1-1

How do we know if a math formula is true?

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

correct effective

Does an algorithm exist?

What is an algorithm?

Does a program exist? ← problems

What is a program? ← models

1-2] A set is "a bunch of stuff"

$\emptyset$ – nothin' in it

$\forall x, x \notin \emptyset$

$\{$pen, phone$\}$ $\{$phone, pen$\}$

$\{$ ✓ ,  $\}$

~~$\forall$~~ pen $\in$ $\{$pen, phone$\}$

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

union – $\cup$ u u

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

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

1-3) "The set of all true math formulas"

A set IS its membership

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

"Is there a god?"

"Will Buffy be remake?"

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

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

The <u>Universe</u> ($\mathcal{U}$)

subset

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

1-4 Our universe is made of strings
and strings are sequences of characters
and chars are elements of an alphabet
an alphabet is a finite set

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

chars chars

"0100001" = a string = s

length = 7 $s(0) = 0$ $s(1) = 1$ $s(2) = 0$

$U = \Sigma^*$ ← special notation

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

epsilon = "" = the string w/ no characters

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

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

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

L5/ #1. Decide a data type to represent alphabets and characters.

```
Alphabet  = List<Character>
Character = Object    | void*
```

we need equality

#2. Decide a data type for Strings

```
interface String { }
class MtString implements String { .. }
class OneString impl String {
    OneString ( Char c, String s ) { ... } }
Zero = new BasicChar('0');   One = new BC('1');
010 = new OneS( Zero, new OneS(One, new
          OneS( Zero, new MtS())));
```

$\{\Sigma\}$

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

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

newList<Chars>
{ Zero, One};

$$
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
\varepsilon & 0, & 00, & 000 \\
 & 1, & 01, & 001 \\
 & & 10, & 010 \\
 & & 11, & 011, \\
 & & & 100 \\
 & & & 101 \\
 & & & 110 \\
 & & & 111 \\
2^0 & 2^1 & 2^2 & 2^3
\end{array}
$$

$|A|^i$ where i = layer I'm looking at

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

lexi B $0 = \varepsilon$

lexi B $1 = 0$

lexi B $2 = 1$

lexi B $6 = 1\cancel{0}1$

2-1/ "1+1" $\longrightarrow$ "2"

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

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

$\emptyset$ $\{x\}$ $A \cup B$

Alphabet Σ Universe Σ^*

$\{0, 1\}$ $\{$ ε, 0110, 000001,

 $\}$

$\mathbb{P}(A)$ 2^A

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

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

$\emptyset \in P(A)$ $\emptyset \subseteq A$ 0110 $\{1, 2\}$

$\{0\} \in$ ″ $\{2, 3\} \in$ $\{0, 2, 3\}$ ↓

$\underline{0}12\underline{3}$

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

. . . .

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

$\{\varepsilon\} \in P(\Sigma^*)$ 0011, ...

all even lengthed strings $\in P(\Sigma^*)$ = { ε, 00, 11, 01, }

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

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

2-2/ $ALL = P(\Sigma^*)$

FIN = the set of finite sets

ALL
- True math
- GIFs
- Even strings

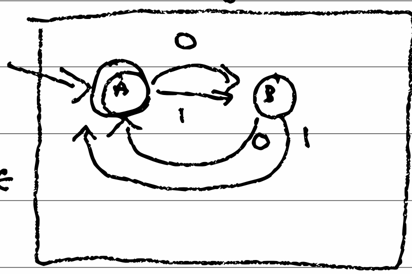
FIN

$\emptyset \in FIN$

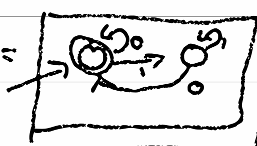
$\forall x \in \Sigma^* .\ \{x\} \in FIN$

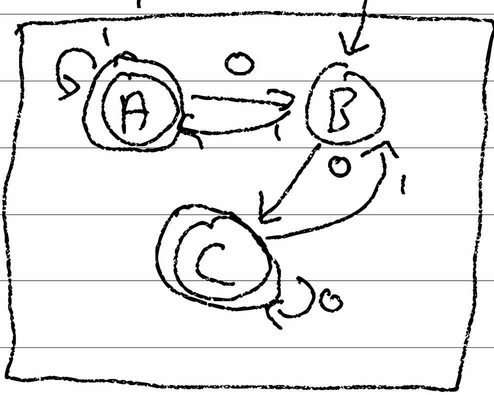
$A \in FIN \wedge B \in FIN$
$\rightarrow A \cup B \in FIN$

All even strings =

$x \in$ 

ε 01 ~~11~~

even nums = 



DFA - a deterministic finite automata

○ - states $\{A, B\}$

→○ - start state A

◎ - accepting states $\{A\}$

$○ \xrightarrow{x} ○$ - transition

x - labels are Σ

	0	1
A	B	B
B	A	A

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

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

	0	1
A	B	A
B	C	A
C	C	B

states, alphabet, start, accepting

2-3) $x \in DFA\ (Q,\ \Sigma,\ q_0 \in Q,\ F \subseteq Q,$
$\delta : Q \times \Sigma \Rightarrow Q$ – transitions)

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

config update function : config $\times$ DFA $\Rightarrow$ config
$[q]w \longrightarrow [q']w'$

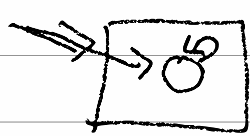
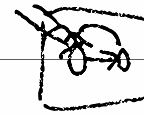
$[q_i]\, x y \Rightarrow [q_j]\, y$ iff $\delta(q_i, x) = q_j$

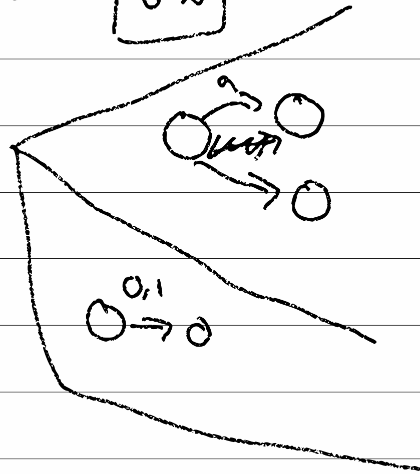
$x \in$ DFA iff $[q_0]\, x \Rightarrow \Rightarrow \Rightarrow \Rightarrow\Rightarrow [q_f]\, \varepsilon$
and $q_f \in F$

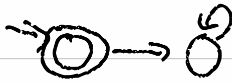
$0110 \in$ EvenLen iff $[A]0110 \Rightarrow [B]110 \Rightarrow [A]10$
$\Rightarrow [B]0 \Rightarrow [A]\varepsilon$ $A \in \{A\}$ ✓

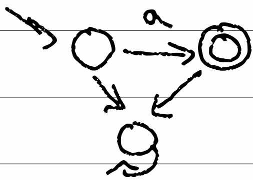
```
class DFA {
  ... Q, Σ, F, q0, δ ...
  public bool accepts (String x) {
    State qi = q0;
    while (!x.~~isEmpty~~ empty()) {
      qi = δ(qi, x.first());
      x = x.rest(); }
    return F.in(qi); } }
```

2-4/

 $= \emptyset =$ 

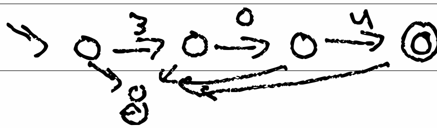


 $= \{\varepsilon\}$

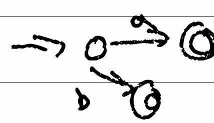


$= \{a\}$

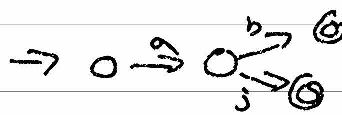
$=$ 

$\{304\} =$ 

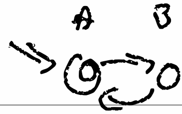
$A \cup B \in DFA$ (if $A \in DFA$ and $B \in DFA$)

$\{a\} \cup \{b\}$ 

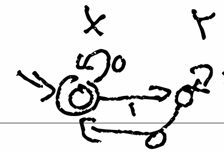
$\{a;\} \cup \{ab\}$



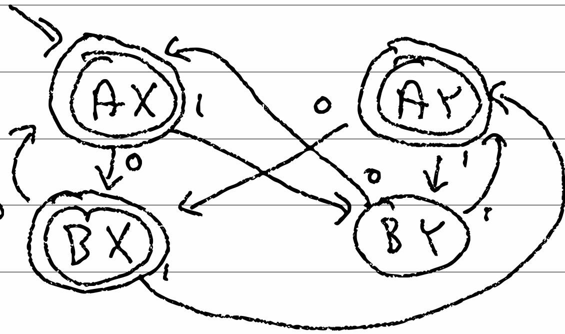
2-5] EvenLen



IsEven



$\cup$



ε ✓
00 ✓
11 ✓
0 ✓
110 ✓

$\cap$

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

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

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

$X = A \cup B$

$Q_X = Q_A \times Q_B$

$q_{0X} = (q_{0a}, q_{0b})$

$F_X = F_A \times F_B$ — $\cap$

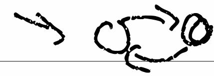
$F_A \times Q_B \cup Q_A \times F_B$ — $\cup$

$\delta_X((q_a, q_b), c) = (\delta_a(q_a, c), \delta_b(q_b, c))$

$x \in A \cap B$ iff $x \in A \wedge x \in B$

2-6/ $x \in A^c$ iff $x \notin A$ $(x \in U)$

Even Len odd len

 $\Rightarrow$ 

$F = \{A\}$ $F' = Q - F$

or F^c (wrt Q)

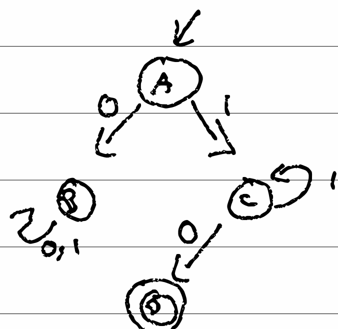
complement

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

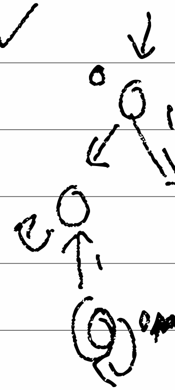
3-1 DFA ⇒ example or false

DFA :

- Q : State → Bool
- Σ : list of characters
- q_0 : state
- δ : (state × (char)) → State
- F : State → Bool



10 ✓



Q	∅	q_0	
{A, B, C, D}	{}	[A]	A ⇒ ε
{B, C, D}	{A}	[B, C]	B ⇒ A, 0 C ⇒ A, 1
{C, D}	{A, B}	[C]	Yes, it's possible.'
{D}	{A, D, C}	[D]	D ⇒ C, 0
{}	{A, B, C, D}	[]	or if not, No!

10

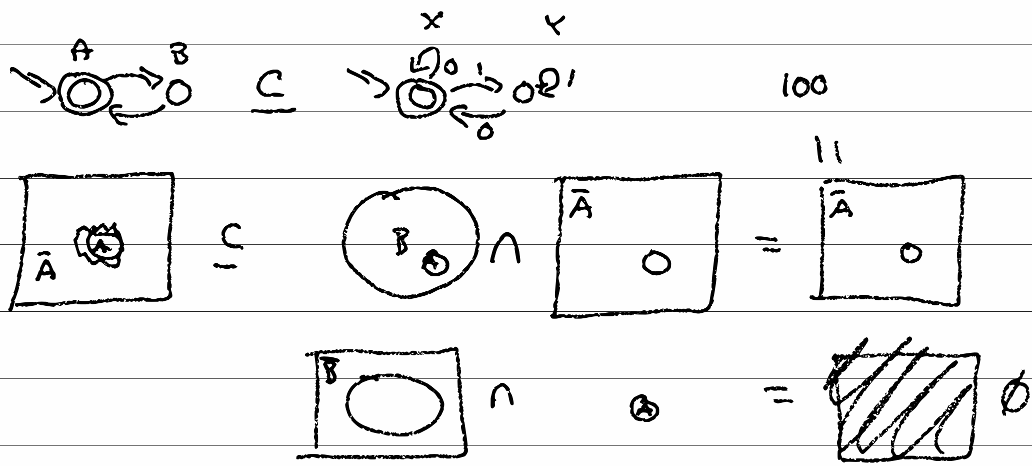
3-2/ subset

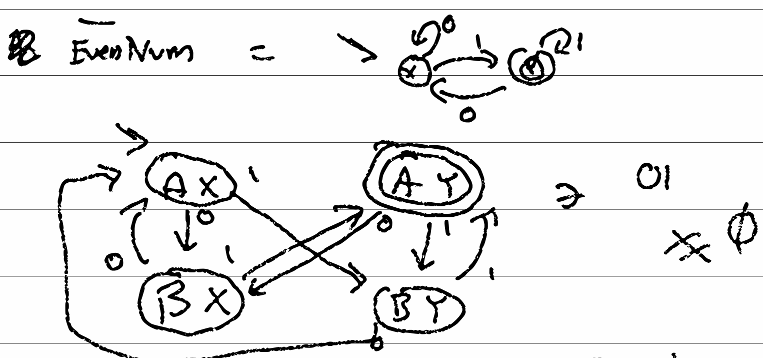
$$A \subseteq B \quad \text{iff} \quad \neg\ \forall x \in U.\ x \in A \Rightarrow x \in B$$

$$\exists x \in U.\ x \in A \wedge x \notin B$$

$\{a, b\} \subseteq \{a, b, c\}$ $\quad U = \{a, b, c\}$

finite means naive works!





model checking

Soundness : model $\subseteq$ theory

completeness : theory $\subseteq$ model

model = theory

3-3/ $0, 1, 2, -1, 5$ $\qquad$ Z, P, N

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

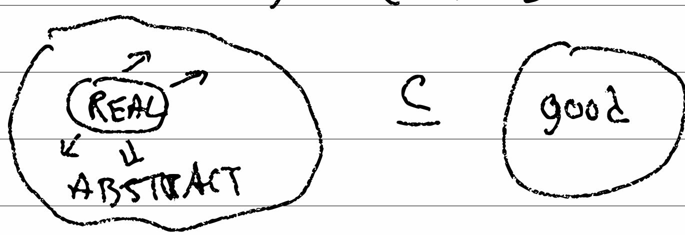
if $x > 0$ then

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

o.w

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

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



3-4) Finite = $\emptyset \quad \{x\} \quad A \cup B \quad A^c \quad A \cap B \quad A \circ B$ $\in$ DFA

Infinite = A^*

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

$\varepsilon \circ y = y$ if $a \in \Sigma$, $(a \circ x) \circ y = a \circ (x \circ y)$

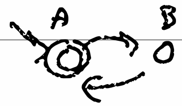
$abcd = a b \circ c d$

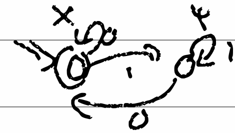
$\{j\} \circ \{m, n\} = \{jm, jn\}$

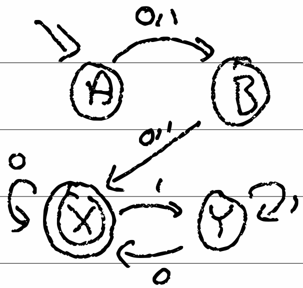
$x \in A^*$ iff $x = x_0 \circ x_1 \circ \ldots \circ x_n$ for $n \in \mathbb{N}$ and $x_i \in A$

$\{jm\}^* \ni \varepsilon, jm, jmjmjmjm$

3-5/ Even Len ∘ Even Num



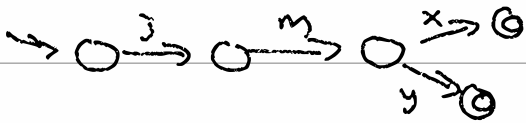


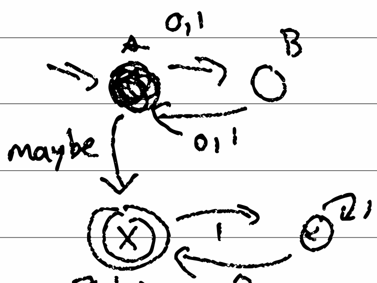


$\underbrace{00}\underbrace{110}$ ✓

0011 ×

$\{jm\} \circ \{x,y\}$

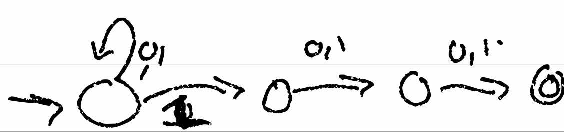




$\Sigma = \{0,1\}$

$x \in$ DFA iff

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



"3rd from end is 1"

3-6/ NFA = non-deterministic
finite automata

old world: the next step was obvious
$\delta: Q \times \Sigma \Rightarrow Q$

new world: crazy options
- do you even read a char?
- which path do you take?

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

$\delta'(A, 1) = \{A, B\}$

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

epsilon

ε Σ e ε

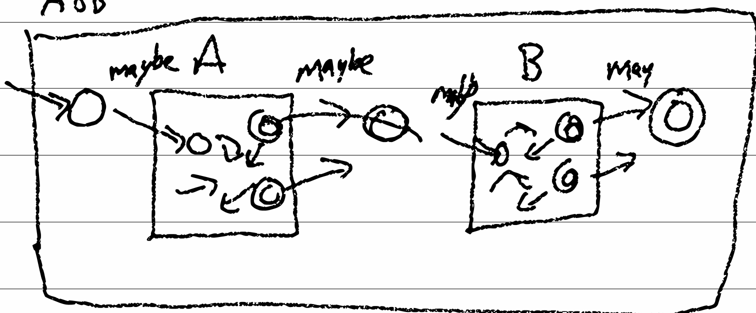
4-1 $A \circ B \in DFA$ iff $A \in DFA$

A^* $B \in DFA$

NFA (N - non D- deterministic)

$\delta : Q \times (\Sigma \cup \{maybe\}) \to P(Q)$ $\delta : Q \times \Sigma \to Q$

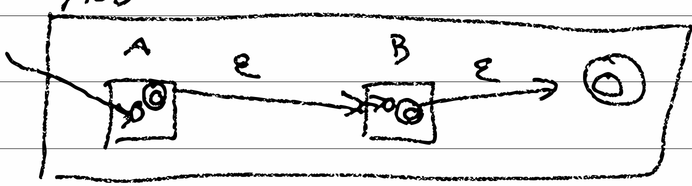
$A \circ B$



maybe written as "ε"

What does (NFA) this mean?

$A \circ B$



NFA ⟺ DFA

4-2 | What do NFAs mean?

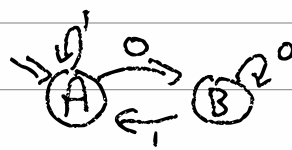
A DFA represents a set and
a set "is" a membership function

$U \to \text{Bool}$

$= \Sigma^* \to \{Y, N\}$

config $= \Sigma^* \times Q$

$\Sigma^* \to Q^*$



$\underbrace{0110 \to ABAAB}$ → a trace

$\Sigma^* \to (\Sigma, Q)^*$ $\qquad \Sigma_e$ ~~make~~ $= \Sigma \cup \{\epsilon\}$

$\underbrace{0110 \to (0,B)\,(1,A)\,(1,A)\,(0,B)}$

0A 1A 1A 0B → N

valid? : $\cancel{Q\times}(\boxed{\Sigma}, Q)^* \to \{Y, N\}$ $\qquad$ DFA

valid q_i ε = Y

valid q_i (c, q_j) : more = if $\delta(q_i, c) \boxed{==} q_j$
valid q_j more
o.w. N

Nvalid? : $Q \times (\boxed{\Sigma_e} \times Q)^* \to B$

Nvalid q_i ε = Y

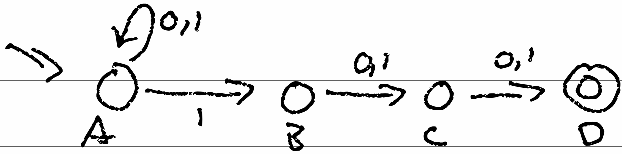
Nvalid q_i (c, q_j) : more =

if $\boxed{q_j \boxed{\in} \delta(q_i, c)}$ then $\qquad$ An Oracle

Nvalid q_j more

o.w. N

4-3



"3rd letter from end is 1"

0100	111	110100	— Y
000	1000	1011	— N

$(0,A)(1,A)(0,A)(0,A)$ ✓

$(0,A)(1,B)(0,C)(0,D)$ ✓

$\underbrace{(0,B)}_{\delta(A,0)=\{A\}}(1,C)(1,D)\underbrace{(0,D)}_{\delta(D,0)=\emptyset}$ X

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

str $\varepsilon = \varepsilon$

str $(c,-)$: move $= c$

$c \circ$ str move

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

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

valid q_0 $t = Y$

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

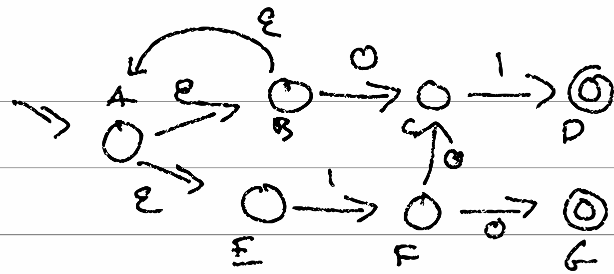
NFA-accepts : $\Sigma^* \to Y/N$

figure all possible traces

check if valid and is strings match

check if last is in $\in F$

4-4



(ε, B) (0, C) (1, D)   01 = ε0001

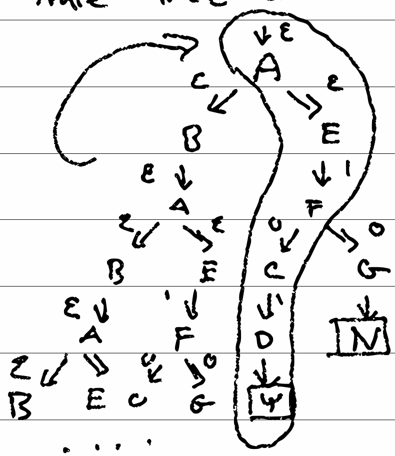
(ε, E) (1, F) (0, C) (1, D)   101 = ε010001

(ε, E) (1, F) (0, G)   10 = ε0100

(ε, B) (ε, A) x   where x is valid

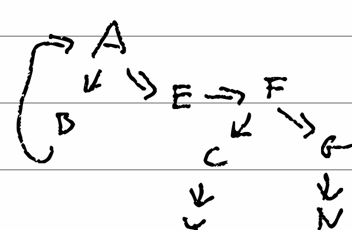
⇒ valid

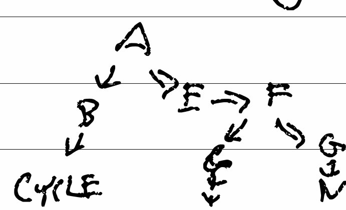
Trace Tree = Y | N | Branch (Σ, Q) (List TT)



101

ε,E   1,F   0C   1D



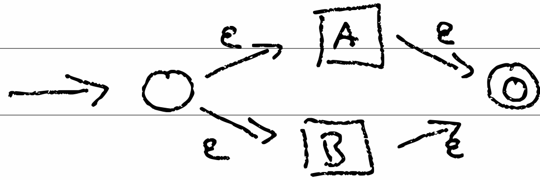


Forking model of NFAs (make TT)

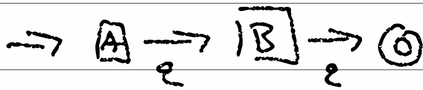
Back-tracking model   (explores TT)

4-5/

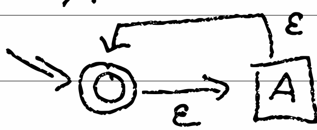
$A \cup B$

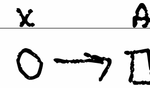


$A \circ B$



A^*





State X transitions to THE start state of A

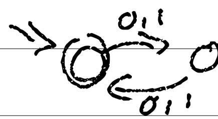
All accepting states of A transition to X

4-6/ $\forall A.\ A \in DFA \iff A \in NFA$

$\Rightarrow$

$\Leftarrow$

DFA $\Rightarrow$ NFA



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

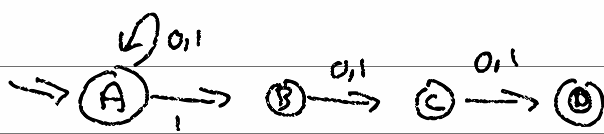
NFA $\delta' : Q \times \Sigma_\varepsilon \to P(Q)$

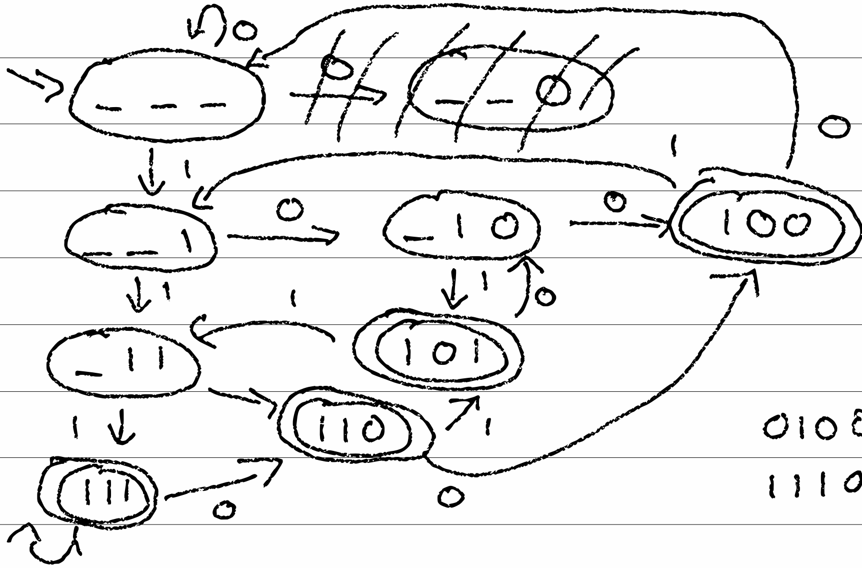
$$\delta'(q_i, \varepsilon) = \emptyset$$

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

NFA $\Rightarrow$ DFA

3rd is 1





0100

111001

4–7/

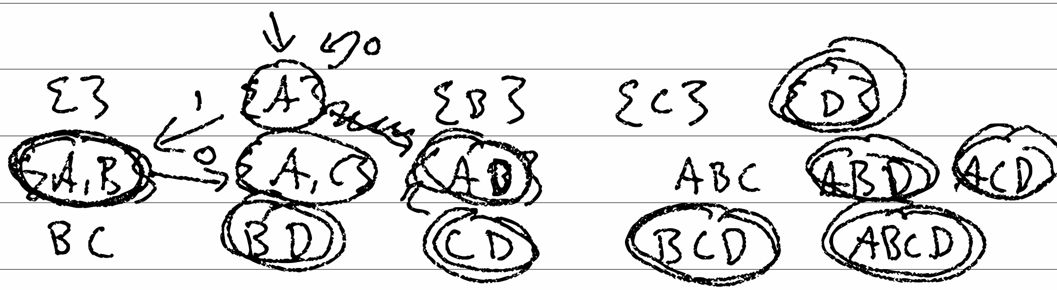
in

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

out

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

$Q' = P(Q)$



$q_0' = \{q_0\}$

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

$\delta'(\{q_i \ldots q_n\}, c) =$

$\bigcup \delta(q_i, c)$

5-1/ $A \cup B$

$$\delta_A : Q_A \times \Sigma \to Q_A$$

$$\delta_B : Q_B \times \Sigma \to Q_B$$

$$\delta' : \underbrace{Q}_{(Q_A \times Q_B)} \times \Sigma \to Q$$

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

char

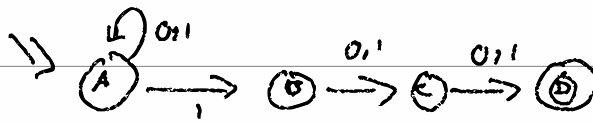
(p, c) => new Pair(deltaa(p.fst, c), deltab(p.snd, c));

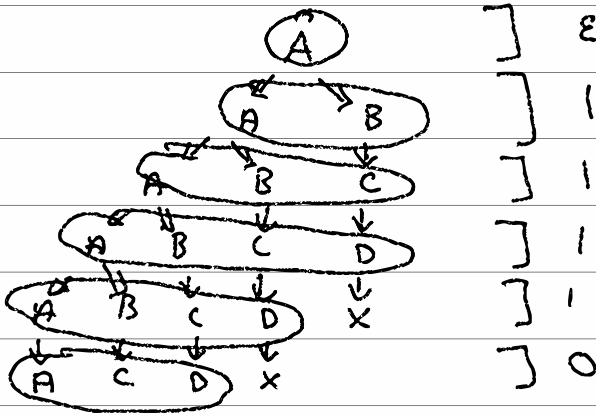
pair<State, State>

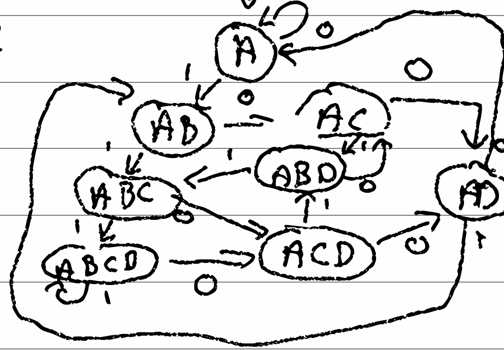
fst snd

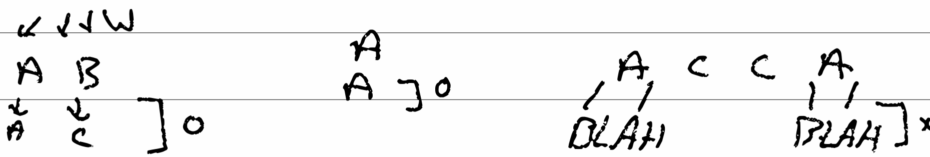
5-2/ NFA $\Rightarrow$ DFA

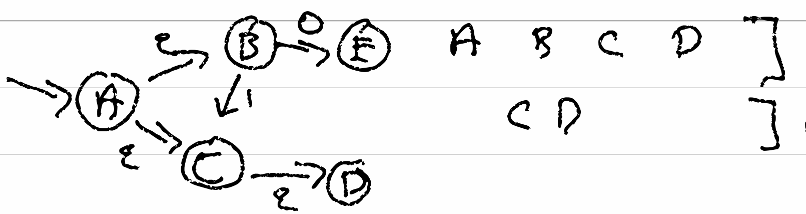
$$(Q, \Sigma, q_0 \in Q, \; \delta : Q \times \Sigma_\varepsilon \to P(Q)), \; F \subseteq Q) \;\Rightarrow\; (Q', \Sigma, q_0' \in Q', \; \delta' : Q' \times \Sigma \to Q', \; F' \subseteq Q')$$











$5\text{-}3$

$$E : Q' \to Q' \quad \text{— follow all } \varepsilon\text{-transitions}$$
$$P(Q) \to P(Q)$$

Trace Tree DFA

Q' = things at the bottom of a trace

set = a set of states of the NFA = $P(Q)$

q_0' = the top of the tree

= the set that has only the first state

$= E(\{ q_0 \}) \in P(Q)$

δ' = maps the bottom of the tree to the next level

= set of all next states of each state in the level of the tree

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

F = any level of three with some accepting state

= any set with an element in F

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

= (set-of-qs →

foreach q_i in set-of-qs

if dfa.F.apply(q_i) then
return true

return false

5-4)

E (set <Q> q_i)

queue <Q> ~~next~~ = ~~empty~~ q_i

set <Q> seen = empty

while (next isn't empty)

add ~~next.first~~ to seen

δ(next.first , ε) add those to next unless in seen

return seen

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

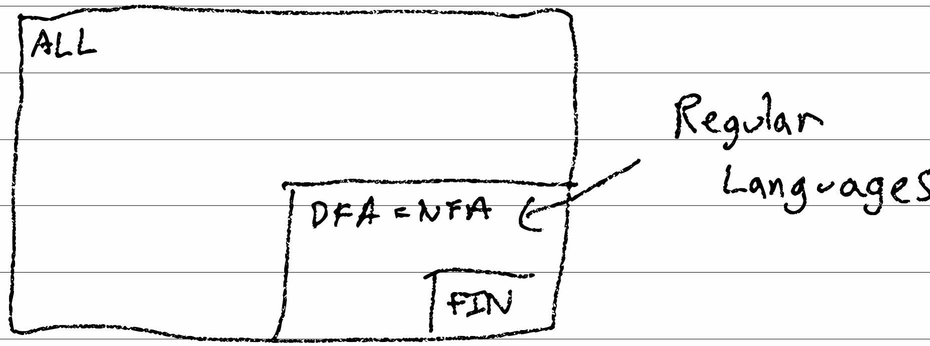
$$E^*(A) = A \cup \bigcup_{q_i \in A} \delta(q_i, \varepsilon)$$

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

$N = D$

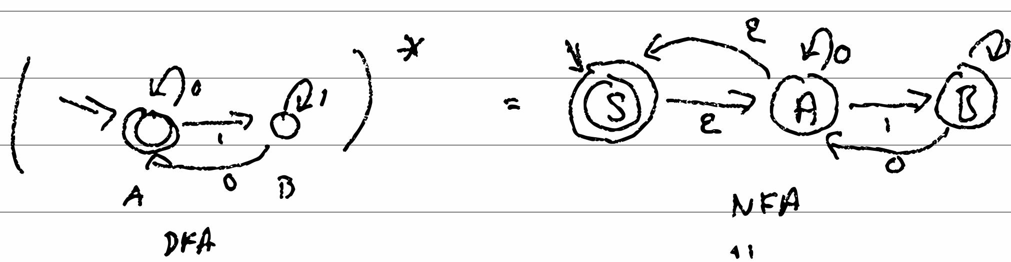
$\forall D \in DFA.\ \exists N \in NFA.$

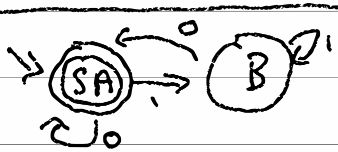
$D = N$



program f .... "f ; g"

program g .... compositional



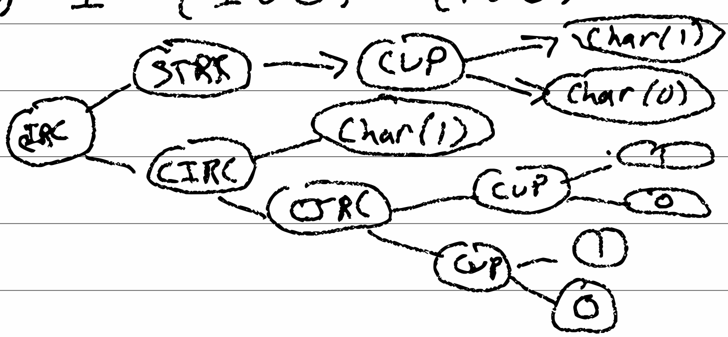


6-2 / Regular Expressions

re :=	ε	EMPTY
	$\emptyset$	NULL
	c	Char
	re $\cup$ re	CUP
	* re	STAR
	re $\circ$ re	CIRC

```
interface RegEx { }
class RE_EMPTY() impl REgex {{ ...
    RE_NULL()
    Re_Char (Char c)
    Re_cup (Regex lhs, Regex rhs)
```

$(1\cup 0)^* \circ 1 \circ (1 \cup 0) \circ (1 \cup 0)$ → "3rd from end is 1"



new Circ(new Star(
  new Cup(
    new Char(1),
    new Char(0))
  ...)

6-3/ $L : RE \to ALL = P(\Sigma^*)$

(ie the language it describes)

(or some.)

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

$L(\emptyset) = \emptyset$

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

R NAME   OR

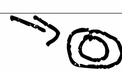
$L(r \cup r') = L(r) \cup L(r')$

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

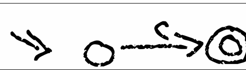
$L(r^*) = L(r)^*$

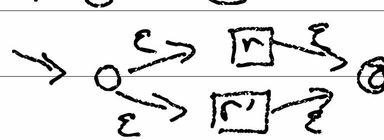
```
class Cup { L() {
        return union(lhs.L(), rhs.L()) }}}
```

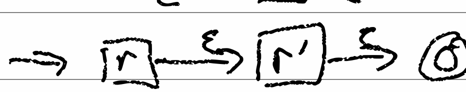
compile : RE → NFA

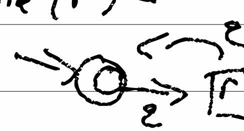
compile (ε) = 

compile ($\emptyset$) = 

compile (c) = 

compile ($r \cup r'$) = 

compile ($r \circ r'$) = 

compile (r^*) = 

6-4)

"" $= \varepsilon$

$\underline{\qquad}$ $= \emptyset$

"c" $= c$

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

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

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

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

"(a|b|c)" $=$ " $(a, b, c, \in \Sigma^*)$

[012]

(zero | one | ~~two~~)

. $= \Sigma$

".*\.mjs" $= \Sigma^* \circ '.' \circ 'm' \circ 'j' \circ 's'$

6-5

gen : RE $\rightarrow$ Σ^* or false

gen ε = ε

gen $\emptyset$ = FALSE

gen c = 'c'

flip coin

$$\text{gen } x \cup y = \overset{H}{\text{gen}}\ x \ \| \ \overset{T}{\text{gen}}\ y$$

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

$$\text{gen } x^* = \boxed{\cancel{\varepsilon}}$$

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

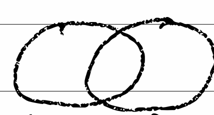
", m;s"

equal : RE × RE $\rightarrow$ Bool

equal x y =

NFA2DFA(compile x)

NFA2DFA(compile y) — dfa equality?



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

6-6

$x + 0 = x$ $\qquad$ $x = x$

$x \cdot 1 = x$

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

$2(3x + 17) = 6x + 34$ $\qquad$ "algebra"

$3x + 17 = 3x + 17$

$3x = 3x$

$x = x$ $\qquad$ $1776 = 1776$?

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

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

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

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

$= \varepsilon \cup \emptyset = \varepsilon$

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

NFA 2 DFA $\qquad$ in: N states $\quad (Q)$

out: 2^N states $\quad (\mathcal{P}(Q))$

6-7/ $\text{size} : RE \to Nat$

$\text{size } \emptyset = 1$

$\text{size } \varepsilon = 1$

$\text{size } c = 1$

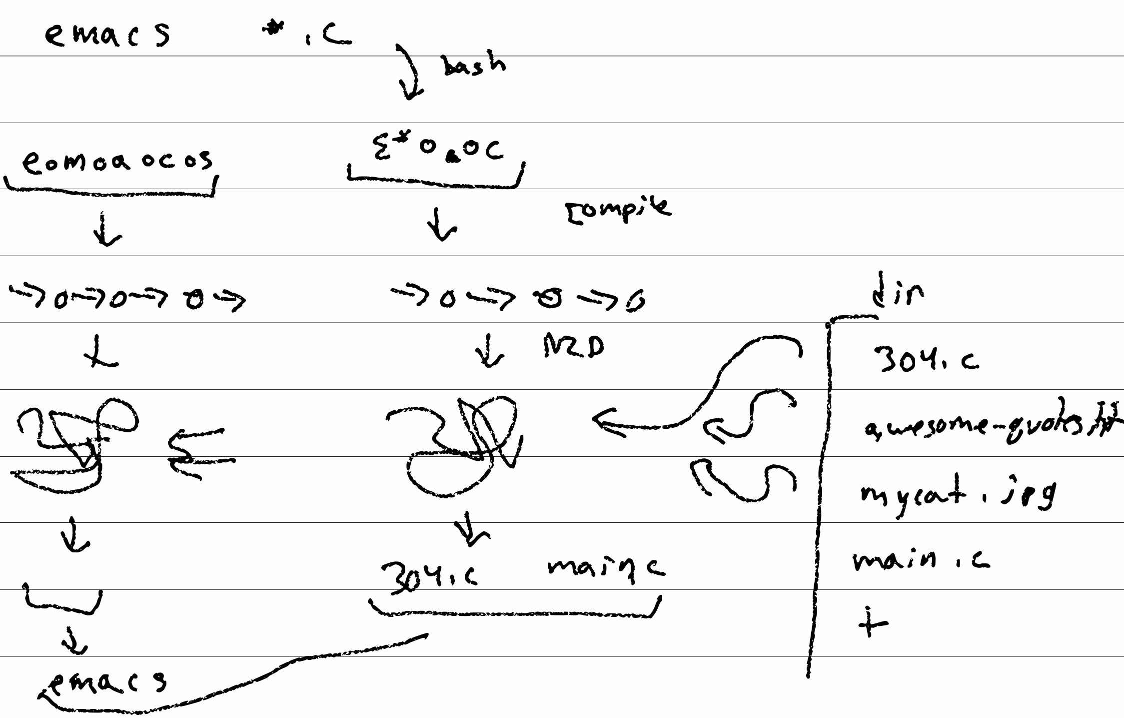
$\text{size } (x \cup y) = sz\ x + sz\ y + 2$

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

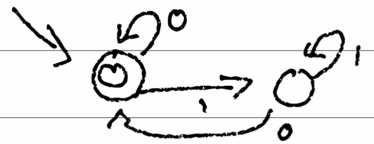
$\text{size } (x^*) = sz\ x + 1$

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

$\uparrow$

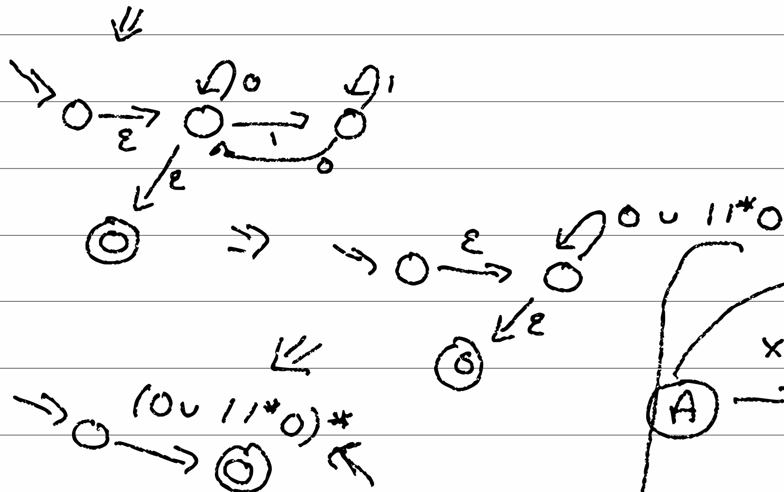


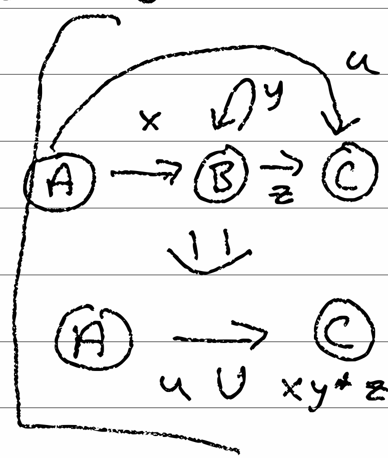
6-8 | decompile : (NFA) or DFA → RE



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

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





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

ε 0 10 111110

01110 1111100

111000111011100101010

G-9/ decompile : N-state NFA

$\Rightarrow$ RE

START : N-NFA $\Rightarrow$ (N+2)-GNFA

RIP : (N+1)-GNFA $\Rightarrow$ N-GNFA

END : 2-GNFA $\Rightarrow$ RE

decompile m = end $\circ$ rip^n $\circ$ start (n)

7-1 DFA/NFAs $\Rightarrow$ RE

$$\text{DFAs} \longleftrightarrow \text{NFAs} \longleftrightarrow \text{RE}$$

Regular Languages

NFA $\Rightarrow$ RE

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

RIP^n: $(n+1)$-GNFA $\Rightarrow$ n-GNFA

OUT: 2-GNFA $\Rightarrow$ RE

$$\text{GNFA} = (Q, \Sigma, \underset{}{\overset{\in Q}{q_s}}, \overset{\in Q}{q_e}, \Delta : (Q - q_e) \times (Q - q_s) \Rightarrow RE(\Sigma))$$

$\delta : Q \times \Sigma \Rightarrow Q$

OUT: 2-GNFA $\Rightarrow$ RE

$(\{q_s, q_e\}, \Sigma, q_s, q_e, \{((q_s, q_e), r)\})$

$\Rightarrow r = \Delta(q_s, q_e)$

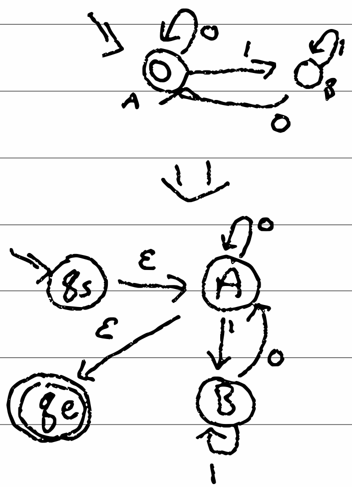
$$\rightarrow (q_s) \xrightarrow{r} (q_e)$$

"Find r"

7-2) IN: NFA → GNFA

$(Q, \Sigma, q_0, \delta : Q \times \Sigma_\varepsilon \to P(Q), F)$ → $(Q', \Sigma, q_s, q_e, \Delta : (Q' - q_e) \times (Q' - q_s) \to RE)$

$Q' = Q \cup \{q_e, q_s\}$



$\Delta(q_i, q_j) = r$

$\Delta(q_s, q_0) = \varepsilon$

$\Delta(q_s, q_j \neq q_0) = \emptyset$

$\Delta(q_f \in F, q_e) = \varepsilon$

$\Delta(q_i \notin F, q_e) = \emptyset$

$\Delta(q_i, q_j) = \bigcup \{ c_{\in \Sigma_\varepsilon} \mid \delta(q_i, c) \ni q_j \}$

7-3/ RIP = (n+1)-GNFA → n-GNFA

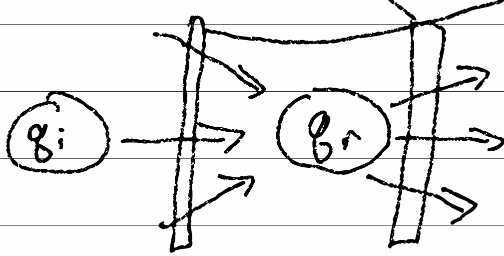
$(Q, \Sigma, q_s, q_e, \Delta)$ → $(Q', \Sigma, q_s, q_e, \Delta')$

main
↓
jay main
↓
$\boxed{e}$
↓
g
↓
exit jay
↓
exit

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

$$\Delta' : \overbrace{(Q' - q_e)}^{q_r \notin} \times \overbrace{(Q' - q_s)}^{q_r \notin} \to RE$$

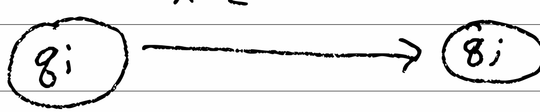
$$\Delta : \underbrace{(Q - q_e)}_{q_r \in} \times \underbrace{(Q - q_s)}_{q_r \in} \to RE$$



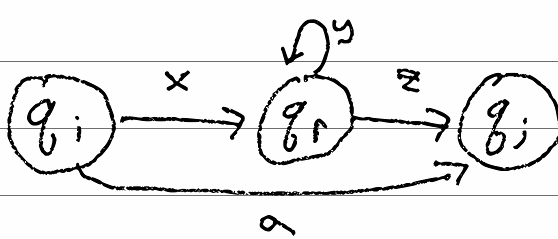
$x \circ y^* \circ z \cup a$

$\Delta'(q_i, q_j) =$

$\cancel{x \circ z}$



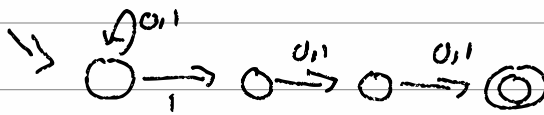
$x = \Delta(q_i, q_r)$ ∘
$y = \Delta(q_r, q_r)^*$ ∘
$z = \Delta(q_r, q_j)$ ∪
$a = \Delta(q_i, q_j)$



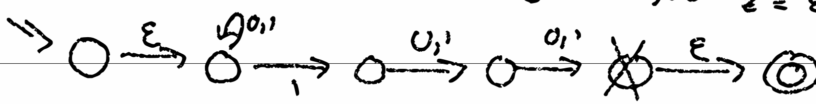
7-4/

```
int main() {
    int x = 8;
    ~~int y = f(x, z);~~
    ....
    int y = x + 2 * z;
```

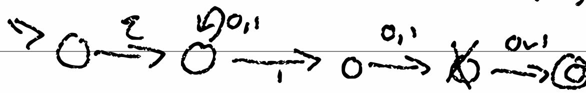
```
int f(int z){
    int 8
    return
    z + 2*8;}
```

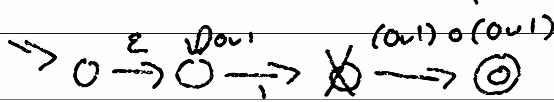


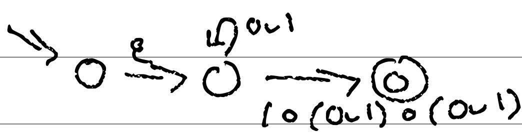
$0\cup 1 = x \quad y=\emptyset \quad z=\varepsilon$ IN $\quad a=\emptyset$



$x\circ y^*\circ z \cup a = (0\cup 1)\circ\emptyset^*\circ\varepsilon\cup\emptyset$
$= (0\cup 1)$

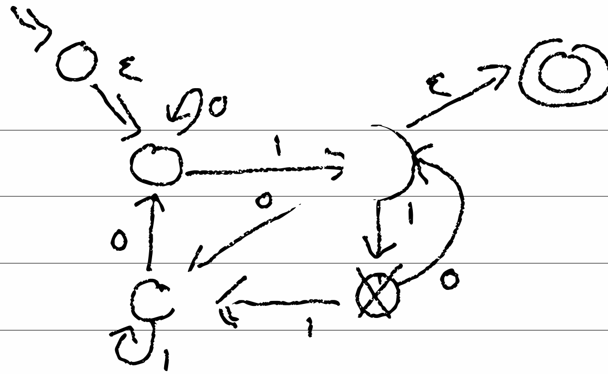


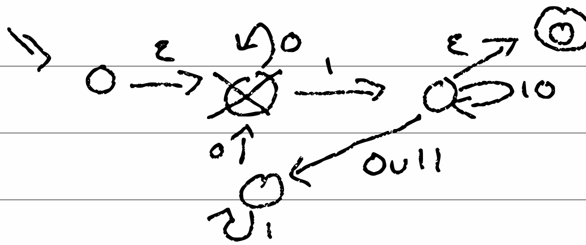


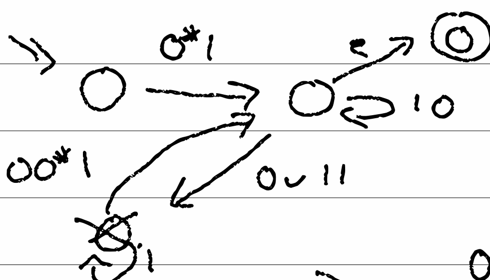


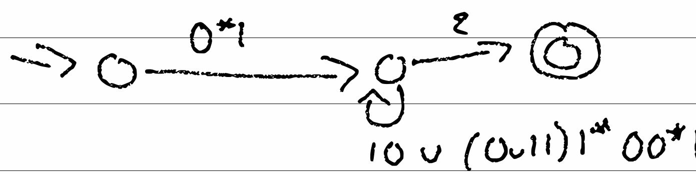
$\varepsilon\circ(0\cup 1)^*\circ 1\circ(0\cup 1)\circ(0\cup 1) = \Sigma^* 1\Sigma\Sigma$

7-5









$(0^*1)(10 \cup (0\cup 11)1^*00^*1)^*$

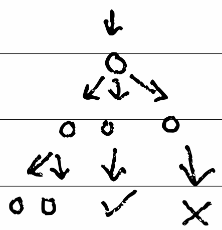
8-1 ε

Object — Char(char c)

Object — Epsilon()

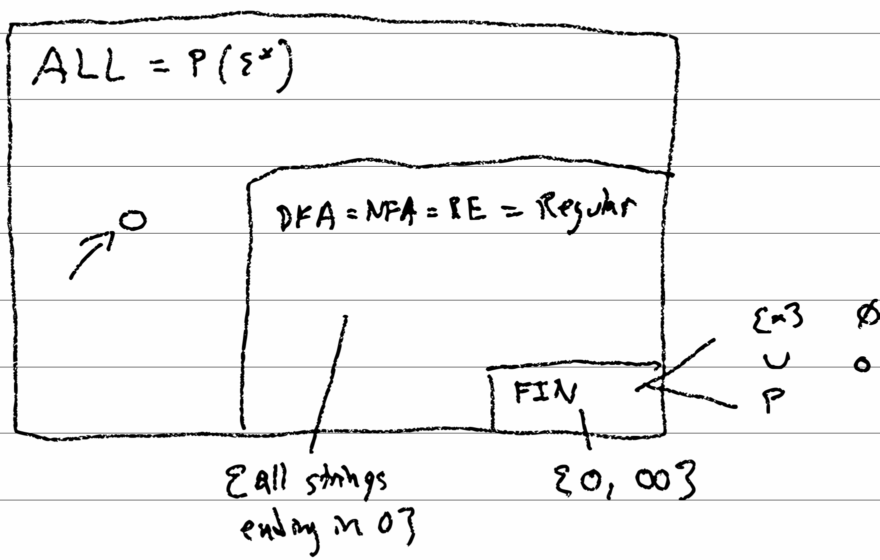
UTF-8

TT



$TT = Succ \mid Fail \mid Branch\ State\ List(TT)$

List(TT) — Char $\times$ TT



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

$\neg A \in REG \iff \neg(\ \uparrow\ ) \iff \forall d.\ d \in DFA \wedge L(d) \neq A$

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

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

8-2 How can know stuff about infinite sets?

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

$P : DFA \to Prop$

$\Rightarrow \quad q_0 \in Q$

$P : \mathcal{P}(\Sigma^*) \to Prop$

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

0. What is P?
1. Prove $\forall A \in REG.\ P(A)$ ← Pumping lemma
2. Prove $\exists B \in ALL,\ \neg P(B)$

---

conclude $B \notin REG$.

$\Sigma = \ldots$

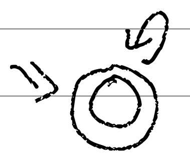
8-31

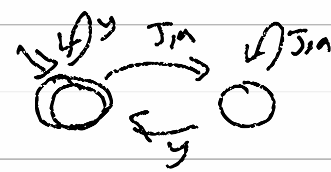
"Jay" $\in d$ $\Sigma \supseteq \{J, a, y\}$

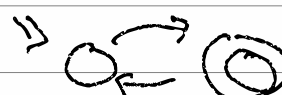
$\Rightarrow q_0 \xrightarrow{J} \xrightarrow{a} \xrightarrow{y} q_f \in F$

$\boxed{\times}$ $\boxed{\checkmark}$ $\boxed{\checkmark}$ $\boxed{\checkmark}$ $\boxed{\times}$

$|d.Q| = 1$







Daphne wins

pick a number of states; 4

she picks a char

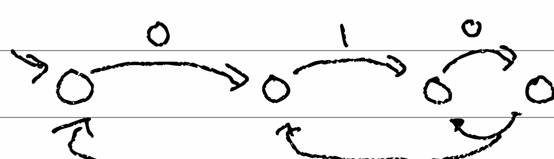
I say what state we goto

I win if I never say same state

she wins if I repeat

How many turns to win?

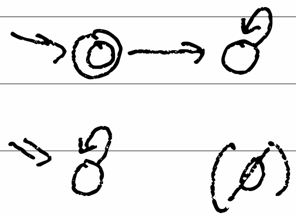
$Y \Rightarrow N \Rightarrow |Q|$



8-4/ $\delta : Q \times \Sigma \to Q$ Total Fun

from $\nearrow$ (Q) 0,1 $\nearrow$ (Σ)

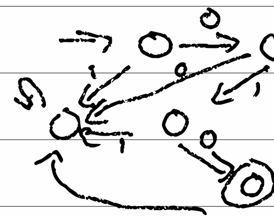
All DFAs have a loop



(ε)

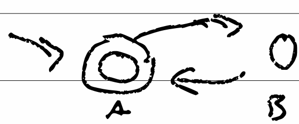
($\emptyset$)

{010}



Some have "exciting" loops

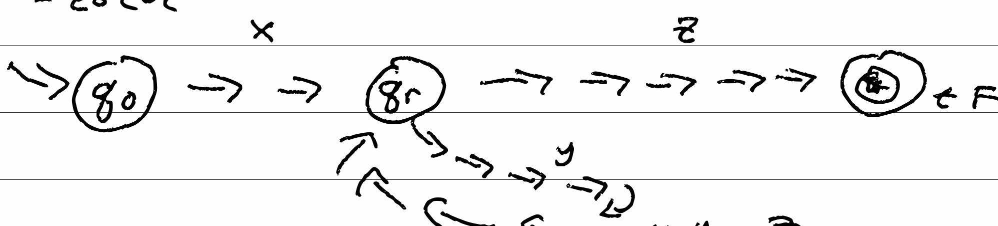
$\exists\, s \in L(DFA)\ .\ s = \ldots\ldots\ \ldots\ldots\ \ldots\ldots$

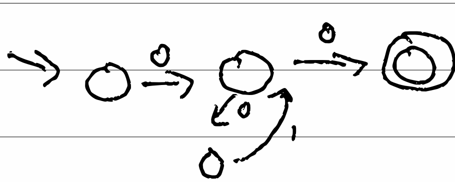


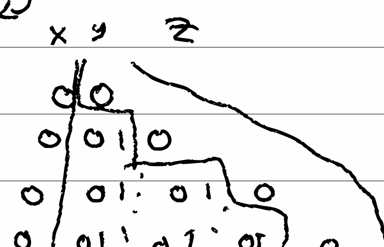
(Even lengthed)

$\varepsilon = A$ $= \varepsilon \circ \varepsilon \circ \varepsilon$

$0110 = ABABA = \varepsilon \circ 01 \circ 10$







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

They are all finite

$L(m) \in FIN$

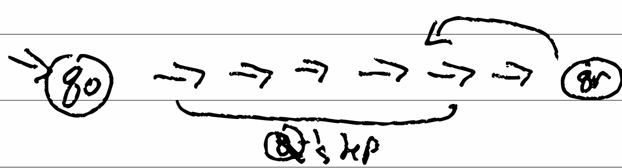
If it does have one ...

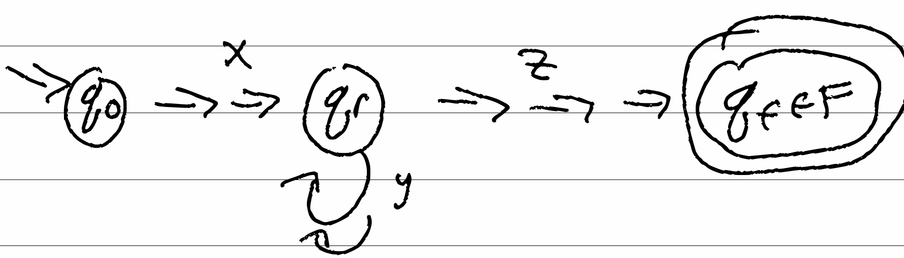
then it must be infinite (regular)

If there is an exciting loop ....

what is the shortest string to "find" it in?

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





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

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

$\neg$($\exists p \in \mathbb{N}.$ // $p = |Q|$

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

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

$\wedge\ |xy| \leq p$

$\wedge\ |y| > 0$)

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

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

$\forall d \in DFA. \exists r \in RE. L(d) = L(r)$ — Coq

$\neg RPP(B) :=$

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

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

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

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

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

8-7) Need: an infinite space problem

0^*1 $\rightarrow$ $\left\{ \begin{array}{l} \texttt{while (getc() == '0') \{\}} \\ \texttt{ungetc()} \end{array} \right.$

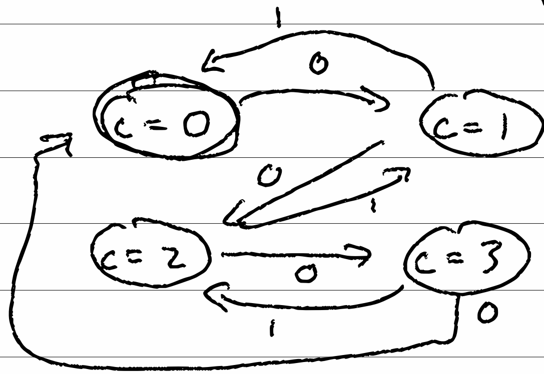
```
if (getc() != '1') {ret false}
return getc() == EOF;
```

$= 2^{32}$ uint2_t

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

$0^n 1^n \in B$

```
uint count = 0;
while (in == 0) count++
while (in == 1) count--
if (in == EOF) {
return count == 0 }
{ ret false; }
```



9-1) RPP(A) → Language = $P(\Sigma^*)$ = ALL

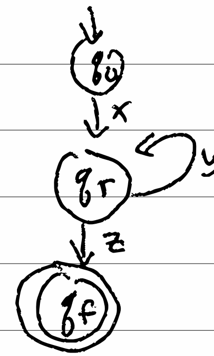
$\exists p \in \mathbb{N}.$ // $p = |Q|$

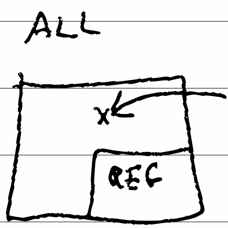
$\forall (s \in A \mid |s| > p).$

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

$\forall i \in \mathbb{N}.$ $|xy| < p$

$xy^iz \in A$ $|y| > 0)$





$\neg RPP(A) =$

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

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

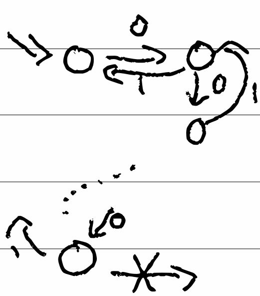
$\forall (x, y, z \in \Sigma^* \mid s = xyz \wedge |xy| < p \wedge |y| > 0)$

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

$xy^iz \notin A$

0^n1^n ie $x \in 0^n1^n$

iff $\exists n \in \mathbb{N}.$ $x = \underbrace{00 \ldots 0}_{n}\underbrace{1 \ldots 1}_{n}$



~~long~~ nat ~~int~~ count = 0

while (see 0) c++

while (see 1) c--

return count == 0

9-2) $\neg RPP(0^n 1^n) =$

$\forall p \in N.$

$\exists (s \in 0^n 1^n \mid |s| \geq p)$

choose s: $s = \cancel{\emptyset} = \varepsilon$

$s = 0^{p/2}\, 1^{p/2}$

$s = 0^p\, 1^p$ $\quad |s| = 2p \geq p$

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

$|xy| < p$ and

$|y| > 0\ )$

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

$y = 0^b$ $\quad d = p$ $\quad b > 0$

$z = 0^c\, 1^d$

$\exists i \in N, \quad xy^iz \notin 0^n 1^n$

$xy^iz = 0^a 0^{bi} 0^c 1^d = 0^{a+bi+c}\, 1^d$

$a + bi + c = d$

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

fun n:

if n == 0 : ret pz

else: let m = n-1

let pm = rec m

ps pm

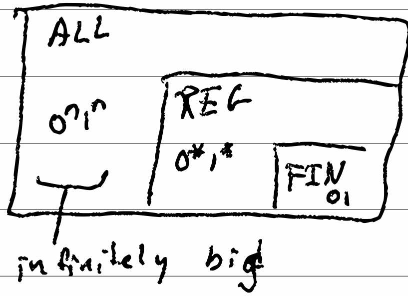
induction on N:

pz: $P\ 0$

ps: th. $P_n \rightarrow P(n+1)$

$\Rightarrow \forall n : P_n$

9-3/ There is stuff outside REG!

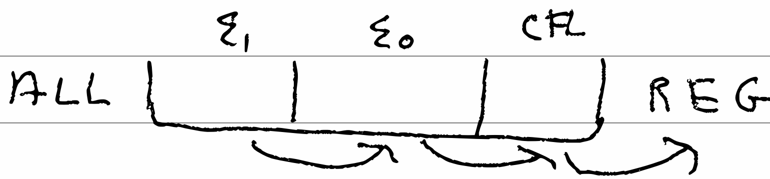


All computers are DFAs.

if $0^n1^n \notin REG$

~~then~~ and $B \in REG$

then $0^n1^n \circ B \notin REG$



$0^n1^n \notin REG$

w where $count(0,w) = count(1,w)$

ww where $w \in \Sigma^*$

$\underbrace{011}\underbrace{011} \in$

ww^R where $w \in \Sigma^*$

9-4) $0^x 1 0^y 1 0^{x+y}$

$010100 \Rightarrow$ "$1+1=2$" ✓

$00100010 00000 \Rightarrow$ "$2+3=5$" ✓

$0010010 \Rightarrow$ "$2+2=1$" X

$\forall p.$

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

$\forall xyz. \quad x = 0^a \quad y = \underbrace{0^b}_{\text{change this}} \quad z = 0^c 1 0^p 1 \underbrace{0^{2p}}_{\text{can't change}}$

$\exists i.$

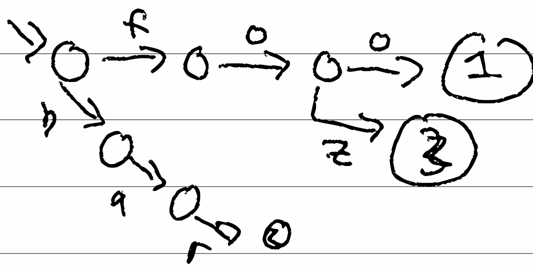
$\underbrace{2}+2=4$

$\Rightarrow \Rightarrow \quad \underbrace{8+2=10}$

12-1 / trie

foo ↦ 1
bar ↦ 2
foz ↦ 3

digital
search
tree



CFGs

$0^n 1^n \notin REG$

start variable

A $S \to \varepsilon$

B $S \to 0S1$

← rule, productions, transitions

rule = $\underbrace{V}_{} \to \underbrace{\quad}_{}$ rhs

rhs = $(V \cup \Sigma)^*$

terminal

variables
non-terminals
symbols

000111

$S \xrightarrow{B} 0S1 \xrightarrow{B} 00S11 \xrightarrow{B} 000S111$

A

"derivation"

000111

$w \in G$ iff $\exists d.\ S \to\to\to w$

12-2 $L\begin{pmatrix} S \to 01S \\ S \to SS \end{pmatrix} = \emptyset$

Context-free grammar

$0101 \in$

(V, Σ, R, S)

alphabet

finite set

$\in V$

$B \to \varepsilon$

$B \to B1N1B$

$N \to \varepsilon$

$N \to 0N$

$P(V \times (V \cup \Sigma)^*)$

$(V \to P((V \cup \Sigma)^*)$

$R = \{(S, 01S), (S, SS)\}$ $V = \{S\}$ $\Sigma = \{0, 1\}$ $S = S$

REG

DFAs

REX $(\circ, \cup, *)$

CFL

$\boxed{}$

CFG

12-3 / A $\qquad$ B

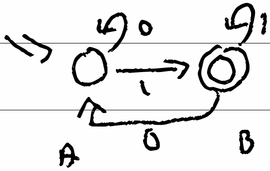
$\rightarrow$ $A \cup B$: $S \rightarrow A, S$

$S \rightarrow B, S$

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

$A \cap B$: X

DFA $\Rightarrow$ CFG



$$\begin{array}{l} \qquad\qquad\quad x \qquad\qquad y \\ A \rightarrow 0A \mid 1B \\ B \rightarrow 1B \mid 0A \mid \varepsilon \\ \qquad\qquad\quad z \qquad\qquad w \qquad v \end{array}$$

01101

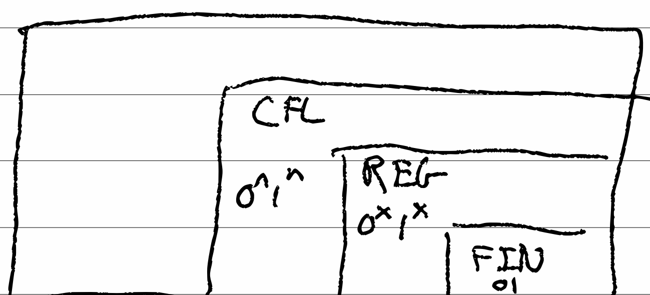
$A \xrightarrow{x} 0A \xrightarrow{y} 01B \xrightarrow{z} 011B$

$\xrightarrow{w} 0110A \xrightarrow{y} 01101B \xrightarrow{v} 01101$

$S = q_0$

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

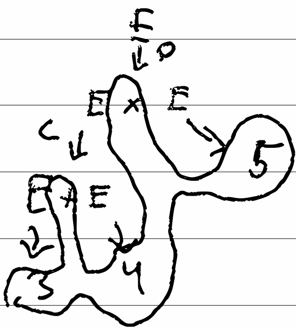
$q_f \rightarrow \varepsilon$ iff $q_f \in F$

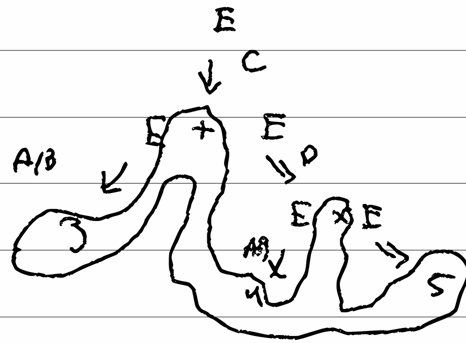


A B C D

12-4/ $E \to 0 \mid 1 \mid E + E \mid E \times E$

$3 + 4 \times 5$





fowim $E \to n$

fowin $0 = 0$

$1 = 1$

$\text{Plus}(L, R) =$ fowin L + fowin R

$\text{Mult}(L, R) =$ fowin L × fowin R

ambiguous = there are multiple trees (derivations) for the same string

if cond

12-5 Nice! amb $\rightarrow$ unamb

not possible

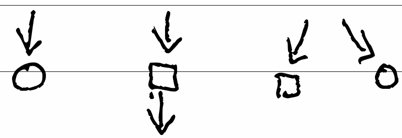
V, Σ, R, S

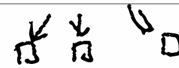
$V \rightarrow (V \cup \Sigma)^*$ $\qquad \varepsilon \qquad \Sigma\Sigma V$

$V \qquad VVVVV$

$\Sigma \qquad V \qquad V\Sigma \qquad V\Sigma$

VVV

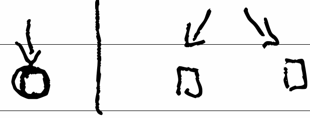




---

want: "pure" binary trees Noam

Chomsky-Normal Form



$\Sigma \qquad VV$

$V \rightarrow \Sigma$ | $A \rightarrow c$

$V \rightarrow VV$ | $A \rightarrow \underbrace{BC}_{\neq S}$

$S \rightarrow \varepsilon$

CFG $\rightarrow$ CNF

12-5/ $S \to \varepsilon \mid 0S1$

Add a new start

$R \to S$

$S \to \varepsilon \mid 0S1$

remove ε-rules

$R \to S \mid \varepsilon$

$S \to 0S1 \mid 01$

remove unit rules

$R \to 0S1 \mid 01 \mid \varepsilon$

$S \to 0S1 \mid 01$

add extra vars for >2

$R \to T1 \mid 01 \mid \varepsilon$

$S \to T1 \mid 01$

$T \to 0S$

add terminal "names"

$R \to TB \mid AB \mid \varepsilon$

$S \to TB \mid AB$

$T \to AS$

$A \to 0$

$B \to 1$

12-6/ 00111 $\in$ $0^n 1^n$

$S \Rightarrow \varepsilon$ A

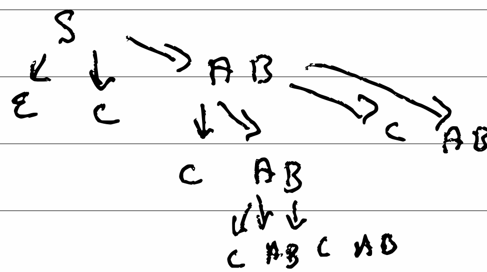
$S \Rightarrow 0S1$ B

$A \Rightarrow c$ $+1$

$A \Rightarrow BC$ $+2$

$B \rightarrow$ $+2$

$A \rightarrow$ done



$S \Rightarrow XYZ$

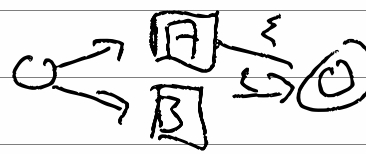
$X \Rightarrow ZZS \mid 0$

$Y \Rightarrow XYX \mid 1$

$Z \Rightarrow YSX$

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

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



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

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

unionstate = start | end | fromA qa | fromB qb

$\delta(\text{start}, \varepsilon) = \{\text{fromA } A.q_0, \text{ fromB } B.q_0\}$

$\delta(\text{fromA } q_i, c) = \text{fromA } (\delta_A(q_i, c))$

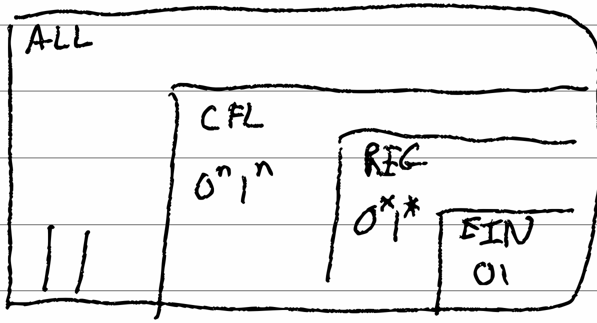
12-7/

```
new DFA( "blah",
         qi -> qi == 0 || qi == 1,
         sigma, delta,
         F )
```

⇓

```
complement (DFA d) {
  newDFA( --, qi -> ! d.F(qi) )
    ( d.Q, d.sig, d.delta,
```

$\boxed{15\text{-}1}$



REGULAR

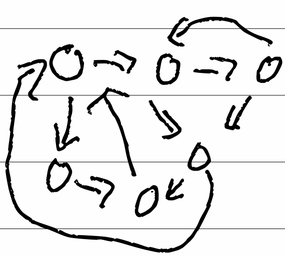
DFAs $\longleftrightarrow$ regex

$\in$ $\{\}$

CFL

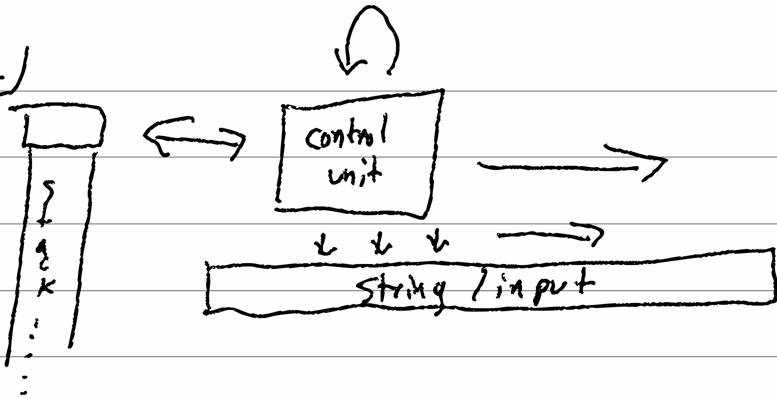
$\boxed{\text{PDA}}$ $\longleftrightarrow$ CFG

push-down automata



$\Sigma^* 1 \Sigma \Sigma$

15-2



DFA: $Q \times \Sigma \to Q$

deterministic PDA: $Q \times \Sigma \times \Gamma \to Q \times \Gamma$

PDA: $Q \times \underbrace{\Sigma_\varepsilon}_{\Sigma \cup \{\varepsilon\}} \times \Gamma_\varepsilon \to P(Q \times \Gamma_\varepsilon)$

$Q \times \Sigma \times \varepsilon \to Q \times \varepsilon$ (ignored stack)

$Q \times \Sigma \times \Gamma \to Q \times \varepsilon$ (pop)

$\times\ \varepsilon \to Q \times \Gamma$ (push)

$\times\ \Gamma \to \times\ \Gamma$ (replace)

$\varepsilon \times \Gamma \to \times\ \Gamma$

$$\frac{\delta(q_i, c) = q_j}{[q_i]\, c w \to [q_j]\, w}$$

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

$$\frac{\delta(q_i, c, \alpha) \ni (q_j, \gamma)}{\beta\alpha\, [q_i]\, c w \to \beta\gamma\, [q_j]\, w}$$

$\alpha \in \Gamma_\varepsilon \quad c \in \Sigma_\varepsilon$

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

15-3/ simulate : PDA × ~~config~~ config → config

$\mathrm{sim}\ (Q, \Sigma, \Gamma, q_0, \delta, F)\ (\overset{g}{\Gamma^*}, q_i, \overset{s}{\Sigma^*}) =$

let $\alpha : \beta \underbrace{=} g$ in

let $c : w \underbrace{=} s$ in

let $(q_j, \gamma) \leftarrow \delta(q_i, c, \alpha)$ in

$(\gamma : \beta,\ q_j,\ w)$

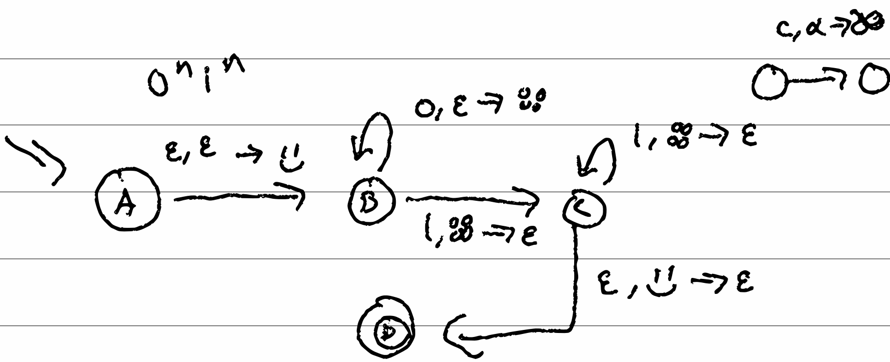
accepts : PDA × Σ^* → bool

accepts p s = while c.s ≠ ε ~~do~~ do

$c = \mathrm{sim}\ p\ c$

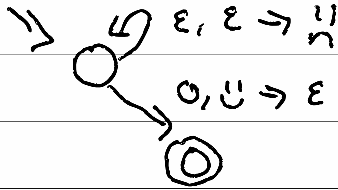
return $c.q_j \in p.F$

where $c_0 = (\varepsilon, p.q_0, s)$



ε[A]0011 → ☺[B]0011 → ☺∷[B]011 → ☺∷∷[B]11 → ☺∷[C]1 → ☺[C]ε → ε[D]ε → YES

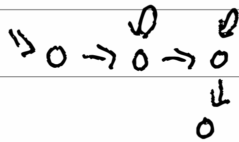
15-4/



CFG → PDA

$S \to \varepsilon$

$S \to 0S1$ ⇒



$S \Rightarrow XYZ\,011\,SYX$

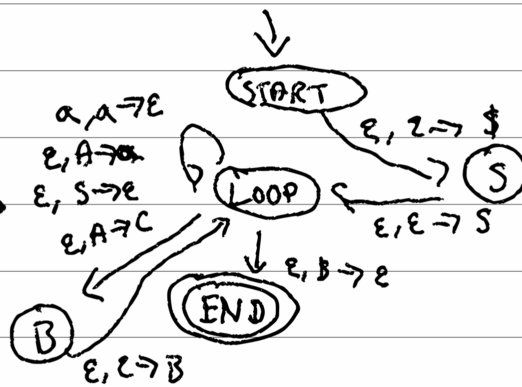
$\Gamma = \Sigma \cup V$

CNF → PDA

$A \to BC$ $(B, C \neq S)$

$S \to \varepsilon$

$A \to a$



A[LOOP] → C[B] → CB[LOOP]

ε[ST] 0011 → $[S]0011 → $S[L]0011 → $1S0[L]0011

$[L]0011

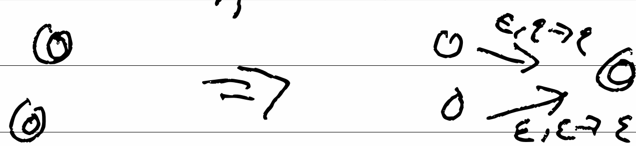
$1S[L]011 ← $11S0[L]011 ← $11S[L]11

$11[L]11 → $1[L]1 → $[L] → [END] → ✓

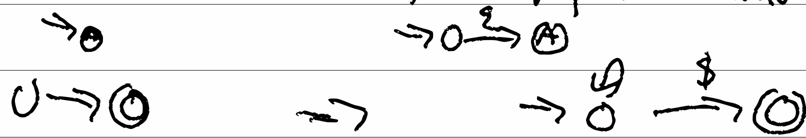
15-5/ PDAs are too complicated

so, we'll simplify with some rules

- Make a single accept



- Guarantee stack is empty on accept



- Always push or pop

push : ε , Γ

pop : Γ , ε

X ignore : ε , ε

X replace : Γ , Γ

Every symbol pushed is eventually popped

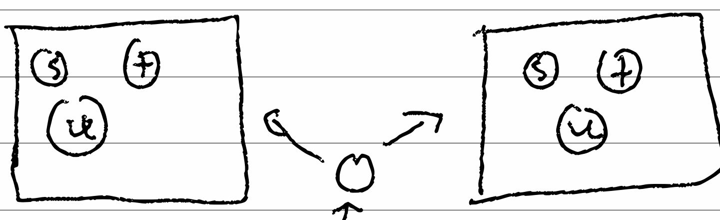
15-6/ $V = (Q \times Q)$ $\qquad S = (q_0, q_f)$

$$\frac{\begin{array}{l}(r, t) \in \delta(p, a, \varepsilon) \\ (q, \varepsilon) \in (s, b, t)\end{array}}{(p, q) \to \underset{\in \Sigma}{a} \quad \underset{\in V}{(r, s)} \quad \underset{\in \Sigma}{b}}$$

$$(p, p) \to \varepsilon$$

$$(p, q) \to (p, r)\,(r, q) \qquad \forall p, q, r$$

$A \quad \langle ast \rangle \qquad\qquad B \quad \langle bst \rangle$

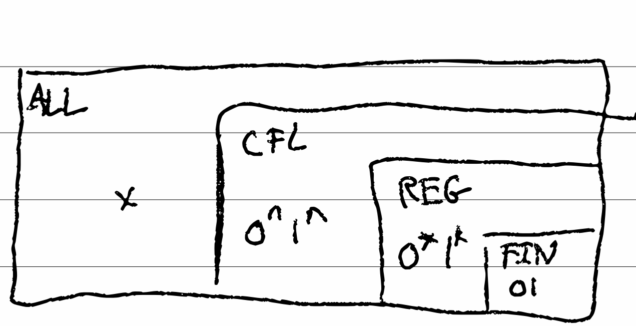


shawn $\langle ast, bst \rangle$

$A \cup B \quad <$

$$\begin{aligned} \text{shawn}\langle x, y\rangle \; = \; & \text{start} \\ \mid \; & \text{anA} \quad x \\ \mid \; & \text{aD} \quad y \end{aligned}$$

16-1



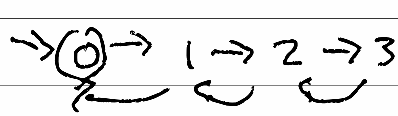
DFAs $\longleftrightarrow$ REX $\qquad$ $0^n1^n \notin$ REG

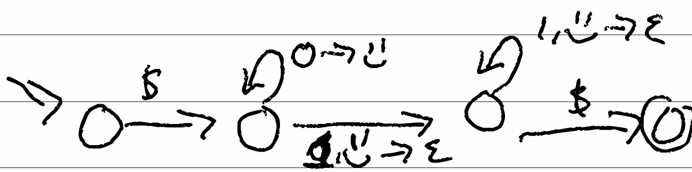
PDAs $\longleftrightarrow$ CFG $\qquad$ $x \notin$ CFL

CFL = ALL

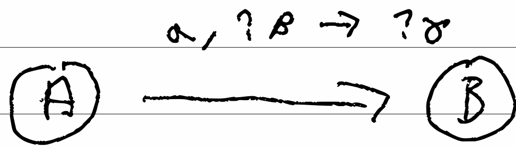
1.	P ...	$\forall r \in \text{REG}.\ P(r)$	$\forall c \in \text{CFL}.\ P(c)$
2.		$\exists x \in \text{ALL}.\ \neg P(x)$	$\exists x \in \text{ALL}.\ \neg P(x)$
3.	$\Longrightarrow$	$x \notin \text{REG}$	$\Longrightarrow x \notin \text{CFL}$

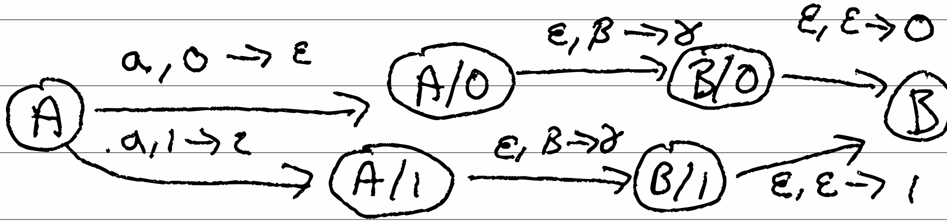
$0^n1^n \rightarrow$





16-2 Look at 2nd thing





$y\,\beta\,0\,[A]\,aw \Rightarrow y\beta\,[A/0]\,w \Rightarrow y\gamma\,[B/0]\,w$

$\Rightarrow y\gamma 0\,[B]\,w$

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

$0^n 1^x 0^y$ s.t. $x+y=2n$

$\varepsilon\,[\,]\,0^n 1^n 0^n$

☺^{2n} $[\,]\,1^n 0^n$ $\qquad$ ☺^{n} $[\,]\,1^n 0^n$

☺^{n} $[\,]\,0^n \to \varepsilon[\,]\varepsilon$ $\qquad$ $\varepsilon\,[\,]\,0^n$

$\varepsilon\,[\,]\,0^n 1^n 0^n \Rightarrow$ ☺$^n[\,]\,1^n 0^n \to$ ☺$^{n-1}$ ⚇ $[\,]\,1^{n-1} 0^n$

$\to$ ⚇^{n} $[\,]\,0^n \to$

16-3/ RPP

$\forall A \in REG,$

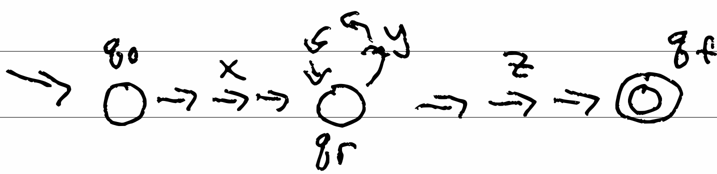
$\exists p \in N,$

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

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

$\forall i \in N,$

$xy^i z \in A.$



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

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

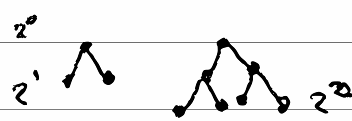
$v_i \Rightarrow v_x v_y$

$v_0 \quad v_1 \quad v_2 \quad \dots \quad v_n \Rightarrow v_r \qquad v_r \in V$

$v_1 \quad v_2\, v_2\, v_2 \quad \vdots\; v_n \quad \vdots\; v_r$

$3^2 - 1$

$2^{n+1} - 1$ $\qquad 2^0$



2^3

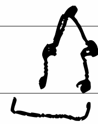
16-4/ minimum length of tree of height n

$S \Rightarrow \varepsilon$

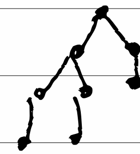
$= 0$

$S \Rightarrow a$ $= 1$

$S \to BC$



2

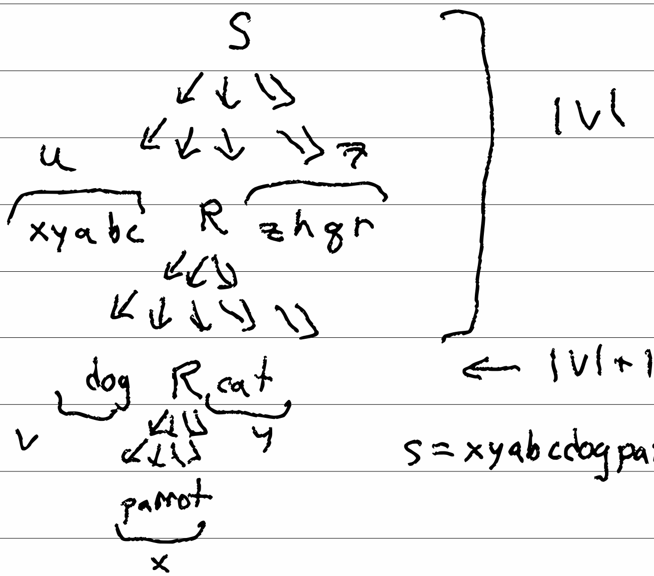


$h = 4$ $\Rightarrow$ len ≥ 3

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

$s \in A$

$|s| > 2^{|V|} + 1$



$s = $ xyabcdogparrotcatzhgr

$S \Rightarrow u R z$

$R \Rightarrow v R y$

$R \Rightarrow x$

16-5/ Context-free pumping property

$\in$: O O C $\in$

$\forall A \in CFL,$

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

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

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

$|vxy| \leq p$

$|vy| > 0 \;)$

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

$uv^ixy^iz \in A$

$S \Rightarrow uRz \qquad S \Rightarrow uRz \Rightarrow uxz \quad (i=0)$

$R \Rightarrow vRy \qquad S \Rightarrow uRz \Rightarrow uvRyz \Rightarrow uv^2Ry^2z$

$R \Rightarrow x \qquad \Rightarrow uv^2xy^2z \quad (i=2)$

$0^n1^n \in CFL \qquad p=4 \quad s = 0011 \qquad |vxy| = 2 \leq 4$

$u=0 \quad v=0 \quad x=\varepsilon \quad y=1 \quad z=1 \qquad |vy| = 2 > 0$

$uv^ixy^iz \in A\,? \qquad 00^i1^i1 = 0^{i+1}1^{i+1} \in A$

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

given p

$0^p 1^p 0^p$

given u, v, x, y, z $\qquad u = 0^{\hat{u}} \quad v = 0^{\hat{v}} \quad x = 0^{\hat{x}} \quad y = 0^{\hat{y}}$

$z = 0^{\hat{z}} 1^n 0^p$

vxy has repetition

$0^p 1^p 0^p$

vxy = 1 symbol

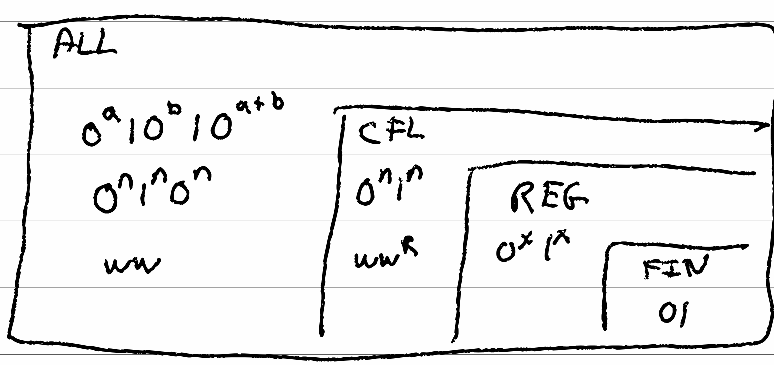
vxy = 2 symbol

3 symbols

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

$vxy = 0^a 1^p 0^b \quad | \quad 1 \leq a + p + b \leq p$

DFA union = Pair <X, Y> $\qquad$ NFA union = Start
state FromLeft X ("",X)
state From Right Y ("",Y)
END



17-1 Finite : $0\ 1$

Regular: $0^*\ 1^*$

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

& CFPP $0^n\ 1^n\ 0^n$

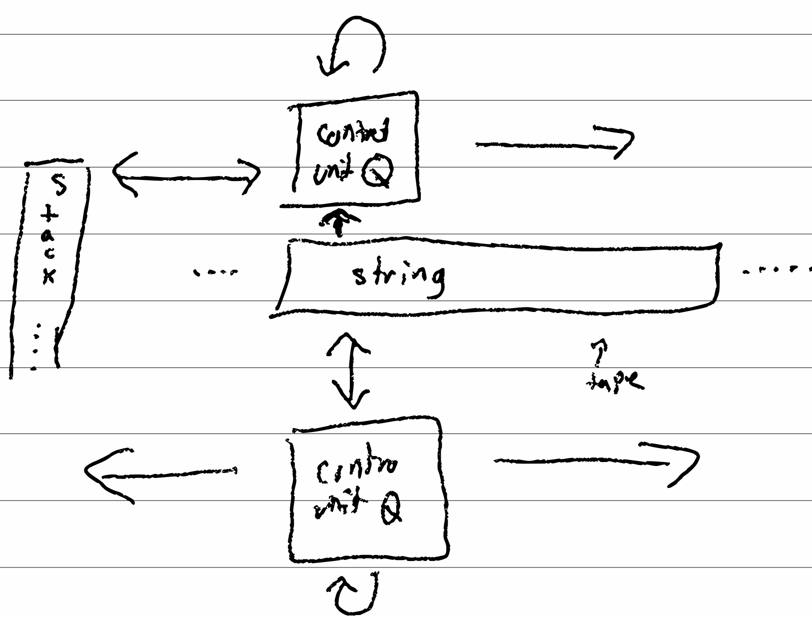
context-free pumping property

$\not\Rightarrow$ CFG, PDA

Turing Machine

Alan Turing

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



$\underline{17\text{-}2}$ $\delta : \underset{Q - \{q_a, q_r\}}{Q} \times \underset{\Gamma}{\cancel{\Sigma}} \to Q \times \Gamma \times \{L, R\}$

states : Q input : Σ $\sqcup \notin \Sigma$ tape : Γ $(\Sigma \subseteq \Gamma)$ $\sqcup \in \Gamma$

$q_0 \in Q$

$q_a \in Q$ (accepting)

$q_r \in Q$ (rejecting)

config:

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

$c_0 = \varepsilon[q_0]w$

for $w \in \Sigma^*$ input

$w \in L(M)$ iff

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

$$\frac{\delta(q_i, a) = (q_j, b, R)}{x[q_i]ay \Rightarrow xb[q_j]y} \qquad \frac{\delta(q_i, a) = (q_j, b, L)}{xc[q_i]ay \Rightarrow x[q_j]cby}$$

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

left : tape $\to$ tape

left ε = $\varepsilon : \sqcup$

left $(x : c)$ = x

right : tape $\to$ tape

right ε = $\sqcup : \varepsilon$

right $(c : y)$ = y

$\bot \leftarrow \boxed{0} \leftarrow \boxed{1} \leftarrow \boxed{1}$

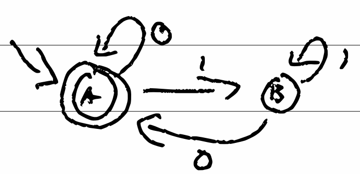
011 [q_7] 101

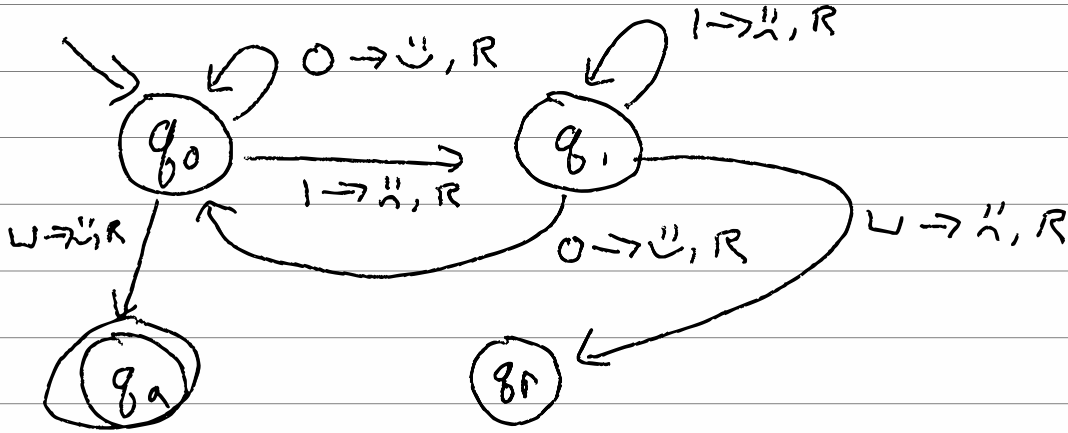
$\boxed{1} \to \boxed{0} \to \boxed{1} \to \bot$

$\boxed{}\,\boxed{q_7}\,\boxed{}$

(DFA)

17-3





$[A]110 \to [B]10 \to [B]0 \to [A]\varepsilon \to$ ✓

$\varepsilon\,[q_0]\,110 \to$ ☹ $[q_1]\,10 \to$ ☹☹ $[q_1]\,0$ ——

$\to$ ☹☹☺ $[q_0]\,\varepsilon \to \sqcup$ ☹☹☺ $[q_0]\,\sqcup \to \sqcup$ ☹☹☺☺ $[q_a]$

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

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

$\Sigma' = \Sigma \qquad \Gamma = \Sigma \cup \{\sqcup\}$

$q_0' = q_0$

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

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

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

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

17-4 / $w \# w$ where $w \in \{0,1\}^*$

I saw a zero

[] 01#01 → ⊔ []1#01 → ⊔1 []#01

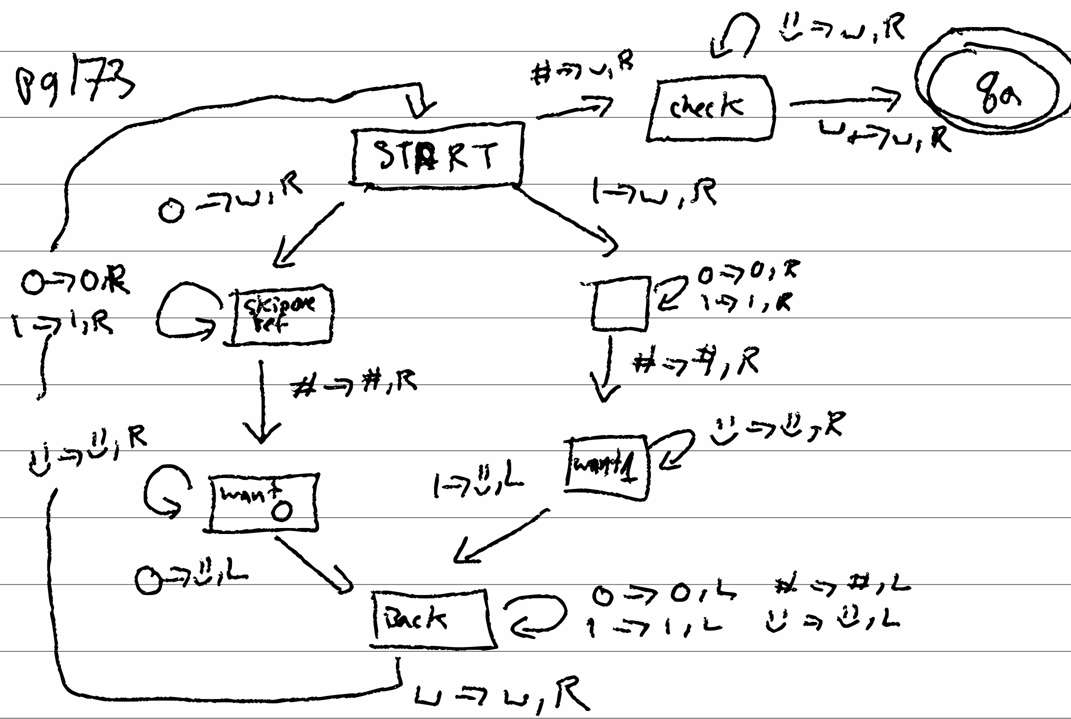
⊔# [I expect a zero] 01 → ⊔1 [I am happy] # ☺1

→ ⊔ [H] 1 #☺1 → [H] ⊔1#☺1

→ ⊔ [B] 1 #☺1 → ⊔⊔ [I saw a 1] # ☺1

⊔⊔ # [I expect a 1] ☺1   ⊔⊔#☺ [lookin 1] 1

pg 173



17-5/ A computable function

f

is a Turing-machine

and

$f(x) = y$

iff f

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

add1

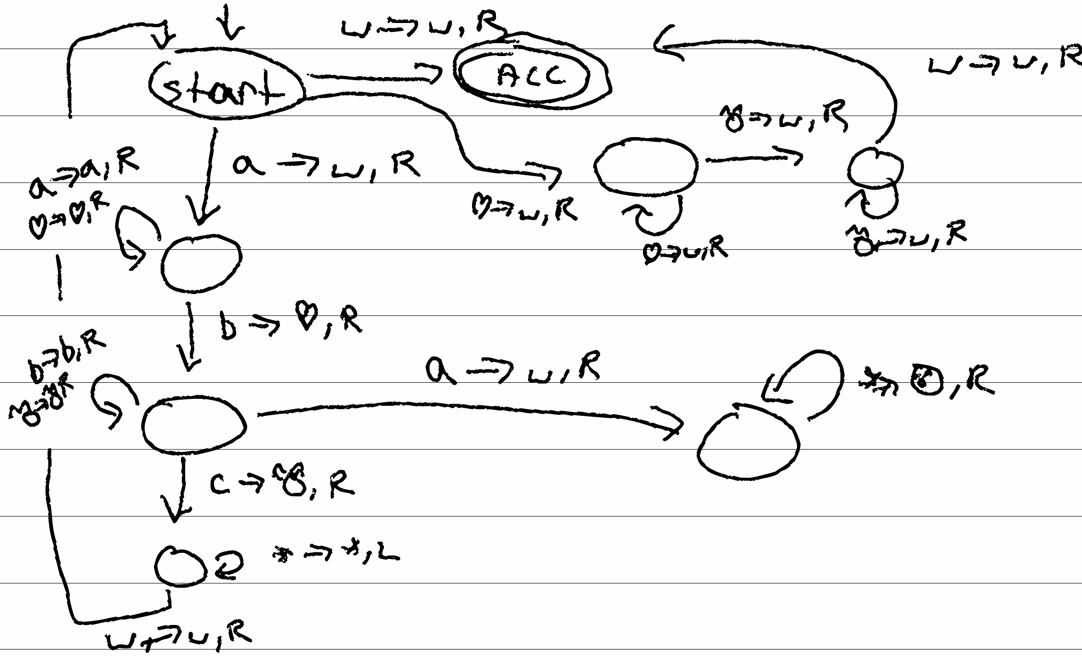
0	= 1
1	= 10
10	= 11
11	= 100

$(\supseteq \Sigma)$

18-1) $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

$Q - \{q_a, q_r\}$

$w\#w \qquad 0^n 1 0^n \qquad a^n b^n c^n$



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

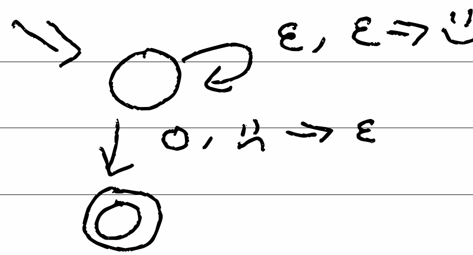
$f(\, 0^x + 0^y \,) = 0^{x+y}$

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

$|w|$

When a PDA on input w?

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



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

$\in$ (ACC) $\notin$ (REJ, LOOP, DIVERGE)

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

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

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

DIVERGING $\forall x, q_i, y.\ \varepsilon[q_0]w \Rightarrow^* x[q_i]y$ implies $\exists x', q_j, y'.\ x[q_i]y \Rightarrow x'[q_j]y'$

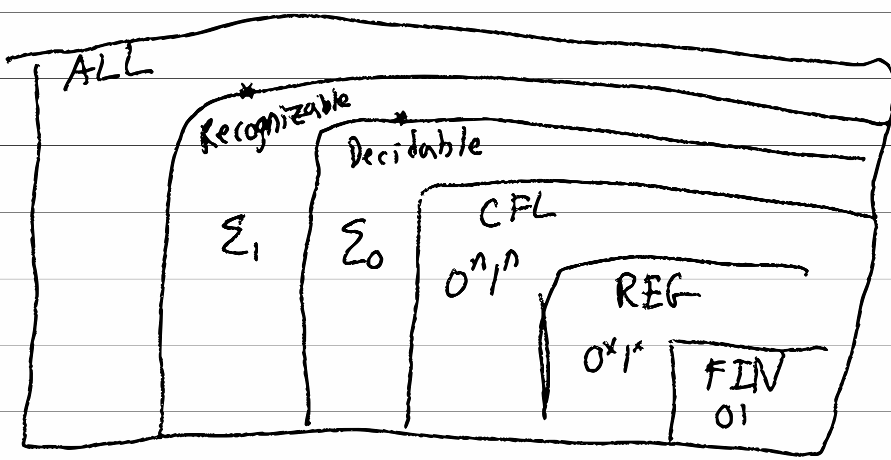
18-3/ A TM is either a

recognizer — may LOOP on some input

decider — always ACC or REJ never LOOP

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

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



Coq CoC CiC

18-4) A Turing enumerator ...

DFAs	REXs
PDAs	CFGs
TMs	enumerators

$(Q, \Sigma, \Gamma, q_0, \delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}, q_p \in Q)$

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

$w \in L(e)$ iff $\varepsilon [q_0] e \Rightarrow^* x [q_p] w$

"recognizer" enumerator $\Rightarrow$ Outback

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

18-5/ DFA union | NFA union | REG $\cup$ $\hookrightarrow$ NFA

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

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

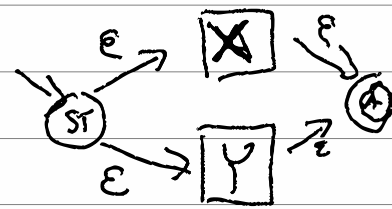
$\Rightarrow$

$(\overbrace{Q_x \times Q_y}^{\text{Pair}\langle Q_x, Q_y\rangle}, \Sigma,$

$(q_{0x}, q_{0y}),$

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

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



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

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

$Q = \{START, ACC\}$

$\cup\ \{0\} \times X$

$\cup\ \{1\} \times Y$

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

$c = \emptyset$

$\delta(ACC, \varepsilon) = \emptyset$

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

(if $q_x \in F_x$ then $\{Acc\}$ if $c = \varepsilon$)

interface NFAUnionState <X, Y>

NUS_START () $= \langle nat \to X \text{ or } Y \rangle$

NUS_ACC ()

NUS_FromX (X)

NUS_FromY (Y)

19-1 Closure properties

A set A

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

"A is closed under f"

The regular languages are closed under

$\complement$, $\cup$, $\cap$, $\circ$, $*$

The CFLs

$\cup$, $\circ$, $*$

What are Σ_0 (decidable)

Σ_1 (recognizable) languages

closed under?

Is Σ_0 closed under complement? ✓

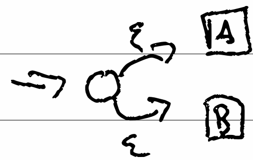
Σ_1

19-2/ Union

DFAs $Q = Q_A \times Q_B$

(did both at same time)

NFAs



TMs   – can't do NFA (deterministic)

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

$Q \to Q_A \times Q_B$ tape $\longrightarrow$ two tapes

TMs — simple	(like assembly)	DFAs
↑	↑	↑
MTMs	C	NFA
↑		
↑		
↑		
↑		
x86		

19-3/ Compiler correctness

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

input: C, assembly / NFA

output: asm, binary / DFA

bi-simulation

$$\begin{tikzcd}
A \arrow[rr, shift left, "\text{compile}"] \arrow[d, "\text{eval}A"'] & & B \arrow[ll, shift left] \arrow[d, shift right, "\text{eval}B"'] \\
A' \arrow[u, bend left=60] \arrow[rr, shift right] & & B'' \arrow[ll, shift right, "\text{decompile}"'] \arrow[u, shift right, bend right=60, "\text{uneval}B"']
\end{tikzcd}$$

$$\begin{tikzcd}
A \arrow[rr] \arrow[dr, "M"'] & & B \arrow[dl, "M"] \\
& M &
\end{tikzcd}$$

Galois connection

19-4) TM w/ "stay"

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

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

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

$\delta(q_i, a) = (q_j, b, S)$
$\Rightarrow$
$(q_K, c, L/R)$
when $\delta(q_j, b) = ?$

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

$u[q_i]av \Rightarrow$

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

$\forall x \in \Gamma$

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

$u[q_j]bv$

19-5/ MTMs (exist to do u/n)

$$\delta_A : Q_A \times \Gamma_A \to Q_A \times \Gamma_A \times \{L, R\}$$
$$\delta_B : Q_B \times \Gamma_B \to Q_B \times \Gamma_B \times \{L, R\}$$

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

$\delta_{A\cup B}((q_a, q_b), (t_a, t_b)) =$
let $(q_a', t_a', d_a') = \delta_A(q_a, t_a)$
$(q_b', t_b', d_b') = \delta_B(q_b, t_b)$ in
$((q_a', q_b'), (t_a', d_a'), (t_b', d_b'))$ if $q_a' = ACC$
then ACC

Multi-tape Turing Machine

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

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

$$\begin{matrix} uc \\ x \end{matrix} \begin{bmatrix} q_i \end{bmatrix} \begin{matrix} av \\ \alpha y \end{matrix} \Rightarrow \begin{matrix} u \\ x\beta \end{matrix} \begin{bmatrix} q_j \end{bmatrix} \begin{matrix} cbv \\ y \end{matrix}$$

19-6

$$\text{MTM config} \xrightarrow{\text{compile}} \text{STM config}$$

eval step $\Rightarrow$ $\downarrow$ $\qquad \Gamma = \{\#\} \cup \Gamma_A \cup (\Gamma_A \times \{X\}) \cup \Gamma_B \cup (\Gamma_B \times \{X\})$ $\qquad \downarrow$

new config $\qquad$ new S config

$$\begin{matrix} u\,c \\ x \end{matrix}\begin{bmatrix} q_i \end{bmatrix}\begin{matrix} a v \\ \alpha y \end{matrix} \longrightarrow [q_i]\, u c \overset{\times}{a} v \# x \overset{\times}{\alpha} y$$

(q_i, a) $\qquad (q_i, a, \alpha)$

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

$\Downarrow$ $\qquad (q_j)$

$$\begin{matrix} u \\ x\,\beta \end{matrix}\begin{bmatrix} q_j \end{bmatrix}\begin{matrix} c\, b v \\ \gamma y \end{matrix} \longleftarrow [q_j]\, u \overset{\times}{c} b v \# x \beta \overset{\times}{\gamma} y$$

Σ_0 is closed under $\cup$ and $\cap$ ✓

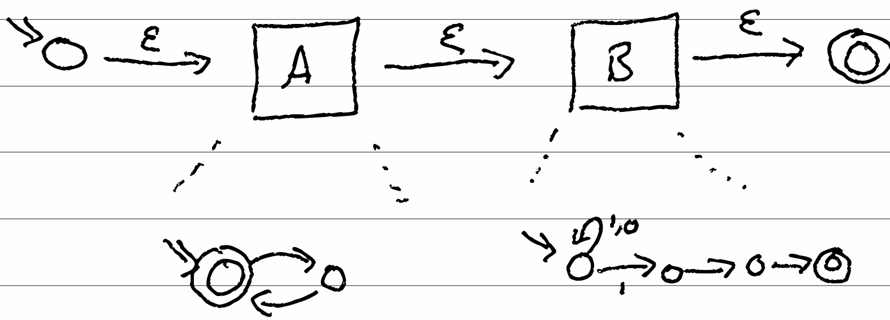
Σ_1 ✓

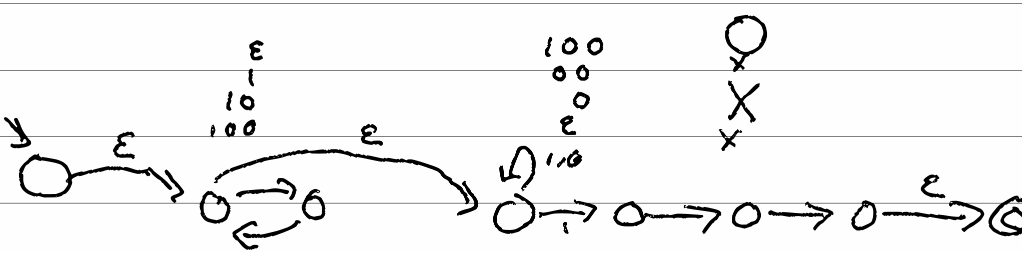
2041) Σ_0 : $C, \cap, \cup, \circ, *$

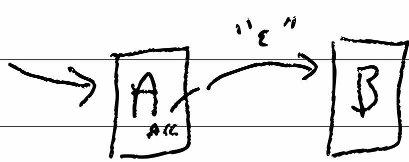
Σ_1 : $\square, \cap, \cup,$

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

$f(g(x))$ is comp







$w\#w \circ 0^n 1 0^m 1 0^{n+m}$

$\boxed{01\#01}\boxed{00101000}$

ε 0 11# 011 001 011#01 00101000

01000

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

N for lang B

Then

tape 0: input (x)

tape 1: u

tape 2: v

tape 3: sim tape

concat (M, N) := Given input x,

split input into $x = uv$ in all possible ways (nondeterministically)

run $M(u)$ (if REJ, then try next)

run $N(v)$ (if REJ, then try next)

ret ACC if both ACC

ret REJ

e	abc	– REJ
a	bc	– M.LOOPS
ab	c	– N.LOOPS
abc	ε	– ACC

Σ_0 : $0 \rightarrow \checkmark$

Σ_1 : $0 \rightarrow \checkmark$

20-3/ Non-deterministic Turing Machine

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

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

$$\text{2-choices} : Q \times \Gamma \Rightarrow \underbrace{(Q \times Q)}_{\text{fork}()} + (Q \times \Gamma \times \{L, R\})$$

(L)

(W)

$$\frac{\delta(q_i, a) = (q_j, b, R)}{x[q_i]ay \Rightarrow xb[q_j]y} \quad (R)$$

$\Rightarrow$: rel(config)

$P(\text{config}, \text{config})$

$=$: rel(num-sentences)

$$\frac{}{0 = 0} \qquad \frac{x = y}{a + x = a + y}$$

$$\frac{x = y \quad a = b}{ax = by} \qquad \frac{\cancel{x=y} \quad a = b}{a(x+y) = bx + by}$$

$$\frac{\delta(q_i, a) = (q_j, q_k)}{x[q_i]ay \Rightarrow x[q_j]ay} \quad (0\text{-ch})$$

$$\frac{\delta(q_i, a) = (q_j, q_k)}{x[q_i]ay \Rightarrow x[q_k]ay} \quad (1\text{-ch})$$

"oracle"-model

reject if $\Rightarrow^* [\,]$

20-4 Forking semantics

F-config = Sequence of config

acc if $\Rightarrow^* [\text{"}x[q_a]y\text{"}, c_2 \ldots c_n]$

for input w $\quad$ $F\text{-config}_0 = [\text{"}\varepsilon[q_0]w\text{"}]$

$$\frac{}{[\text{"}x[q_r]y\text{"}, c_2 \ldots c_n] \Rightarrow [c_2 \ldots c_n]}$$

$$\frac{\delta(q_i, a) = (q_j, b, R)}{[\text{"}x[q_i]ay\text{"}, c_2 \ldots c_n] \Rightarrow [c_2 \ldots c_n, \text{"}xb[q_j]y\text{"}]} \; R$$

$$\frac{\delta(q_i, a) = (q_j, q_k)}{[\text{"}x[q_i]ay\text{"}, c_2, \ldots, c_n] \Rightarrow [c_2, \ldots, c_n, \text{"}x[q_j]ay\text{"}, \text{"}x[q_k]ay\text{"}]}$$

$[c_1, \ldots c_n]$

$\hat{c}_1 \# \hat{c}_2 \# \ldots \# \hat{c}_n$

$\hat{\ }$: $x[q_i]y \Rightarrow x q_i y$, $\nearrow Q$

$\Gamma' = \Gamma \cup Q$

$F\text{config}_0 \longrightarrow \text{config} \downarrow \downarrow \downarrow \downarrow \text{config} \longrightarrow F\text{config}_1$

$F\text{config}_0 \xrightarrow{\text{run}} F\text{config}_1$

$0 \to \text{ND} \to 2\text{ND} \to \text{BT}$

$2\text{ND} \to \text{Forking} \to \checkmark$

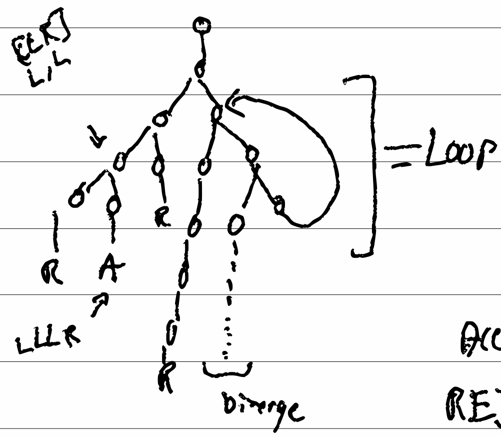
20-5

ACC — $\delta(q_i, a) = (q_{ACC}, \sqcup, \sqcup)$

REJ — $\delta(q_i, a) = (q_{REJ}, \sqcup, \sqcup)$

— $\delta(q_i, a) = (q_j, b, L/R)$

— $\delta(q_i, a) = (q_j, q_k)$



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

explore in lexicographic order

ACC if some map leads to ACC

REJ if entire "level" (length) is REJ

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

$k 2^k$

tape 0: w

tape 1: current map $\rightarrow$ add 1

tape 2: simulation tape

20-6

$$
\begin{tikzcd}
& & & 0 \arrow[d] & \emptyset \arrow[l] \\
& & & ND \arrow[d] & \\
& & & 2ND \arrow[dl] \arrow[dr] & \\
\cup \arrow[drr] & \cap \arrow[dr] & \text{fork} \arrow[d] & & BT \arrow[dll] \\
& & MT \arrow[d] & C \arrow[dl] & \\
& & TM & &
\end{tikzcd}
$$

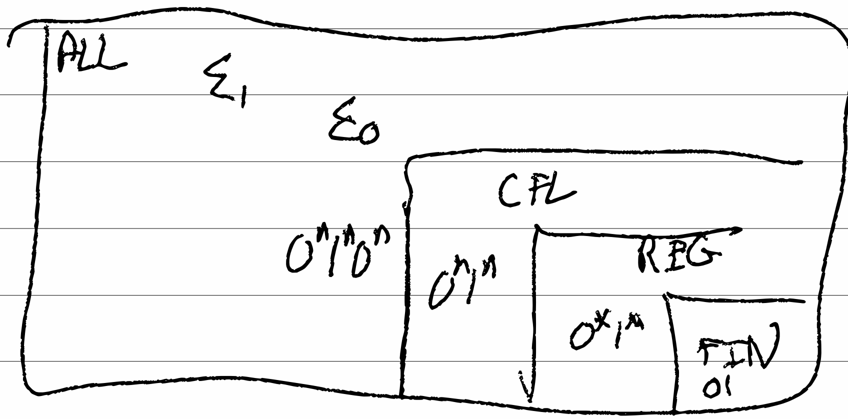
$w \in A^*$ iff

$w = w_1 \dots w_n$

where $w_i \in A$

21-1) Σ_0 (decidable - Y/N)   c, $\cup$, $\cap$, $\circ$, $*$

Σ_1 (recognizable - Y/N/L) $\square$ $\cup$, $\cap$, $\circ$, $*$



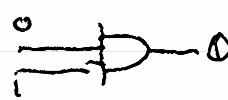
Church-Turing Thesis

"Algorithm means Turing Machine or λ-calculus term"

Human = $Math_1$ = $Math_2$

TM ⇄ λ

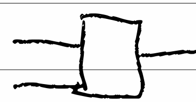
0 ← (0,1)   (+1,-1)



trad   qbit

(0.5, 0.5)

(0.3, 0.7)



(0.4, 0.6)

21-2/

Humans counting means $N(0, +1)$

Love means Okcupid / Tinder Match.com

Movie taste means Netflix advice

Algorith = TM $\rightleftarrows$ λ-calculus

TM
$\updownarrow$
ND
$\updownarrow$
Stack
$\updownarrow$
Reg
$\updownarrow$
x86
$\updownarrow$
Jay's phone
"Hardware"

λ-calculus
$\updownarrow$
ISWIM
$\updownarrow$
Algol
$\updownarrow$
C
$\downarrow$
Java
$\downarrow$
Racket $\leftrightarrow$ "Software"

4-3 10th: "Devise an effective method to find the integer roots of any polynomial of any number of variables"

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

$dx^3 + \ldots$

$ex^4 + \ldots$

"one variable"

$k = 2$

$c_0 = c \quad c_1 = b$

$c_2 = a$

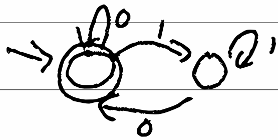
$$\sum_{i=0}^{k} c_i x^i = 0$$

try $x = 0, +1, -1, +2, -2, +3, -3$

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

21-4) $\Sigma_0 \neq \Sigma_1$

$w = \varepsilon$



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

$\{\}, \{\}, \varepsilon, \{\}, \{\}$

010110 # {0}

$\langle \quad \rangle \rightarrow$ 011111000100110

$DFA(REG) \subseteq TM(\Sigma_0)$

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

$$\delta'(q_i, \sqcup) = \left(\begin{array}{ll} acc & \text{if } q_i \in F \\ rej & \text{o.w.} \end{array}, \sqcup, R\right)$$

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

A_{NFA}

A_{CFL}

$A_{TM} \ni \langle M, w\rangle$ st M is a TM and $w \in L(M)$

Universal Turing Machine

21-5/ $A_{TM} \in \Sigma_0$ or $\in \Sigma_1$?

look at code

look for loops

none $\rightarrow \Sigma_0$ yes $\Rightarrow \Sigma_1$

A_{TM} (< M runs forever , 0110 >)

= should = Reject (Σ_0)

reject or loop (Σ_1)

$\boxed{\begin{array}{c} A_{TM} \in \Sigma_1 \\ A_{TM} \notin \Sigma_0 \\ \Sigma_1 \neq \Sigma_0 \end{array}}$

pretend $\in \Sigma_0$

Make a new TM D(<M>) =

" Run A_{TM} on <M, <M>>.

Output the opposite of A_{TM}."

D<M> = accept if M does not accept <M>

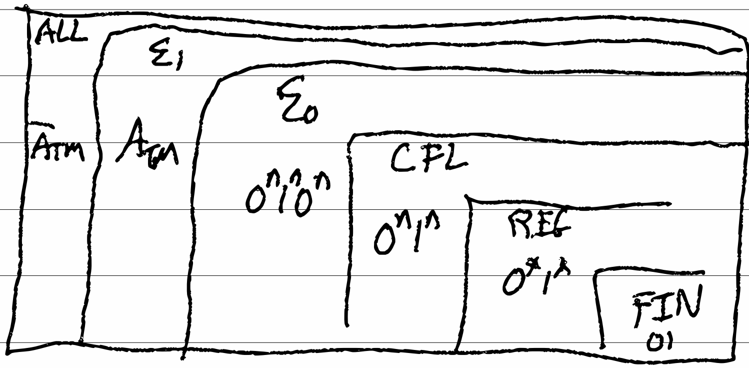
reject if M does accept <M>

Run D on <D> = accept if D d.n.a <D>

reject if D d.a. <D>

Liar's Paradox Quine's Paradox

22-1



$A \in \Sigma_0$ iff $A \in \Sigma_1 \wedge \overline{A} \in \Sigma_1$

$\Rightarrow$: trivial

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

$M_Y(x) = Y$ if $x \in A$ $\qquad$ $M_N(x) = Y$ if $x \notin A$

NorLoops o.w. $\qquad$ NorLoops o.w.

$M(x) = Y$ if $x \in A$

N if $x \notin A$

M runs M_Y and M_N simultaneously

and uses the first answer

know: $A_{TM} \notin \Sigma_0$ $\longrightarrow$ know: $B \in \Sigma_0$ iff $B \in \Sigma_1 \wedge \overline{B} \in \Sigma_1$

know: $A_{TM} \in \Sigma_1$ $\qquad$ $\neg(A_{TM} \in \Sigma_1 \wedge \overline{A_{TM}} \in \Sigma_1)$

$\Rightarrow A_{TM} \notin \Sigma_1 \vee \underline{\overline{A_{TM}} \notin \Sigma_1}$

2222/ $A_{TM} \ni \langle M, w \rangle$ iff M accept w

$\overline{A_{TM}} \ni x$ iff $x \neq \langle M, w \rangle$

or $x = \langle M, w \rangle$ and

M $\underbrace{\text{does not accept}}_{\text{not "rejects"}}$ w

$\overline{A_{TM}}$ must predict when a TM will loop/diverge

predict if TM M "halts"

$\overline{A_{TM}}$ is the "Halting Problem"

is unrecognizable

22-3

```
bool x = read()
obj; o = NULL;
if (x) {
    o = new Cat(); }
else {
    o = new Dog(); }
.... (ignore o or x)
if (x) {
    o.purr(); }
else {
    o.bark(); }
```

Coq

Typed Racket (occurrence typing → TypeScript, Hack)

$\widehat{ATM}$ o $0^x 1^*$

22-4 Sizes of infinity

$\aleph_0$

$x_1 \ldots x_{\aleph_0}$

N	Z	Q	R		$\boxed{X}$
0, 1, 2	−2, +3	5/4, 8/1711	π, e, $0.\overline{3}$		∞

→ $R^{\infty} =$

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

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

$|0^* 1^*|$

Peano

$N = 0 \mid S\ N$

$\emptyset \quad P$

$f : X \Rightarrow Y$

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

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

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

one-to-one

$\forall a, b \in X.\ f(a) = f(b) \Rightarrow a = b$

onto:

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

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

samesize(A, B) := $\exists f \in A \to B.\ \text{o2o}(f) \wedge \text{onto}(f)$

$\Rightarrow$

$|A| = |B|$

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

$|N|$

$=$

$|\{\text{even numbers}\}|$

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

$f(a) = 2 \times a$

$2 \times a = 2 \times b$

$\Rightarrow a = b$

$\forall e. \exists n.\ 2 \times n = \underbrace{e}_{2 \times m}$

$|N| = \aleph_0$ (Georg Cantor)

"countable"

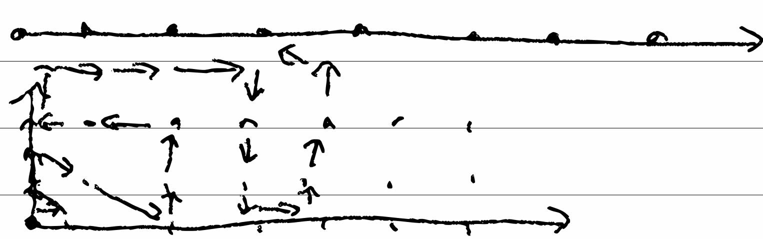
N

$=$

$N \times N$

$=$

N^2



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

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

22-6

$|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$

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

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

find $g : \underbrace{\mathbb{N} \times \mathbb{N} \times \mathbb{N}}_{k+1} \to \mathbb{N}$ ex $f : \underbrace{\mathbb{N} \times \cdots \mathbb{N}}_{k} \to \mathbb{N}$

$g(\text{before} \ldots, \text{last}) =$
$\text{cantor}(f(\text{before}), \text{last})$

sort: $(A \times A \to \text{Bool}) \times \text{List}(A) \to \text{List}(A)$
property-based testing

$f : \mathbb{N} \to \text{List}(A)$ f one onto (bijection)
fuzz / enumerative testing

Fair Enumeration

22-7) $|\Sigma_0|$ and $|\Sigma_1|$ and $|$the TMs$|$ $= |\mathbb{N}|$

$$|\Sigma_0| \leq |\Sigma_1| \leq |\text{TMs}| \leq |\mathbb{N}|$$

$$\text{TM} = Q \times \Sigma \times \Gamma \times q_0 \times (\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}) \times q_a \times q_r$$

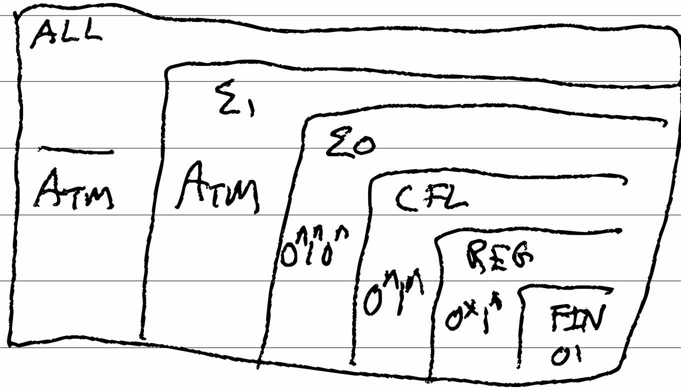
$$= (\{k\} \mathbb{N}) \times (\{g\} \mathbb{N}) \times \{0, \ldots, k\} \times (\{0 \ldots k\} \times \{0 \ldots g\} \times \{0 \ldots k\} \times \{0 \ldots g\} \times \{L, R\})$$

$$\times \{0 \ldots k\} \times \{0 \ldots k\}$$

$$\leq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

$$\simeq \mathbb{N}$$

23-1



$$\mathbb{N} \simeq \mathbb{N}\times\mathbb{N} \simeq \mathbb{N}^k \cong TM > \Sigma_1 \geq \Sigma_0$$

$<$

~~[illegible]~~ $\mathbb{R} \geq$ IBS $\overset{?}{\simeq}$ ALL

Real numbers, $\mathbb{R}$, "numbers with decimals"

$1.2357 \qquad \dfrac{12357}{10000} \in \mathbb{Q}$

"weird numbers, like pi"

$\pi \quad e \quad \sqrt{2}$ "have infinitely many digits"

vector space of bases

$i = (1,0,0) \qquad u = (1,1,0)$

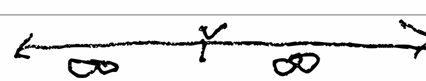
$j = (0,1,0) \qquad v = (-1,0,3)$

$k = (0,0,1) \qquad w = (-1,-1,0)$

Cauchy sequence

$\mathbb{Q}, \mathbb{Q}, \mathbb{Q}, \mathbb{Q}, \dots, \mathbb{R}$

Dedekind cut



23-2

"infinitely many digits"

$1/3$ "always 3" $0.\overline{3}$

$0.1\overline{25}$ $\frac{62}{495}$

$\mathbb{R}$ .... $\mathbb{R}_{[0,1)}$

$\mathbb{N} \longrightarrow$ digit $\{0, 1, 2, , \}$

place after zero — which digit

$0.\overline{3}$ = x: return '3'

$0.\overline{12}$ = x: if even? x then: 1
else: 2

0.133 = x: case x of
$0 \to 1$ $2 \to 3$
$3 \to 3$ def $\to 0$

Infinite Binary Sequences = IBS = $\mathbb{R}_{[0,1)}$

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

23-3 Is $N \to \{0,1\}$ countable?

$(= |N|)$

onto:

$\forall x, y.$

$f\ x = f y$

$\to x = y$

one-to-one:

$\forall x \in IBS\ \exists y \in N$

$f\ y = x$

$\exists f : IBS \to N.$

onto and one-to-one

$\exists f : N \to IBS$

onto-and onetoone

claim:

$\neg(\exists f : N \to IBS.$

$\forall x \in IBS.$

$\exists y \in N.$

$f\ y = x)$

$\leftrightarrow$

$\forall f : N \to IBS (= N \to \{0,1\})$

$\exists x \in IBS (= N \to \{0,1\})$

$\forall y \in N.$

$f\ y \neq x$

$\leftrightarrow$

i	f(i)
0	0,1011001...
1	0,0110111...
2	0,0111110...
3	0,1111000..
4	0,10101100...

0.00000....

$\forall f : N \to N \to B.$

$\exists x : N \to B.$

$\forall y \in N.$

$\exists i \in N.\ f(y)(i) \neq x(i)$ $i = y$ $f(y)(y) \neq x(y)$

$\neg f(y)(y)$

$\boxed{\text{Cantor's Digonalization Proof}}$

$x(a) = \neg f(a)(a)$

✓

23-4 | IBS is the same size as ALL

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

$ALL = P(\Sigma^*)$

$\quad\quad\quad\quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8$

$= P(\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\})$

$= \{ \underset{\{0,1\}}{\overset{\{\}}{\emptyset}}, \overset{\{0\}}{\{\varepsilon\}}, \overset{\{1\}}{\{0\}}, \{1\}, \ldots$

$\{\varepsilon, 0\}, \{\varepsilon, 011001\}, \ldots$

$\{0, 2, 4, 6, 8, \ldots\}$

$\{\varepsilon, 1, 01, 11, 001, \ldots\}$

$\{\varepsilon, 00, 01, 10, 11, 0000, \ldots\} \quad \ldots \}$

$N \to B \ (= IBS)$

$x \in IBS \qquad x(i) = \ldots$

$y \in ALL \qquad y = \{\varepsilon, 0, \ldots\}$

$ALL \to IBS \ (= N \to B)$

$f : \cancel{IBS \to ALL} \ \wedge \ onto(f) \ \wedge \ one\text{-}to\text{-}one(f)$

$f(A)(i) = lexi(i) \in A$

$ALL \approx IBS$

<u>23-5</u> ✓

	IBS	$>$	$\mathbb{N}$		
	$\geq$		$=$		
	$\mathbb{R}[0,1)$		$\mathbb{N}\times\mathbb{N}$		
	$<$		$=$		
	$\mathbb{R}$		$\mathbb{N}^k$		
	$=$		$\geq$		
	ALL		TM		
	$\aleph_1$		$\geq$		
			Σ_1	$=$	$\aleph_0$
			$\geq$		
			Σ_0		

23-6 Reducibility

A is reducible to B ($A \leq_m B$) if
$\exists f$ (a computable function)
$\forall w.\quad w \in A \text{ iff } f(w) \in B$

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

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

$$\boxed{\forall B.\ A_{TM} \leq_m B \Rightarrow B \notin \Sigma_0}$$

$\forall B.\ \overline{A_{TM}} \leq_m B \Rightarrow B \notin \Sigma_1$

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

$A_{TM} \leq_m E_{TM}$ $\quad \exists f.\ \langle M, w \rangle \rightarrow \langle M' \rangle$

M accepts w iff $L(M') = \emptyset$

$M'_{w}(x) =$ run M on w if yes $\rightarrow$ reject x
o.w. accept

23-7) $REG_{TM} \ni \langle M \rangle$ iff $L(M) \in REG$

$f: \langle M, w \rangle \longrightarrow \langle M' \rangle$

M accepts w iff $L(M') \in REG$

$M'(x) =$ if M accepts w then ret YES

o.w. check if $x \in 0^n 1^n$

$EQ_{TM} \ni \langle M_1, M_2 \rangle$ iff $L(M_1) = L(M_2)$

$E_{TM} \leq_m EQ_{TM}$

$f: \langle M \rangle \to \langle M_1, M_2 \rangle$

$L(M) = \emptyset$ iff $L(M_1) = L(M_2)$

$M_1 = M$ $\quad M_2(x) =$ reject