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 $\underbrace{\text{alphabet}}_{\text{some finite set } \Sigma}$

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 ~~a~~ ~~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 $P \in \{ \varepsilon, P, PP \}$

$x \in X \cup Y$ 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 $x \in U$ – 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

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

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

1-31 lexi : num $\rightarrow$ string of Σ ($\Sigma = \{0,1\}$ $|\Sigma| = 2$)

lexi 0 = ε lexi 1 = 0 lexi 2 = 1

lexi 3 = 00

lexi n = sz = size of Σ

if $n < sz^0$ then ret ε of len 1

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

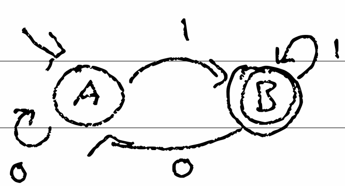
$(n - sz^0) - sz^1 < sz^2$ convert of len 2

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

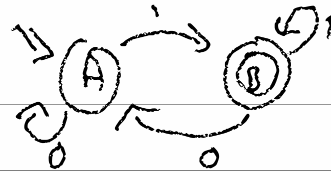
$\varepsilon \in \Sigma^*$ PPCBSPPPP $\in \Sigma^*$

010111 $\in \Sigma^*$

Deterministic Finite Automata (DFA)



2-1) DFA



◎ means Yes

○ means No

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

transition function (edges)

$Q \times \Sigma \to Q$

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

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

Σ: $\{0,1\}$

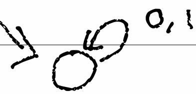
q_0: start state $= A$

F: accepting states $\{B\}$

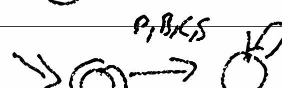
	A	B
0	A	A
1	B	B

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

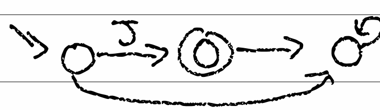
No string DFA :



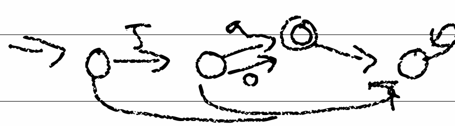
only empty string :



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.member (d.q_0)

accepts d (c :: s)

: DFA : Q : Σ^*

accepts d s = helper d d.q_0 s

helper d q_i ε = $q_i \in$ d.F

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

q_j = d.δ(q_i, c)

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

✓ trace

2-3) 0110 ⇒ Even, Odd, Odd, Even

Transducers are DFAs where trace matters

Moore machines

L(d) = the language of DFA d

= { s | accepts d s = true }

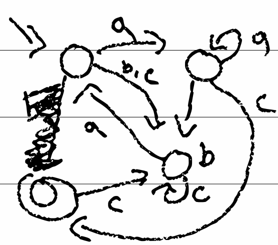
may be infinite

Given a DFA, return a string that would be accepted

example : DFA ⇒ Σ^* or false

s.t. if example d returns s then

accepts? d s = true

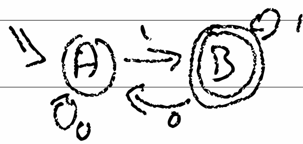
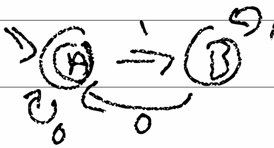


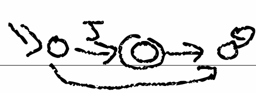
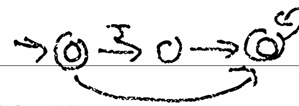
'b'

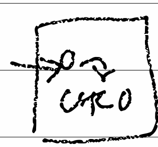
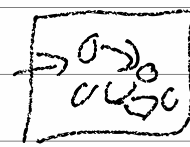
2-4/ Suppose that d is a DFA, construct d' where $L(d') = \overline{L(d)}$ (i.e. d' says yes when d says no & vice versa)

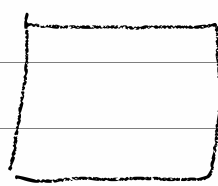
negate : DFA → DFA

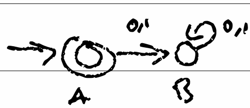
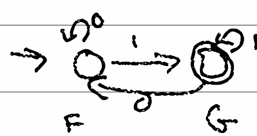
negate (Odds) = Evens

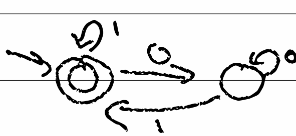
in:  out: 

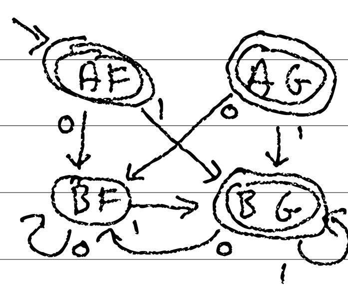
in:  out: 

X:  Y: 

X ∪ Y: 

X:  Y: 

X ∪ Y: 

Z: 

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

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

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

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

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

and to make intersect

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

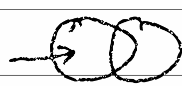
$X \subseteq Y$ (subset) iff
$\forall q \in X.\ q \in Y.$

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

$\text{Subset?} : DFA \times DFA \to bool$

Subset? $(\varnothing)$ X = Yes
(epsilon) (Evens) = Yes
(epsilon) (Odd) = No

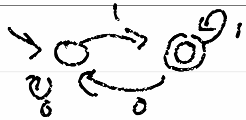
empty



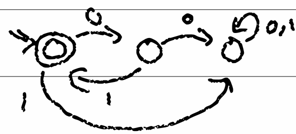
X Y

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

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

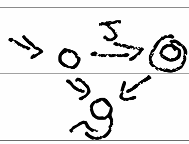
$X =$ 

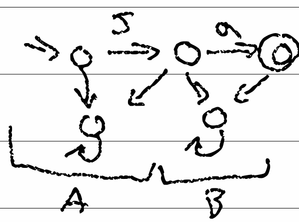
$x \in X = 0111$

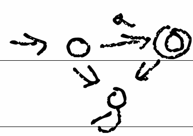
$Y =$ 

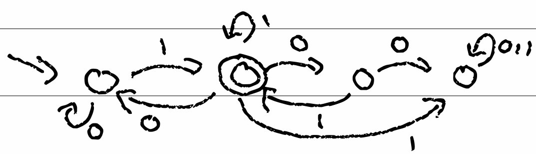
$y \in Y = 010101$

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

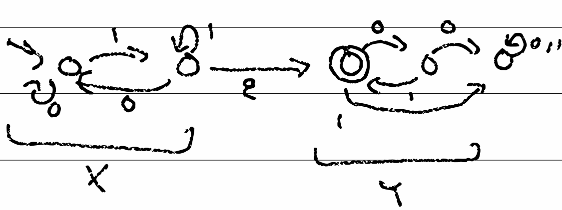
$A =$ 

$A \circ B =$ 

$B =$ 



// illegal, not a DFA!



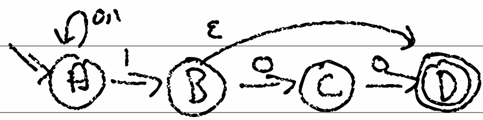
(skip from F to G at the right time)

3-2/ NFA – non-deterministic finite automata

DFA – $(Q, \Sigma, q_0, \delta : Q \times \Sigma \to Q, F \subseteq Q)$

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

sigma epsilon



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

$\forall t.$ oracle $n\ t$ = ff

(or "↯ t")

trace = a sequence of $Q \times \Sigma_{or}\ \varepsilon$

(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, ...)

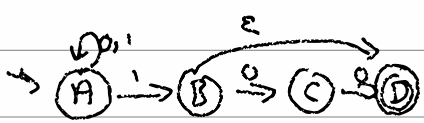
set

all : NFA $\times$ Σ^* $\rightarrow$ tt

all N s = helper N N.q_0 s

helper N q_i s =
branch q_i (case s where
[] $\rightarrow$ if $q_i \in$ N.F then
{accept}
o.w
{reject}

c :: s' $\rightarrow$
{ tt | tt = helper N q_j s'
where $q_j \in N.\delta(q_i, c)$ })
$\cup$ { tt | tt = helper N q_j s
where $q_j \in N.\delta(q_i, \varepsilon)$ and $q_i \neq q_j$ }



0100

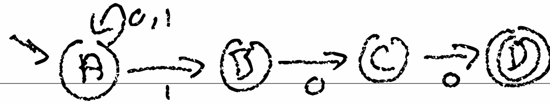
0 1 A $\xrightarrow{0}$ A $\xrightarrow{0}$ A $\rightarrow$ Reject

A $\rightarrow$ A

$\xrightarrow{1}$ B $\xrightarrow{0}$ C $\xrightarrow{0}$ D $\rightarrow$ ACCEPT
= (branch D (accept))

$\xrightarrow{\varepsilon}$ D
= (branch D $\emptyset$)

3-4/



"ends in 100"

011100

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

$\quad \xrightarrow{1} B \quad \xrightarrow{1} B \quad \xrightarrow{1} B \xrightarrow{0} C \xrightarrow{0} D \rightarrow$ ACC

backtrack : NFA × String Σ^* → Bool

backtrack N s = helper N $N.q_0$ s

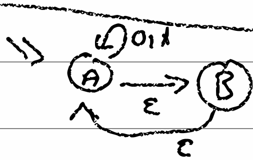
helper N q_i s =

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



maybe DS = tree.

unfold : A → DS(B)

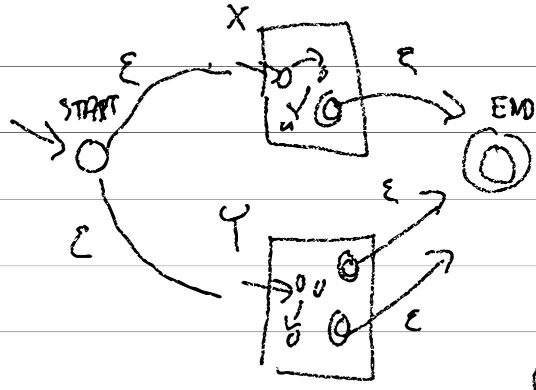
fold : DS(B) → C

Haskell

there always exist

combined : A → C

4-1



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

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

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

$F_z = \{END\}$

$q_{0z} = START$

$Q_z = \{START, END\}$

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

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

$\delta_z(q_i, c) =$

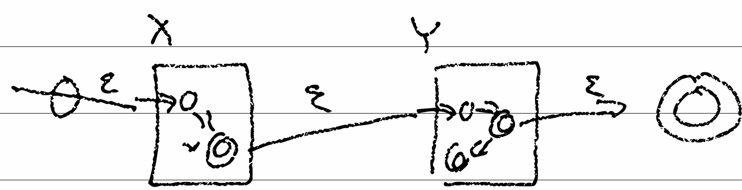
$(START, \varepsilon) = \{(q_{0x}, 0) \; (q_{0y}, 1)\}$

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

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

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

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

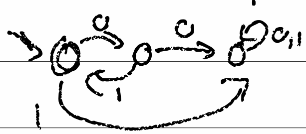


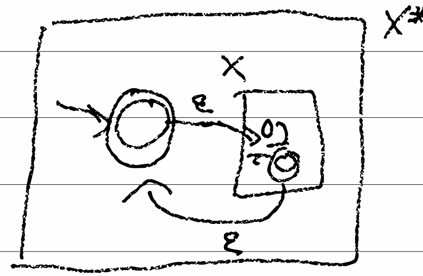
4-2/ Kleene-star X^*

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

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

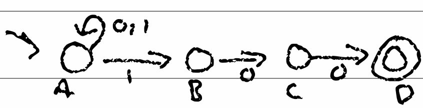
$(\{0\}\circ\{1\})^*$ = any number of 01 sequences



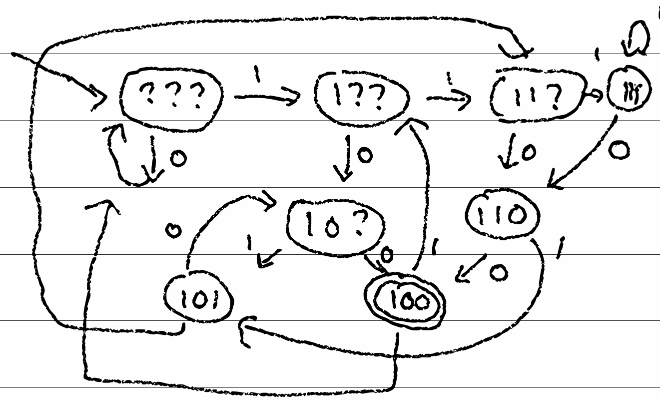


DFAs: $\cup$, $\cap$, $\overline{}$ DFA → NFA

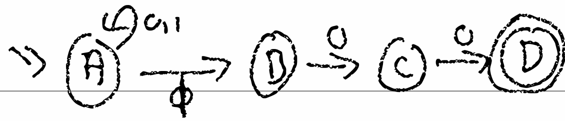
NFAs: $\cup$, $\circ$, $*$ want: NFA → DFA

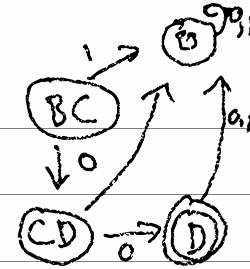


01100



4-3/





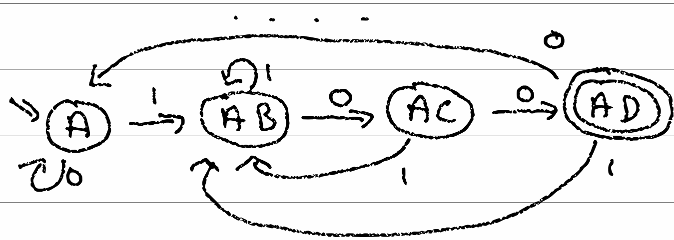
01100

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

B ~~B~~ B C D ← ✓

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

$\emptyset$ $\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$



01100 ✓

011001 X

1111 X

111100 ✓

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

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

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

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

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

4-4)

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

= accepts (NFA→DFA N) x

= pick a random number

(lexi n)

write the NFA manually as a DFA...

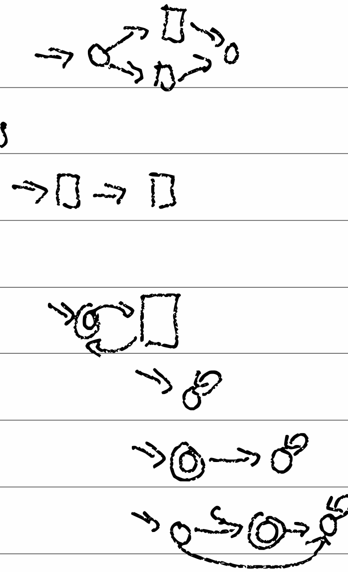
use the dFA equality checker to see that

dfa-equal? manual (N→D N) = #true

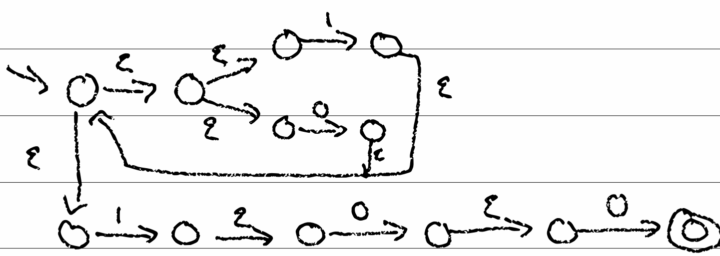
5-1/ regular expression := $x \cup y$ / $x \cap y$ / $x \circ y$ / $\overline{x}$ / x^* / $\emptyset$ / ε / $c \in \Sigma$

x, y r.e

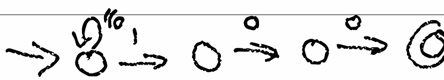
Then regular operations



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



=



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

5-2/ interface RegEx;

```
class Empty implements RegEx;   ()
      Epsilon          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'))))
```

=

$(1 \cup 0)^* 100$

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

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

5-3 generate : $RE \to \Sigma^*$ 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∘t)
= h x (lambda s: h y (lambda t: pr s∘t))

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

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

numbers

5-4 $\quad X \cdot 0 = 0 \qquad x + y = y + x \qquad x \cdot 1 = x$

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 $x \leq z$

DFAs $\longleftrightarrow$ NFAs

REX

1) $\bigcirc \xrightarrow{1} \circ \xrightarrow{0} \circ \xrightarrow{0} \circledcirc \implies (1 \cup 0)^* 100$

G-1

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

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

$\to$ GNFA (2)

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

NFA: $o \xrightarrow{c} o$

$o \xrightarrow{\varepsilon} o$

GNFA: $o \xrightarrow{r} o$

GNFA (generalized NFA)

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

one state, not a set

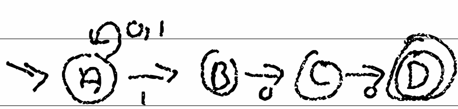
$\Delta : (Q \times Q) \to \text{Rex}$ $\delta : Q \times \Sigma \to P(Q)$

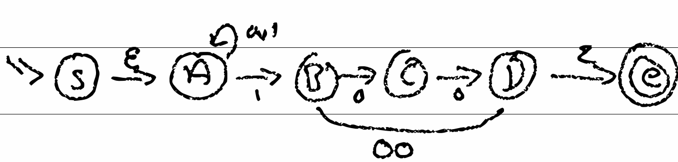
$(Q - q_f)$ $(Q - q_a)$

You can't leave q_f You can't go back to q_a

$(s) \xrightarrow{r} ((e))$

$\Delta(q_i, q_j) = r$ ~~if~~ if $x \in r$ then $q_i \xrightarrow{x} q_j$ in the NFA

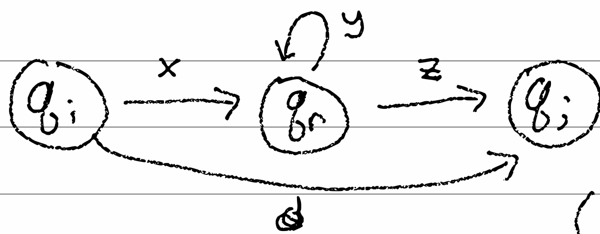




G-2/ Rip : GNFA(n+1) $\Rightarrow$ GNFA(n)

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

q_r is gone

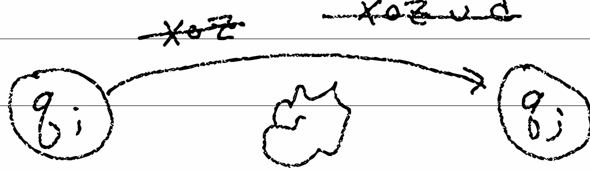


$(x \circ y^* \circ z) \cup d$

$\Rightarrow$

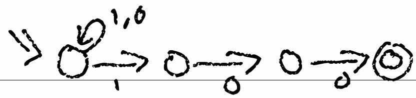
...

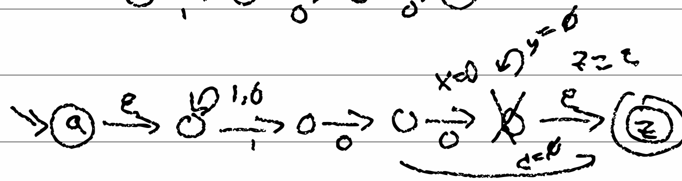
~~$x \circ z \cup d$~~ $\quad x \circ y^* \circ z \cup d$



G-3

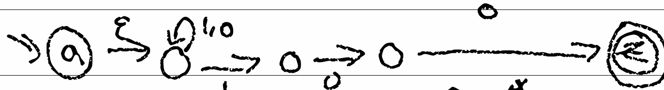
IN





RIP

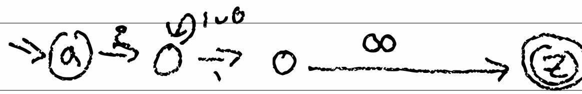
$xy^*z \cup d$



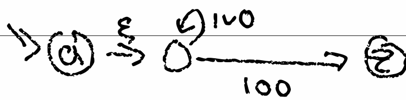
$0 \cdot \emptyset^* \circ \varepsilon \cup \emptyset$

$= 0$

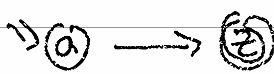
RIP



RIP



RIP

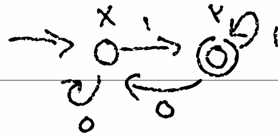


$\varepsilon \circ (1 \cup 0)^* \circ 100 \cup \emptyset$

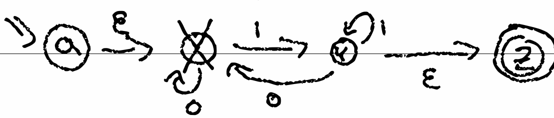
OUT

$= (1 \cup 0)^* \circ 100$ ←

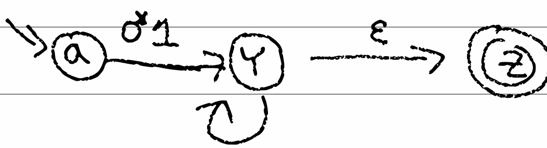
6-4/

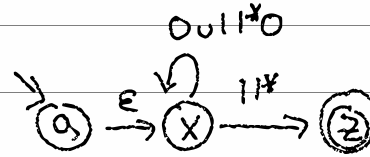


$\Downarrow$ IN



$\Downarrow$ RIP





$\Downarrow$ RIP

$\Downarrow$ RIP

$0^*1 \circ (1 \cup 00^*1)^* = (0 \cup 11^*0)^* 11^*$

$=$

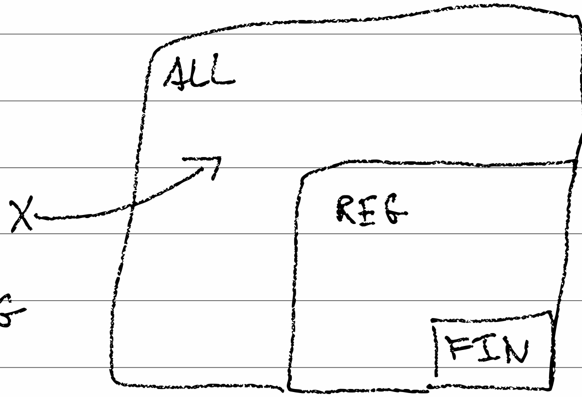
$(0 \cup 1)^* 1$

DFA $\leftrightarrows$ REX

$\forall d.$ let r = dfa→rex d

accepts? d (generate r) = true

6-5)



REG = language described by DFA, NFA, or RE

FIN = finite languages

$X \notin REG$

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

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

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

{all of Jay's lectures}

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

$\ldots \}$

REG = ALL?

2-1/ $\exists X \in ALL.\ X \notin REG$

language (some problem)   all possible languages   languages defined by DFAs

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

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

alphabet

finite set of states

start somewhere

many stop

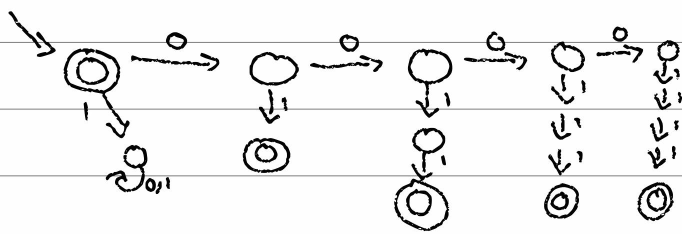
$Q \times \Sigma \to Q$

EQ

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

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

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



$= EQ^4$

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

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

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

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

$\wedge\ 0^n 1^n \in EQ^m$ 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')
ungetc(c); count++;
while ( c = getc(); c == '1')
    count--;;
return count==0;
```

$\Longrightarrow$

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

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

~~EQ...~~

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

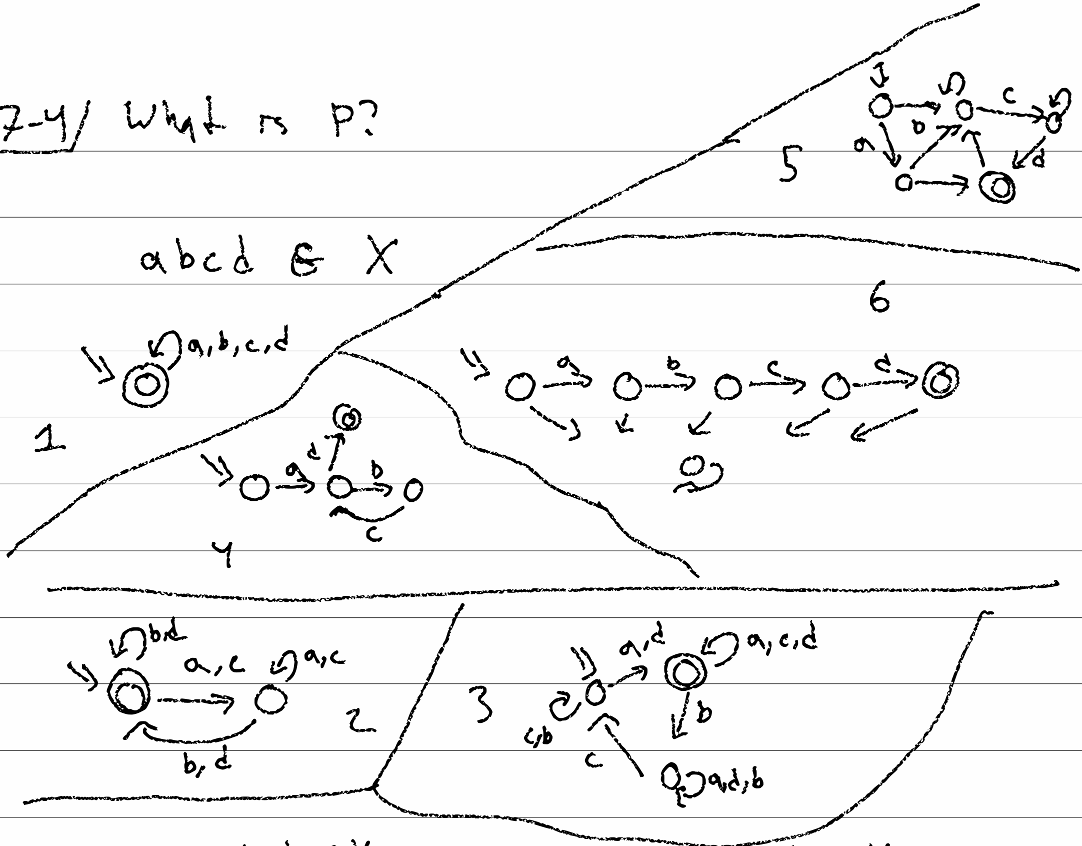
$=$

X

step 1: what is X? ✓

7-4/ What is P?

abcd ∈ X



1: aaaabcd ∈ X

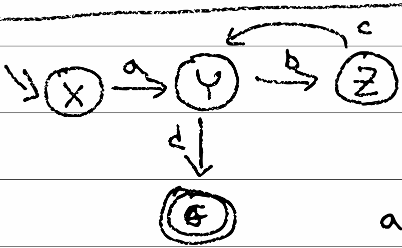
3: abcabcabcd ∈ X

2: ababababcd ∈ X

4: abcbcbcd ∈ X

5: x

6: x



abcd string (i=1)

X Y Z Y G trace

abcbcbcd string (i=3)

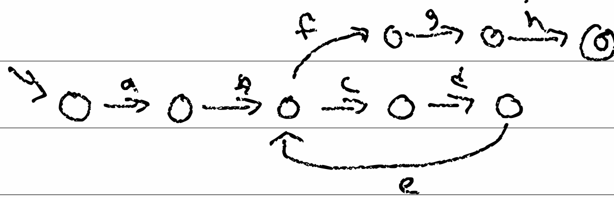
X YZYZYZYG trace

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

ad string (i=0)

X Y G

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: $s = abcd$ $\quad$ s visited Y twice $\quad$ $|Q| = 4$

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

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

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

That means for all i

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

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

7-6/ Regular Pumping Property (RPP)

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

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) $\neg RPP(EQ)$

$= \forall p \in N.$

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

$\forall (x, y, z \in \Sigma^* \mid$

$|y| > 0$

$|xy| < p)$

$\exists i \in N$

$xy^iz \notin A$

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

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

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

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

$\forall$ = you don't choose it

$\exists$ = you DO choose it

$\neg RPP(EQ)$

given p.

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

$s = 0^p 1^p$

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

$s = xyz$ $0^p1^p = xyz$ $b > 0$ $a+b+c = p$

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

choose i $i = 0$

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

iff $a + bi + c \neq p$

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

$bi \neq b$

$i \neq 1$

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

then

$xy^*z \in A$

regular expression

$xy^*z \subseteq A$

ALL $\neq$ REG because EQ $\in$ ALL

EQ $\notin$ REG

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

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

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

$000 \circ 0011 \in MEQ^3$

9-1/ EQ ⇒ $0^n 1^n$ for some n

① machine DFA ⇄ ② "programming language" Regular expressions] REG

⇓

machine PDA ← "PL" ① CFG] CFL

CFG : context-free grammar   substitution

$S \to \varepsilon$ RHS → RHS — rule

$S \to 0S1$ LHS is always a   production / derivation

$\underbrace{S \to \varepsilon \quad S \to 0S1}_{\text{a CFG for } 0^n1^n}$ 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"

$E \to N \mid E\ O\ E$

$N \to$ '0' | '1' | '2'

$O \to$ '+' | '-' | '×' | '÷'

q2/ membership $x \in A$

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

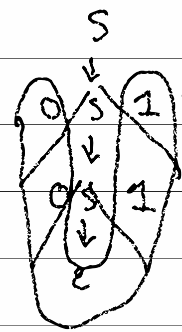
generation A produces $s_1, s_2, s_3, \ldots$

$S \to \varepsilon \mid 0S1 \quad = g$

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

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

=



= parse tree

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

pt2str (V, seg) = seg2str seg

seg2str [] = ε

seg2str $c \in \Sigma$ :: seg = c :: seg2str seg

seg2str PT::seg = pt2str PT ++ seg2str seg

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

9-3/ ~~cfggen~~

some set — alphabet — $\in V$

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

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

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

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

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

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

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

```
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 )
```

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

all_n_deep 0^n1^n 2 = ε , 01 , 0011

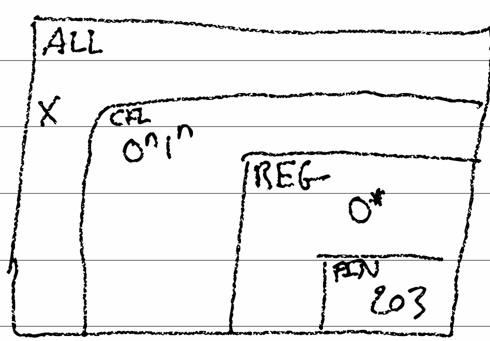
helper g n v = if n=0 , don't return , o.w.
let rules = g.R v
for rhs ← rules ; do
return (V, map s ← rhs ; do
iterator
if s ∈ V, helper g (n-1) s
o.w. s)

S → 0 S 1 → 00 S 11

REG = a language defined by some DFA
CFL - context-free languages = a lang defined by a CFG

VS → S O V
S → NP | A NP | A A Np |
Np → N | P N
V → ... | Adver V

9-5



$0^n1^n \notin REG$

$0^n1^n \in CFL$

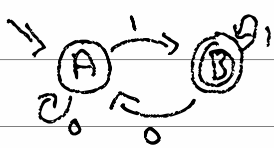
$REG \subsetneq CFL$

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

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

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

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



$$A \Rightarrow \overset{①}{1B} \mid \overset{②}{0A}$$
$$B \Rightarrow \overset{③}{\varepsilon} \mid \overset{④}{1B} \mid \overset{⑤}{0A}$$

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

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

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: $\cup$, $\cap$, $\circ$, $*$, $\overline{}$

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

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

g_1	g_2
$S_1 \rightarrow \dots$	$S_2 \rightarrow \dots$
$V_1 \rightarrow \dots$	$U_1 \rightarrow \dots$
$V_2 \rightarrow \dots$	$U_2 \rightarrow \dots$

$\rightarrow g_1$

$\rightarrow g_2$

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

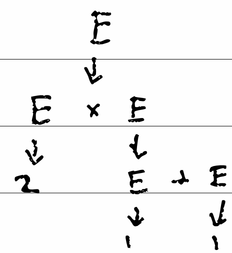
$g_1^* \quad = \quad S \rightarrow \varepsilon \mid S_1 S$

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

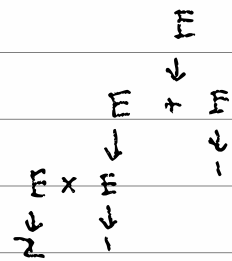
$$E \Rightarrow E \times E \mid E + E \mid 1 \mid 2$$

$2 \times \overset{1}{\cancel{2}} + 1$

pt 1:



pt 2:



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

$$E \Rightarrow E + E \mid F$$
$$F \Rightarrow F \times F \mid 1 \mid 2$$

] — unambiguous

ambiguous? : CFG → Bool

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

LL(k) LALR LR

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

$$x + y = y + x$$

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

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

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

CFG == NFA (Noam)

$\Updownarrow \qquad \Updownarrow$ CNF = Chomsky Normal Form

CNF == DFA

A grammar g is in CNF iff

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

or $r = S \Rightarrow \varepsilon$

or $r = A \Rightarrow a$

$a \in \Sigma$

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

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

$X \to AS$

$Y \to a$

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

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

$B \to b$

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 $V \to A$

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

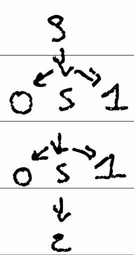
$S \to 0S1 \mid 01$ add them

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

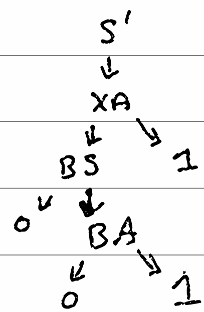
$S \to XA \mid BA$

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

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



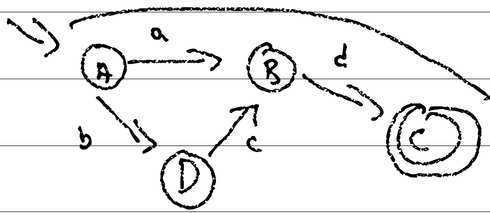
N-ary tree



Binary tree!

CNF parse trees are binary!

11-1/ Generating a String accepted by DFA



→ ad

gsabh {set of unvisited nodes} × path to here × here

gsab (DFA d) = gsabh $(d.Q - d.q_0)$ ε $d.q_0$

gsabh Remaining Path q_i =

```
if q_i ∈ d.F then return Path
if Remaining is empty then return FALSE
for (c, q_j) in δ(q_i) :
    if q_j ∈ Remaining :
        p = gsabh (Remaining - q_j) (Path ++ c) q_j
        if p false : return p
return FALSE
```

11-2/ $011\varepsilon0 = 0110$

"011" ∘ "" ∘ "0" = "0110"

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

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

$\to \Sigma \cup \{\varepsilon\}$

$\leftarrow Q$

Task #27

In Task #30, 0110

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

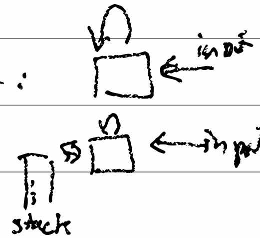
DFAs machine PDA

$\iff$ REX $\iff$ PL $\iff$ CFG

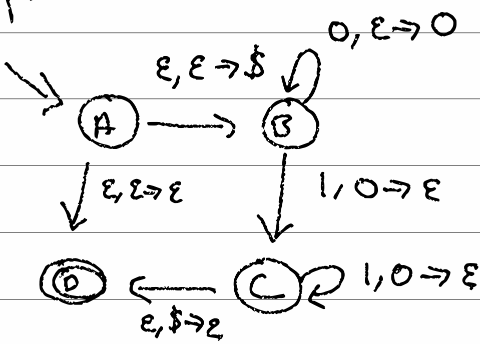
PDA - push-down automata

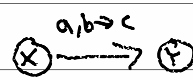
DFA:

PDA:



$0^n 1^n$





Read a from input

Pop b from stack

Push c to stack

config = stack $\times$ Q $\times$ input $\Rightarrow$ stack[Q] input

ε[A]0011 $\Rightarrow$ \$[B]0011 $\Rightarrow$ \$0[B]011 $\Rightarrow$ \$00[B]11

$\Rightarrow$ \$0[C]1 $\Rightarrow$ \$[C]ε $\Rightarrow$ ε[D]ε $\Rightarrow$ ✓

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

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

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

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

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

11-4/ pda-oracle P $(os : \text{List}\ (\Sigma \cup \{\varepsilon\}, \Gamma \cup \{\varepsilon\}, Q, \Gamma \cup \{\varepsilon\}))$

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 ε

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

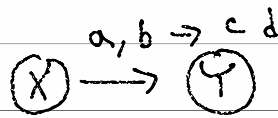
NFA

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

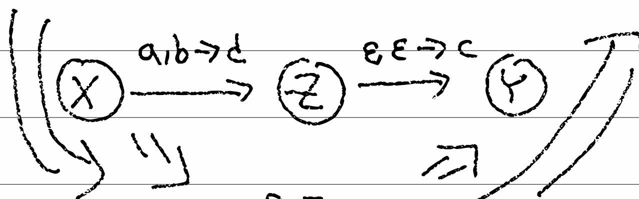
PDA stack

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

11-5/ Can a PDA push multiple things?



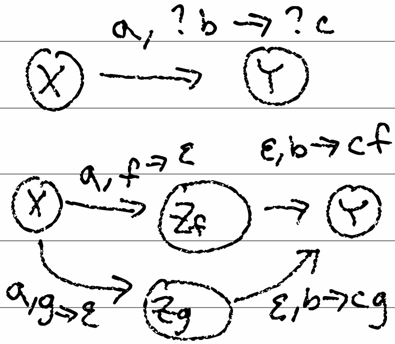
$u\,b\,[X]\,a\,v \Rightarrow u\,d\,c\,[Y]\,v$



$\Downarrow$ $u\,d\,[Z]\,v$ $\Rightarrow$

Can a PDA read more than 1 back?

$f \in \Gamma$



$u\,b\,f\,[X]\,a\,v$ $\quad$ $u\,b\,g\,[X]\,a\,v$

$u\,b\,[Z_f]\,v$ $\quad$ $u\,b\,[Z_g]\,v$

$u\,c\,f\,[Y]\,v$ $\quad$ $u\,c\,g\,[Y]\,v$

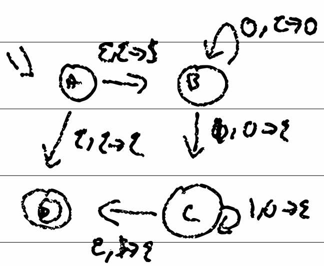
Z_i forall $i \in \Gamma$

11-6

DFA	A	$\to$	a B
CFG	A	$\to$	a B c C d
???	ABC	$\to$	a B c C d
TM	AbCd	$\to$	a B c D

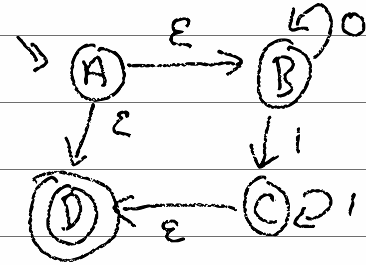
$= Q' = Q \times \Gamma^n$

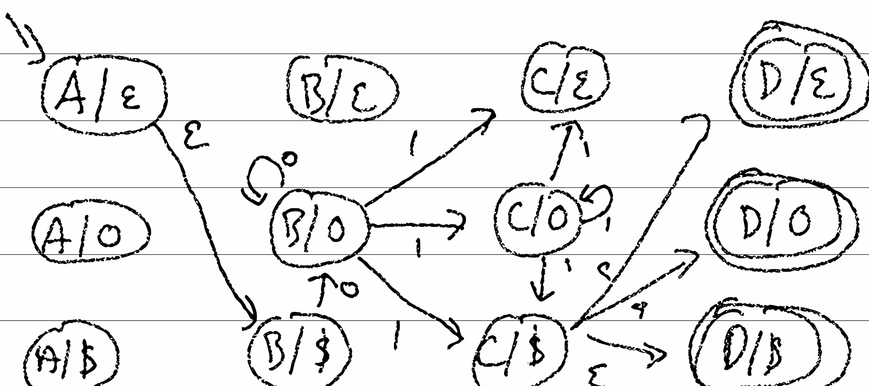
PDA to DFA (PDA p) (Nat n)



$0;$

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





12-1 CFG → PDA

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

assume in CNF $R = A \to a$

$A \to BC$

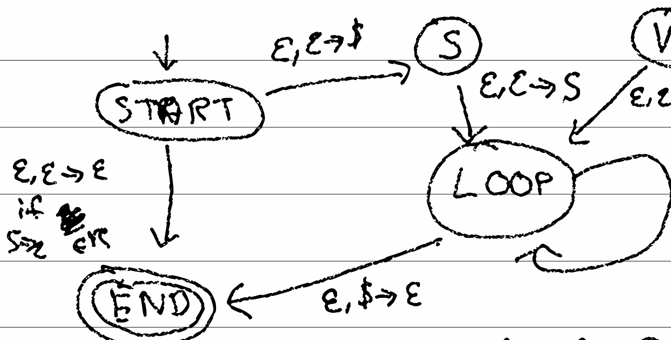
$S \to \varepsilon$

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

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

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

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



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

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

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

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

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

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

input : 000111

ε [START] 000111 $\rightarrow$ \$ [S] $0^3 1^3$ $\rightarrow$ \$S [LOOP] $0^3 1^3$ $\rightarrow$

\$1S0 [LOOP] $0^3 1^3$ $\rightarrow$ \$P [LOOP] $0^2 1^3$ $\rightarrow$ \$11S0 [L] $0^2 1^3$ $\rightarrow$

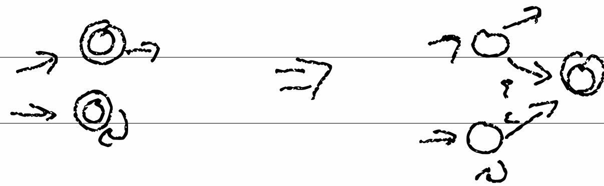
\$11S [L] $0 1^3$ $\rightarrow$ \$111S0 [L] $0 1^3$ $\rightarrow$ \$$1^3$ [L] 1^3 $\rightarrow$ \$$1^3$ [L] 1^3

\$$1^2$ [L] 1^2 $\rightarrow$ \$1 [L] 1 $\rightarrow$ \$ [L] $\rightarrow$ [END] $\rightarrow$ ✓

12-3 / PDA → CFG

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

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



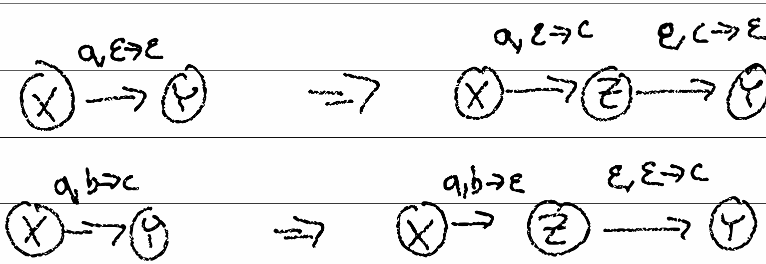
assume 2 : every transitions pushes XOR pops

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

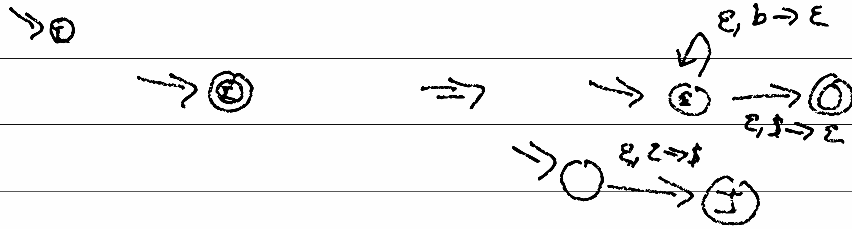
pop : $a, b \to \varepsilon$ (poped b)

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

X replace : $a, b \to c$



assume 3 : the stack is empty on accept



$\forall b \in \Gamma,\ b \neq \$$

$\$ \notin \Gamma$

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

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

$S = (q_0, q_f)$

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

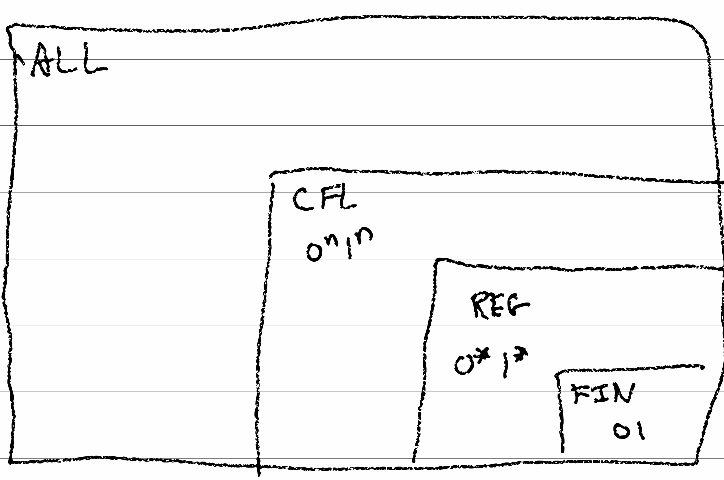
$(r, t) \in \delta(p, a, \varepsilon)$

$(q, \varepsilon) \in \delta(s, b, t)$

$(p, q) \to a\ (r, s)\ b$

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

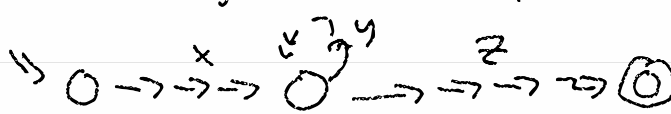
15-1



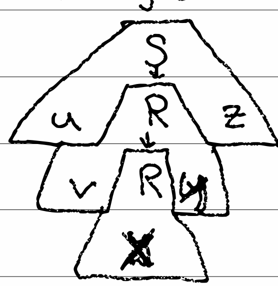
$ALL \neq CFL \Leftarrow \exists x \in ALL.\ x \notin CFL.$

$\Leftarrow \exists P.\ (\forall x \in CFL.\ P(x)) \wedge (\exists y \in ALL.\ \neg P(y))$

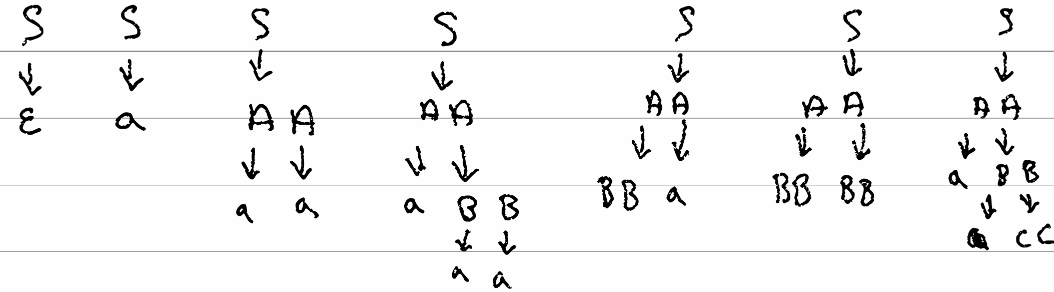
Regular PP. $s = xyz$ and $xy^iz \in A$



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



15-2/ Suppose G is CFG in CNF



$S \Rightarrow AA \Rightarrow$

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

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

How many chars are in a tree of depth K?

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

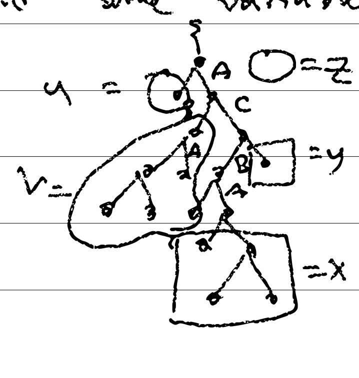
How deep is the tree of a string(accepted) with N chars?

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

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

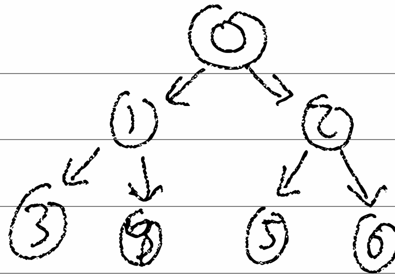


$S \to uAz$

$A \to vAy$

$A \to x$

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) =

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

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

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

&& $|vy| > 0)$

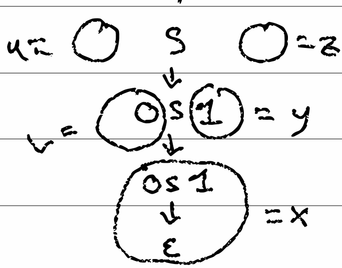
$\forall i \in \mathbb{N}.$

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

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

$S \Rightarrow \varepsilon \mid 0 S 1$

$s = 0011$



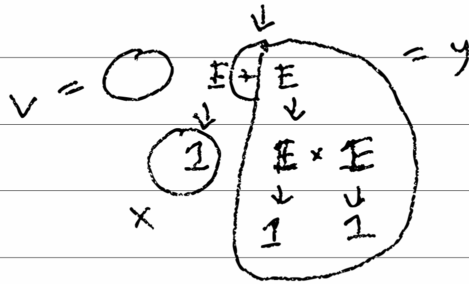
$u = \varepsilon$ $z = \varepsilon$

$v = 0$ $y = 1$

$x = 01$

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

$1 + 1 \times 1$ $u = \bigcirc$ E $\bigcirc = z$



$u = \varepsilon$

$v = \varepsilon$

$x = 1$

$y = +1 \times 1$

$z = \varepsilon$

$uv^0 x y^0 z = 1$

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

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

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

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

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

$\qquad \& \; |vy| > 0)$

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

$uv^i x y^i z \notin A$

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

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

00

init: see 0, push 0, goto init

~~replace pop 0, push~~ 00 $\not$ 0

see ~~0~~ 1, ~~pop 0~~, goto mid

~~00 1 0 push, pop 0~~

mid: see 1, ~~pop 0~~, goto mid

see 0, pop 0, goto end

end: see 0, pop 0, goto end

done → yes

[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 s.t. $|vxy| \leq p \wedge |vy| > 0$

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

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

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

case 4: $\overset{0^p}{u}\ \underbrace{1^p\ 0^p}_{vxy}\ z$ (in between 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 $LO,1$

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

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

case 2.1 : $v = 0^{\hat{v}}$ $x = 0^c 1^d$ $y = 1^{\hat{y}}$

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

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

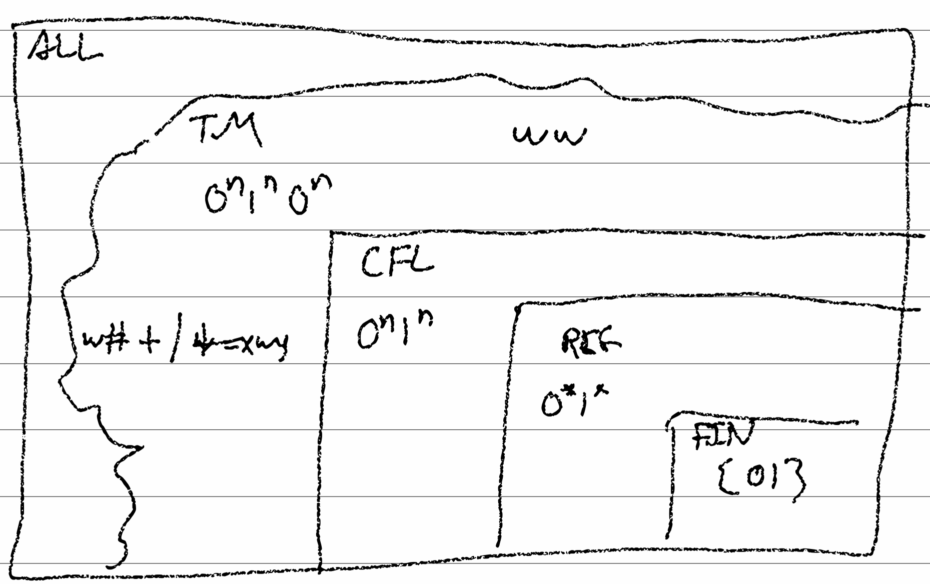
iff $\hat{u} + \hat{v}i + c = d + \hat{y}i + \hat{z} = p$

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

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

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

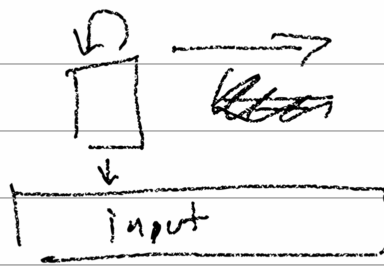
$w \# w \notin$ CFL

$w \# w^R \in$ CFL

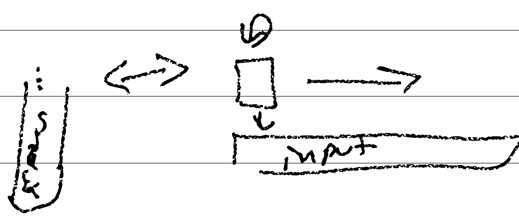
Turing Machines (TM) Alan Turing

- Turing Test -
- ENIGMA

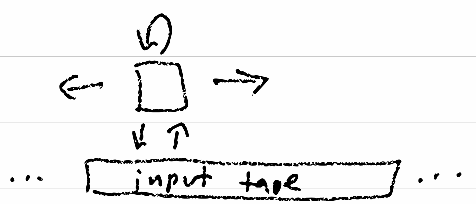
DFA



PDA



TM



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

input state, input sym, output state

PDA: $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to P(Q \times \Gamma_\varepsilon)$

optional input, optional stack pop, nondet, optional stack push

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

tape read, tape write, direction

input, tape

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

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

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

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

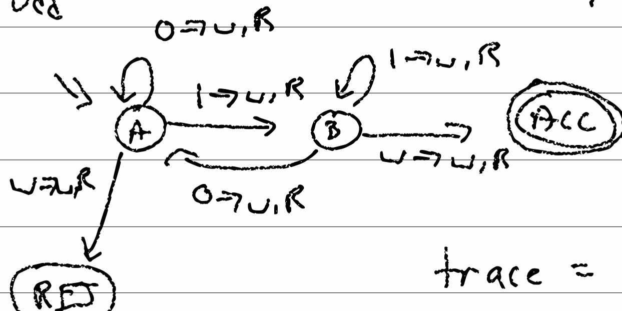
The tape is infinitely long and starts as

$\ldots \sqcup$ input $\sqcup \ldots$

starts here

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

17-3/ Odd



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

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

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

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

$[B] \sqcup \to [ACC] \to$ ✓

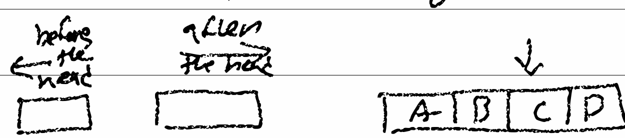
If $d = (Q, \Sigma, q_0, \delta : Q \times \Sigma \to Q, F)$

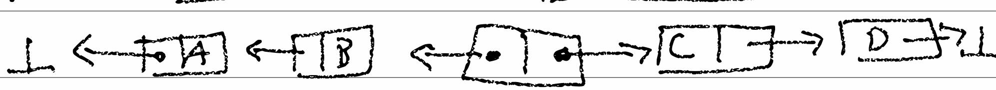
then $t = (Q \cup \{ACC, REJ\}, \Sigma, \Sigma \cup \{\sqcup\}, q_0,$
$\delta', ACC, REJ)$

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

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

REJ if $q_i \notin F$

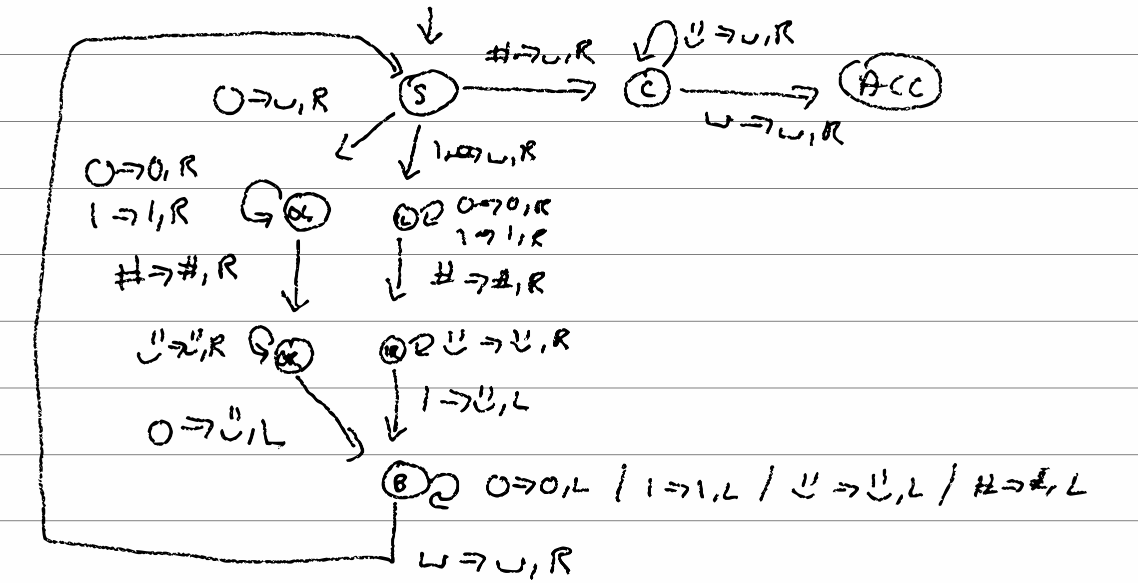
Tape = 



17-4/ w#w

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

$\Gamma = \{0, 1, ☺, ⊔\}$



[S] 010#010 ⇒ [0L] 10#010 $\Rightarrow^2$ 10 [0L] #010 ⇒ 10# [0R] 010 ⇒

10 [B] #☺10 ⇒ [B] ⊔10#☺10 ⇒ [S] 10#☺10 ⇒' [S] 0#☺☺0

[S] #☺☺☺ ⇒ [C] ☺☺☺ $\Rightarrow^3$ [C] ⊔ ⇒ ACC ✓

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

[7-5]

simulate : TM $\times$ input $\to$ trace

simulate t s = h t ([], t.q0, s)

h t $\overset{c=}{n}$(before, q_i, after) = cons cn

(c, after') = case after of

[] $\Rightarrow$ (⊔, [])

c: after' $\Rightarrow$ (c, after')

(q_j, c', d) = t.δ(q_i, c)

case d of

L $\Rightarrow$ case before of

[] $\Rightarrow$ h t ([], q_j, ⊔:c':a')

b: before' $\Rightarrow$ h t (before', q_j, b:c':a')

R $\Rightarrow$ h t (c':before, q_j, after')

h t (before, ACC, after) = (list T)

REJ = (list F)

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

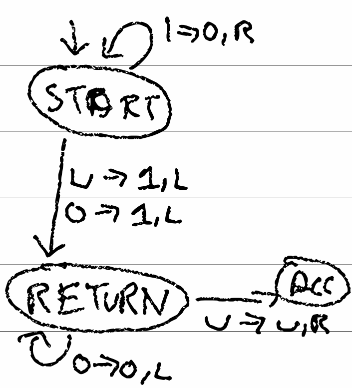
acceptance

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

Computable function

---

add1



add1(0) = 1

add1(1) = 01

add1(01) = 11 ← 1

add1(11) = 001 ← 3

add1(001) = 101 ← 2

[S]01 → [R]⊔11 → [Acc]11

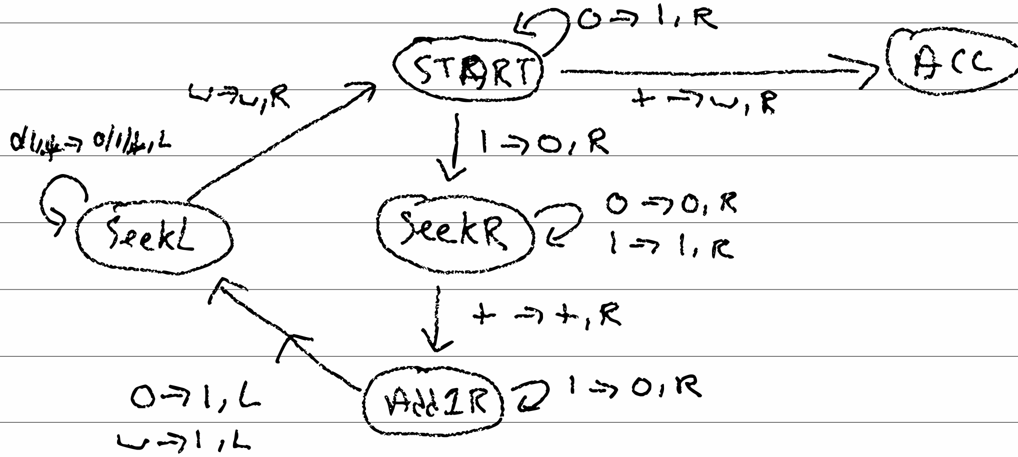
[S]001 → [R]⊔101 → [A]101

[S]11 → 0[S]1 → 00[S]⊔ → 0[R]01

[R]001 → [R]⊔001 → [A]001

18-1) $X + Y$ $\qquad 0+Y \Rightarrow Y$

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



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

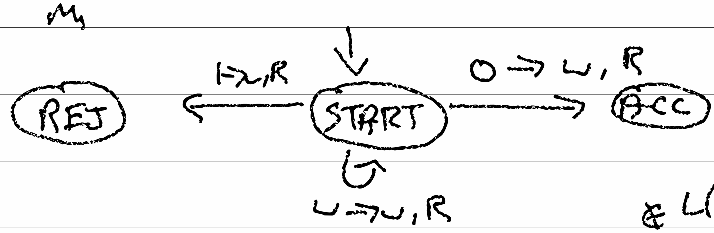
Σ_1

$A \in \Sigma_1$ 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

M_1



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

$\varepsilon \notin L(M_1)$ M_1 diverges

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

18-3

TMs

recognizers	$\supseteq$	deciders
$\parallel$		$\parallel$
May diverge		Never diverge
$\forall x \in \Sigma^*,\ M(x)=ACC$		$\forall x \in \Sigma^*.\ M(x) = ACC$
$\vee\ M(x)=REJ$		or $M(x)=REJ$
$\vee\ M(x)$ diverge		

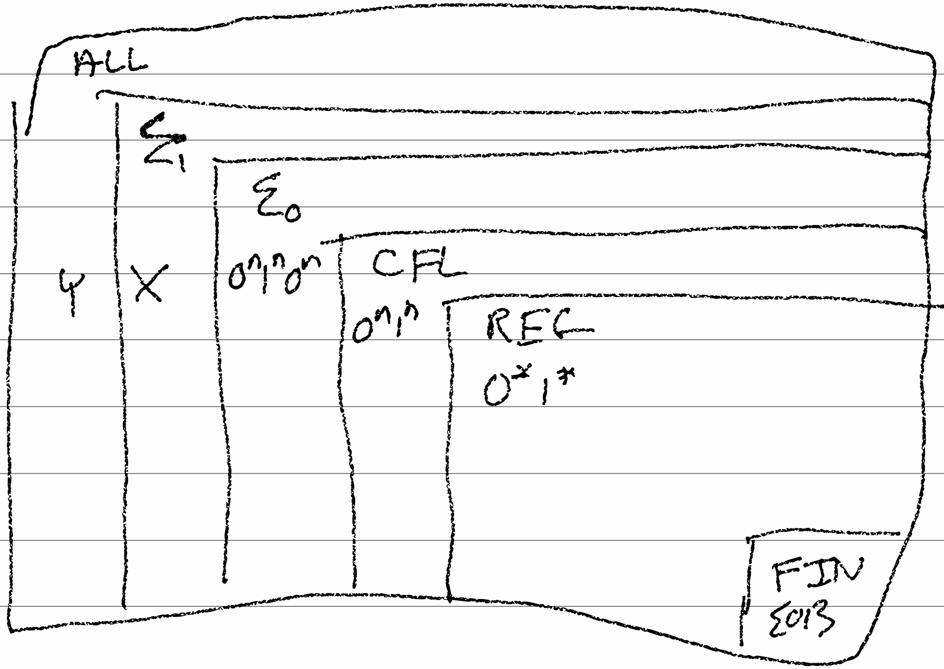
Σ_1 = Turing-recognizable $A \in \Sigma_1$ iff $\exists t \in$ recog. $L(t)=A$

Σ_0 = Turing-decidable

$A \in \Sigma_0$ 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, Char>~~

```
~~Ex~~ 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 $\longleftrightarrow$ NFAs $\longleftrightarrow$ REX

compile : NFA $\Rightarrow$ DFA

decompile : DFA $\Rightarrow$ NFA

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

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

Galois-connection

19-3) Stay-Put TM

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

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

$$\frac{\delta(q_i, a) = (q_j, b, L)}{u\,c\,[q_i]\,a\,v \Rightarrow u\,[q_j]\,c\,b\,v} \qquad \frac{\delta(q_i, a) = (q_j, b, R)}{u\,[q_i]\,a\,c\,v \Rightarrow u\,b\,[q_j]\,c\,v}$$

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

$$\begin{array}{ccc}
u\,[\hat{q}_i]\,a\,v & \longleftarrow & u\,[q_i]\,a\,v \\
\downarrow R & & \Downarrow \quad g \\
u\,b\,[NS]\,v & & u\,[q_j]\,b\,v \\
\downarrow L & \nearrow & \\
u\,[\hat{q}_j]\,b\,v & &
\end{array}$$

$$q_i \xrightarrow{a \Rightarrow b, S} q_j$$

$$q_i \xrightarrow{a \Rightarrow b, R} NS \xrightarrow{x \Rightarrow x, L} q_j$$

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

$0 \Rightarrow 0, L$

$1 \Rightarrow 1, L$

$\omega \Rightarrow \omega, L$

19-3/ Multi-tape TM

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

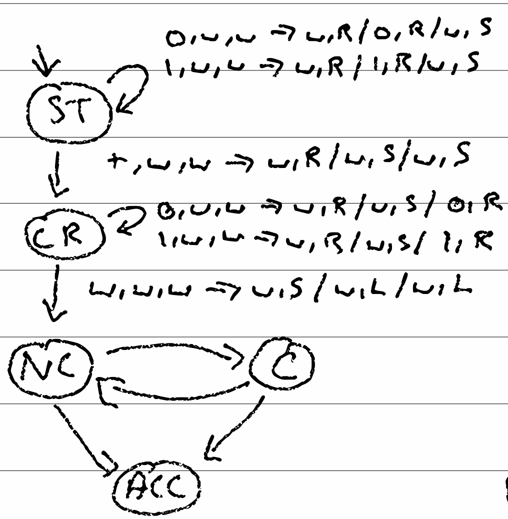
$$\delta: Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{L, R, S\})^k$$

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

$$\begin{array}{c} uc \\ f \end{array} \left[q_i \right] \begin{array}{c} av \\ \cancel{x}g \end{array} \Rightarrow \begin{array}{c} u \\ f\ y \end{array} \left[q_j \right] \begin{array}{c} cbv \\ zg \end{array}$$

binary addition MTM $\qquad$ $10 + 01 = 11$

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



$\left[ST\right]\ x + y$

$x \left[CR\right]\ y$

$\left[\ \ \right]$

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 $\cap$

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

X/ $q_{x_1} \xrightarrow{a \to b, R} q_{x_2}$

Y/ $q_{y_1} \xrightarrow{c \to d, L} q_{y_2}$

X∪Y/ $q_{x_1, y_1} \xrightarrow{a, c \to b, R, d, L} q_{x_2, y_2}$

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

α : $\begin{matrix} u \\ x \end{matrix} [q_i] \begin{matrix} av \\ by \end{matrix} \longrightarrow [q_i]\, u \dot{a} v \# x \dot{b} y$: B

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

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

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

$q_i \longrightarrow q_i/L$

$\Longrightarrow q_i/L^{\Delta}/q_i$

$q_j \longleftarrow$ seek deq $/q_j \longleftarrow$

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) (Y sees tape 0)

check both finish

Non-det TM

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

forking: $u[q_i]av$

$\swarrow \qquad \searrow$

$u[q_L]av \qquad u[q_R]av$

config: seq (det-config)

$\varepsilon \Rightarrow REJ$

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

$u[q_a]v; \ldots \Rightarrow ACC$ where and

$u[q_i]av; REST \Rightarrow REST; u'[q_j]v'$ $u', v' = \text{tape}(u, a, v, b, d)$

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

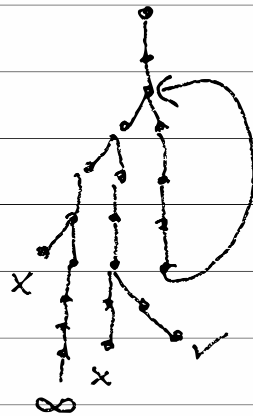
iff $\delta(q_i, a) = (q_L, q_R)$

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

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

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

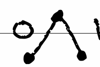
20-3/ Back-tracking non-det TM



breadth-first-search

rop = $(0 \cup 1)^*$

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

10-4

$f(x)$ $g(y)$

$g(f(x))$

PL for TM:

e = By-hand
| $e \cup e$ | $e \cap e$
| $e \circ e$ | e^*
| $e(e)$

---

enumerator = ...

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

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

$Q - \{q_a, q_r\}$

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

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

$\in Q_P$

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

enum $\Rightarrow$ TM :

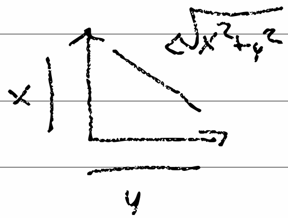
TM $\Rightarrow$ enum :

21-1

Church-Turing Thesis

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

"Add" $\quad +$



$x + 0 = x$

$x + y = y + x$

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

$\Rightarrow$ A is unsolvable

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

$\Rightarrow$ A is undecidable

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

$\Rightarrow$ A is un/infeasible / intractable

21-2) 1900 - World Congress of Mathematics

David Hilbert ~~proposed~~ 15 problems
proposed

for $[1900, 2000]$

Polynomial Root Problem

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

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

A polynomial over n variables $(x_0 \ldots x_n)$ is defined by co-efficients $a_{0 \ldots n}^{0 \ldots m}$ of degree m

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

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

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

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

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

a b c

$\cancel{-7 + 21x - 7x^2 - 1x^3 + 2x^4}$

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

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

$c_{max} = 7$

$[-5, 5]$ 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 $w \in \Sigma^* \}$

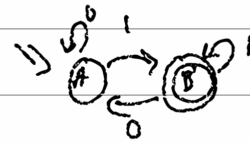
DFA $d = (Q, \Sigma, q_0, \delta, F)$ $\Sigma = \{0, 1\}$

$1^n 0$ 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 $q_i \in F$



$\delta =$

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

$|Q| = 110$ 11000

$q_0 = 0$ $= 0011$

$F = 01$

$\delta = 0011$

1100010011 01101 $\in A_{DFA}$

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

22-1/ $A_{TM} = \{ \langle M, w \rangle \mid$ 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 $H \in \Sigma_0$

$H(\langle M, w \rangle) =$ accept if M accepts w
reject if M does not accept w
~~NO LOOP~~

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

$D(\langle M \rangle) =$ accept is M does not accept $\langle M \rangle$
reject if M accepts $\langle M \rangle$

Run D on $\langle D \rangle$ = accept if D does not acc $\langle D \rangle$
reject if D accepts $\langle D \rangle$

What answer is returned? $\rightarrow$ 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 $\neq$ decidable

recognizers $\neq$ deciders

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

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

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

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

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

~~interleave~~ run Y on w

if X acc, we acc

if Y acc, we reject

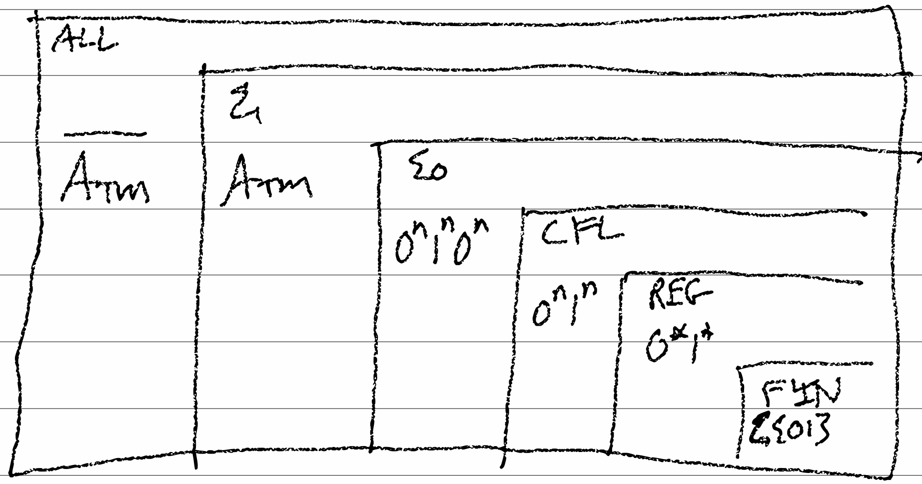
$A_{TM} \notin \Sigma_0 = \neg(A_{TM} \in \Sigma_0)$

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

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

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

22-3



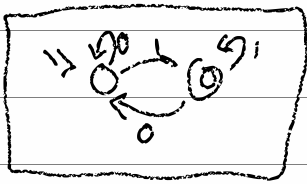
ALL
$\overline{A_{TM}}$
Σ_1
A_{TM}
Σ_0
$0^n1^n0^n$
CFL
0^n1^n
REG
0^*1^*
FIN

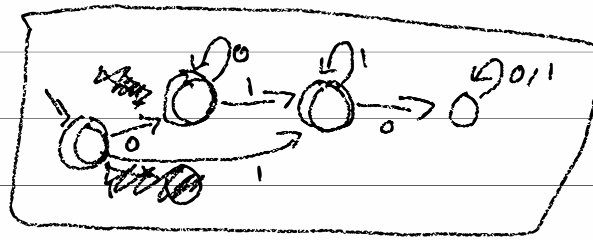
22-4) When are two sets the same size?

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

$3 = 3 \Rightarrow \checkmark$

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





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

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

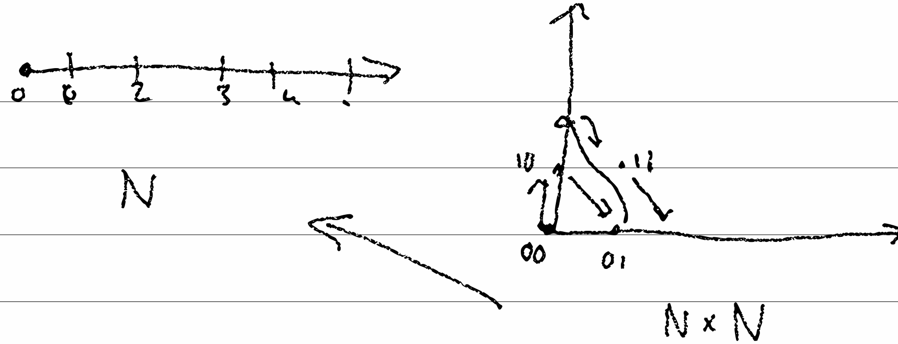
The natural numbers are the same size as the even numbers

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

$f(x) = 2x$

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

22-3)



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

Cantor-pairing function

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

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

$N \approx N \times N = N^2$

$N \approx (N \times N) \times N$

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

$|TM| < N^{10}$

$N \approx \Sigma^*$ (lexi) $\quad$ lexi : $N \Rightarrow \Sigma^*$

$A \lesssim N$:= "countable"

22-4/ IBS = infinite binary sequence

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

$0000000\ldots \in IBS$

fun(i){ return 0; }

$f = g$ iff

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

$01010101\ldots \in IBS$

fun(i){ ~~...~~ i % 2 ~~...~~ }

$111100\ldots \in IBS$ = fun(i){ return i<5; }

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

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

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

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

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

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

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

given: f pick: b = fun(x){
return $\neg f(x)(x)$; }

given: a pick: $i = a$

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

Cantor's Diagonalization Theorem

$\neg f(a)(a)$

T	$\neq$	F
F	$\neq$	T

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

$$\begin{array}{ccc} \boxed{N} & \not\approx & \text{IBS} \\ \approx & & \approx \\ \text{TM} & & \text{ALL} \end{array}$$

$\approx \implies \text{TM} \not\approx \text{ALL}$

$\Sigma_1 \implies \Sigma_1 \not\approx \text{ALL}$

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

$\parallel$

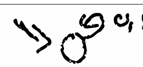
N-bit number

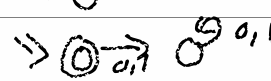
$=$

IBS

$P(\{0,1\}) = P(A) =$ $|A|$-bit number

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

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

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

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