exists a \in \mathbb{N}.\ f(a) = b)$

$\forall f \in \mathbb{N} \to \text{IBS}.\ \exists b \in \text{IBS}.\ \forall a \in \mathbb{N}.\ f(a) \neq b$ =

$\exists i \in \mathbb{N}.\ f(a)(i) \neq b(i)$

given : f pick : b = fun (x) {

return $\neg$ f(x)(x); }

given : a pick : i = a,

$f(a)(i) = f(a)(a) \neq b(a)$ =

$\neg f(a)(a)$

Cantor's Diagonalization Theorem

T $\neq$ F

F $\neq$ T

22-5/ IBS $\approx_c$ $\mathbb{R}$ (real numbers)

$$\begin{array}{ccc} \mathbb{N} & \not\approx_c & IBS \\ \approx_c & & \approx_c \\ TM & & ALL \\ \approx_c & \Rightarrow & TM \not\approx_c ALL \\ \Sigma_1 & \Rightarrow & \Sigma_1 \not\approx_c ALL \end{array}$$

$ALL = P(\underbrace{\Sigma^*}_{\text{infinite set } (\mathbb{N})})$

$\parallel$

$\mathbb{N}$-bit number

$=$

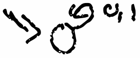
IBS

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

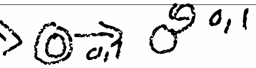
00	$\emptyset$
10	$\{0\}$
01	$\{1\}$
$\underbrace{1}_{0\in}\underbrace{1}_{1\in}$	$\{0,1\}$

$= P(A) =$ $|A|$-bit number

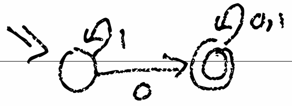
$0\ldots \in IBS$ $\quad \emptyset =$



$10\ldots \in IBS$ $\quad \{\varepsilon\} =$



$01010\ldots \in IBS$ $\quad \{$ strings w/ a 0 $\} =$



23-1/ $X \notin \Sigma_0$ $(\in \Sigma_1)$ $\Rightarrow$ $\overline{X} \notin \Sigma_1$

Mapping Reducibility

A is m.r. to b $(A \leq_m B)$ if

$\exists f \in$ computable function (Σ_0) where

$\forall w.$ $w \in A$ iff $f(w) \in B$

If $A \leq_m B$ and $B \in \Sigma_0$, then $A \in \Sigma_0$

If $A \leq_m B$ and $A \notin \Sigma_0$, then $B \notin \Sigma_0$

If $A_{TM} \leq_m B$, then $B \notin \Sigma_0$

so,

$\exists f \in$ c.f. $\forall w.$ $w \in A_{TM}$ iff $f(w) \in B$

$\Rightarrow$ $B \notin \Sigma_0$

$E_{TM} = \{ \langle M \rangle \mid M$ is a TM and $L(M) = \emptyset \}$

$A_{TM}(\langle M, w \rangle) = E_{TM}(M')$

$M'(x) =$ if $x == w$, then simulate M on w

o.w, reject

23-2) $REG_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \in REG \}$

$A_{TM}(\langle M, w \rangle) = REG_{TM}(M')$

where $M'(x) =$ if $x \in 0^n 1^n$, then accept

o.w. ~~if~~ if $M(w)$ accepts, accept

o.w. reject

$\swarrow \qquad \searrow$

$0^n 1^n$ if M d.n.a w $\qquad$ Σ^* if M d.a. w

$EQ_{TM} = \{ \langle M_x, M_y \rangle \mid M_{x,y} \text{ is a TM and } L(M_x) = L(M_y) \}$

$E_{TM}(\langle M \rangle) = EQ_{TM}(\langle M, M_\emptyset \rangle)$

Where $M_\emptyset(x) =$ reject

Rice's Theorem: All non-trivial semantic properties of TMs are undecidable.

semantic = behavior $\qquad$ syntactic = form

non-trivial$(P) = \exists A \in ALL.\ P(A)$

$\wedge\ \exists B \in ALL.\ \neg P(B)$

23-3/ LBA - linear bounded automata

LBA is a Turing Machine but the tape is finite

TM rule: $u[q_i]v \Rightarrow \sqcup u[q_i]v\sqcup$

TM config_0: $\sqcup^*[q_0]w\sqcup^*$

LBA — no $\sqcup$ rule

config_0: $\sqcup[q_0]w\sqcup$

Every TM we wrote has an LBA

LBA	TM		
✓	✓	Accept	$\Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow$ ✓
✓	✓	Reject	$\Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow$ X
	✓	Diverge	$\Rightarrow\ \Rightarrow\ \ldots\ \Rightarrow\ \Rightarrow\ \ldots\ \Rightarrow\ \Rightarrow$
✓	✓	Loop	$\Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow\ \Rightarrow$

23-4| $A_{LBA} = \{ \langle M, w \rangle \mid M$ is an LBA and $w \in L(M) \}$

$A_{LBA} \in \Sigma_0$

If ~~q~~ M has q states ($|Q|$) and
g symbols ($|\Gamma|$) then
there are only $q \times n \times g^n$ configs for
input of length n

tape $= g^n$ head pos $= n$
state $= q$

$$2^{16+64+16*30+2^{16\times 30}}$$

E_{LBA} is undecidable ($\notin \Sigma_0$)

$A_{TM}(\langle M, w \rangle) = E_{LBA}(M')$

$M'(x)$ = accepts if $x = c_0 c_1 \dots c_n$

where $c_0 = [M \cdot q_0] w$

and $c_n = u [M \cdot q_a] v$ for some u, v

and $c_i \Rightarrow c_{i+1}$ by TM rules

23-5

```
interface RegEx

class REmpty    : Regex      ()
      REpsilon  : Regex      ()
      RChar     : Regex     (Char c)
      RUnion    : Regex     (Reegex x, y)
      RConcat   : Regex     (Rege x, y)
      RStar     : Regex     (Reegex x)
```