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 ↑ ?

"Is there a god?"

"Will Buffy be remake?"

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

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

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

$$A \overset{\swarrow \text{subset}}{\subseteq} B \quad \text{iff} \quad \forall x.\ x \in A \Rightarrow x \in B$$

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

$\Sigma = \{\underset{\text{chars}}{0}, \underset{\text{chars}}{1}\}$ $\{0, 1, \sqcup, \$, +\}$

$\underbrace{\text{"}\overset{0}{0}\overset{1}{1}\overset{2}{0}0001\text{"}}_{\text{length} = 7}$ = a string = s

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())));
```

{Σ}

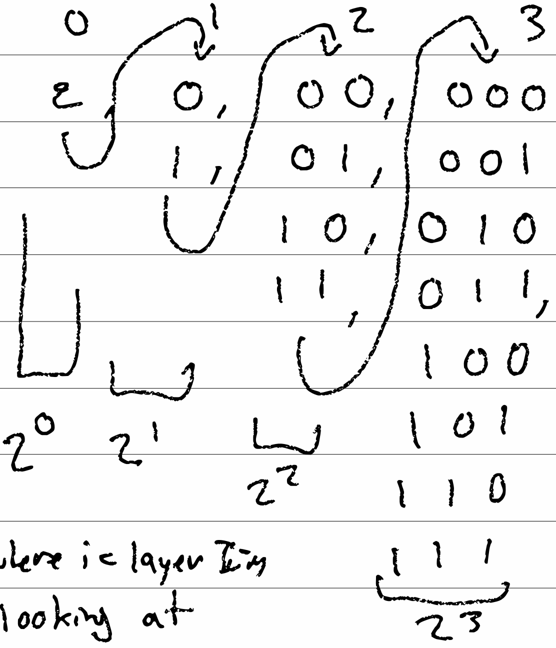
1-6/ Every alphabet has a lexicographical ordering of the strings in Σ^*

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

" $= B$

new List<Chars>

{ Zero, One};



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

lexi : $\Sigma \times N \rightarrow \Sigma^*$

lexi B 0 = ε

lexi B 1 = 0

lexi B 2 = 1

lexi B 6 = 1~~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^*

$\{0, 1\}$ $\{ \varepsilon,$ 0110, 000001,

$\}$

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

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

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

$\emptyset \in P(A)$ $\emptyset \subseteq A$

$\{0\} \in \nearrow$ $\{2, 3\} \in$ $\{0, 2, 3\}$

0110 $\{1, 2\}$

$\downarrow$

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

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

. . . .

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

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

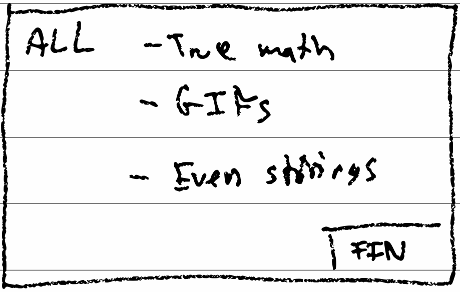
all even lengthed strings $\in P(\Sigma^*)$ $= \{\varepsilon, 00, 11, 01, \}$

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



$\emptyset \in$ FIN

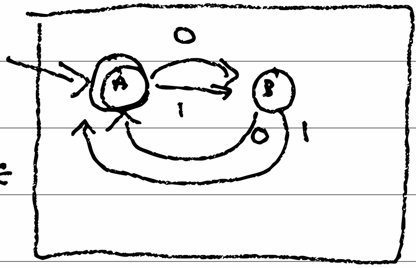
$\forall x \in \Sigma^*.\ \{x\} \in$ FIN

$A \in$ FIN $\wedge$ $B \in$ FIN

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

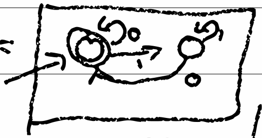
All even strings =

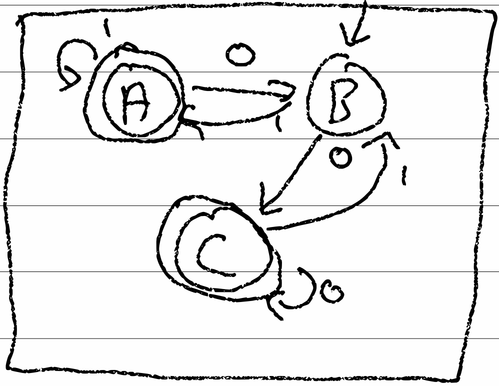
$x \in$



ε 01 11

even nums =





DFA - a deterministic finite automata

○ - states $\{A, B\}$

→☐ - start state A

◎ - accepting states $\{A\}$

$\circ \xrightarrow{x} \circ$ - 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

2-3/ $x \in DFA\,(\overset{\text{states}}{Q},\ \overset{\text{alphabet}}{\Sigma},\ \overset{\text{start}}{q_0 \in Q},\ \overset{\text{accepting}}{F \subseteq Q},$

$\delta : Q \times \Sigma \Rightarrow Q$ – transitions)

DFA configuration $= Q \times \Sigma^*$

$[\hat{q}]\,w^{\nearrow}$

config update function : config × DFA ⇒ 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 $q_f \in F$

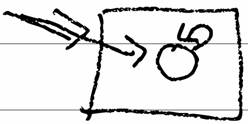
$0110 \in EvenLen$ iff $[A]0110 \Rightarrow [B]110 \Rightarrow [A]10$

$\Rightarrow [B]0 \longrightarrow [A]\,\varepsilon \quad A \in \{A\}$

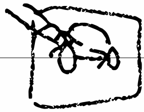
✓

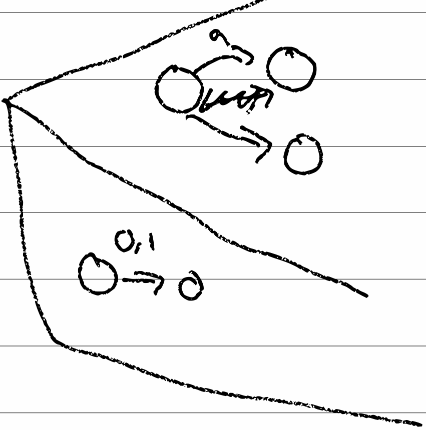
```
class DFA {
  ... Q, Σ, F, q0, δ ...
  public bool accepts (String x) {
    State qi = q0;
    while (!x.[illegible] empty()) {
      qi = δ(qi, x.first());
      x = x.rest(); }
    return F.in(qi); } }
```

2-4/

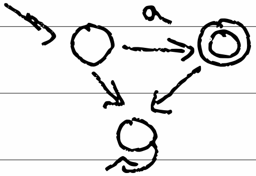


= ∅ =



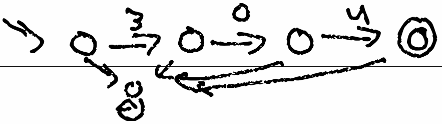


= {ε}



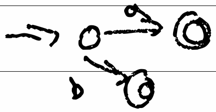
= {a}

= 

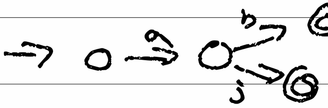
{304} = 

A ∪ B ∈ DFA (if A ∈ DFA and B ∈ DFA)

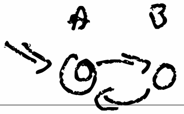
{a} ∪ {b}



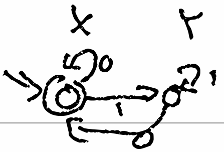
{a;} ∪ {ab}



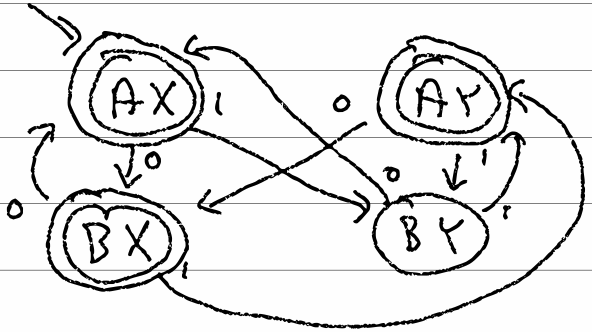
2-5] EvenLen



IsEven



$\cup$



ε ✓
00 ✓
11 ✓
0 ✓
110 ✓

$\cap$

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

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

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

$X = A \cup B$

$Q_X = Q_A \times Q_B$

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

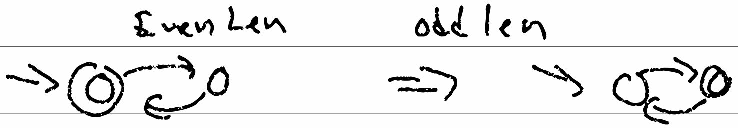
$F_X = F_A \times F_B$ — $\cap$

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

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

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

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



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

or F^c (wrt Q)

complement

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

3-11 DFA ⇒ example or false

DFA :

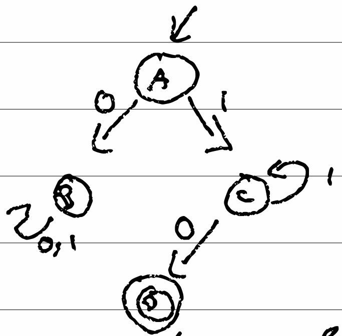
Q : State → Bool

Σ : list of characters

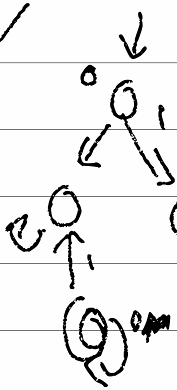
q_0 : state

δ : (state × (char)) → State

F : State → Bool



10 ✓



Q	∅	q_0
{A, B, C, D}	{}	[A]
{B, C, D}	{A}	[B, C]
{C, D}	{A, B}	[C]
{D}	{A, D, C}	[D]
{}	{A, B, C, D}	[]

A ⇒ ε

B ⇒ A, 0

C ⇒ A, 1

Yes, it's possible.

D ⇒ C, 0

or if not, No!

10

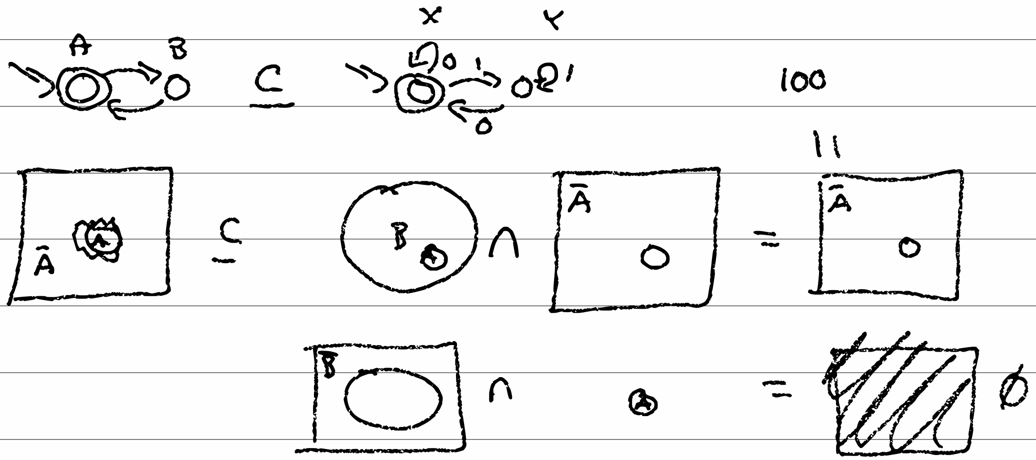
3-2/ subset

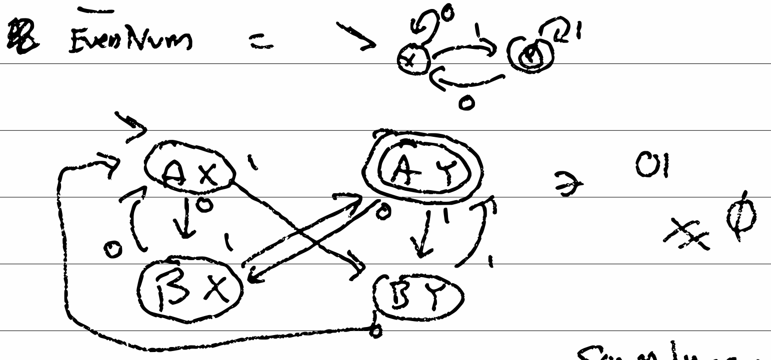
$\exists x \in U.\ x \in A \wedge x \notin B$

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

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

→ finite means naive works!





model checking

Soundness: model ⊆ theory

completeness: theory ⊆ model

model = theory

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

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

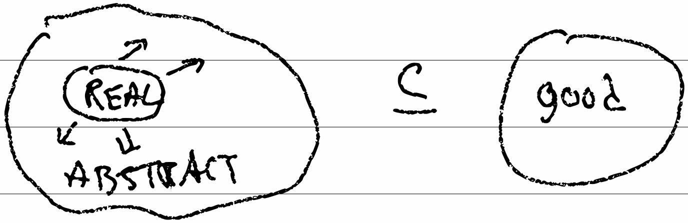
if $x > 0$ then

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

o.w

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

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



3-4) Finite $=$ $\emptyset$ $\{x\}$ $A \cup B$ $\in$DFA

A^c $A \cap B$ $A \circ B$

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

$abcd = a \circ b \circ c \circ d$

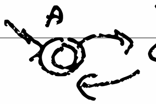
$\{j\} \circ \{m, n\} = \{jm, jn\}$

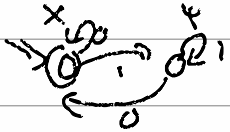
$x \in A^*$ iff $x = x_0 \circ x_1 \circ \ldots \circ x_n$ for $n \in \mathbb{N}$

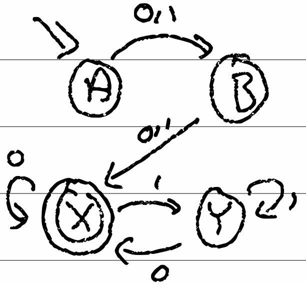
and $x_i \in A$

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

3-5/ Even Len ∘ Even Num



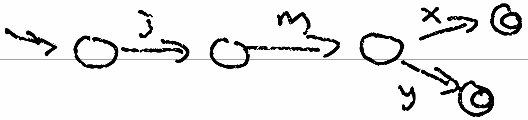


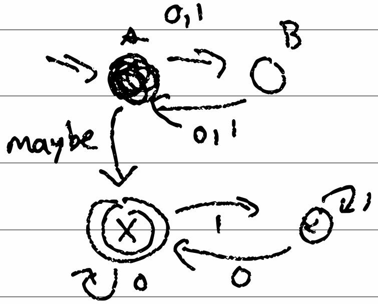


$\underbrace{00}\underbrace{110}$ ✓

0011 ×

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

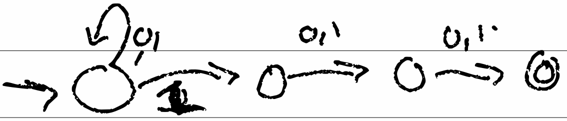




$\Sigma = \{0, 1\}$

$x \in$ 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 = <u>non</u>-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?

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

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

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

epsilon

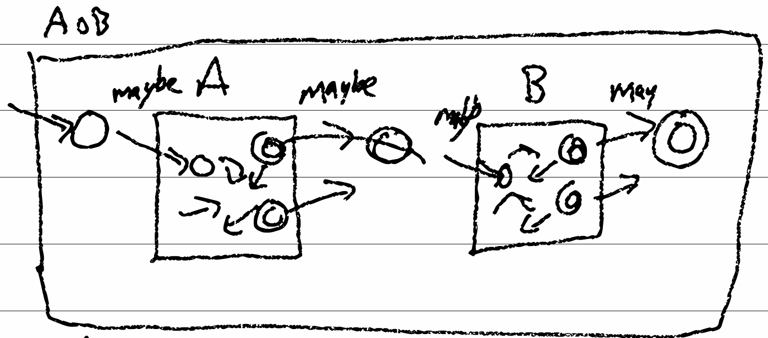
$\varepsilon \quad \Sigma \quad e \quad \textcircled{\varepsilon}$

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

A^* $B \in DFA$

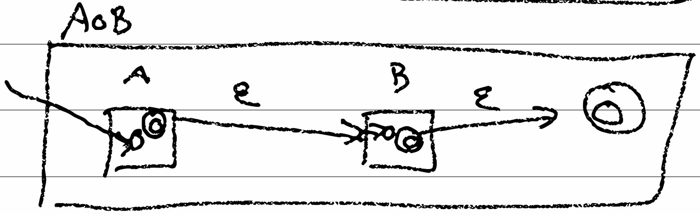
NFA (N - <u>non</u> D- deterministic)

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



maybe written as "ε"

What does (NFA) this mean?



NFA ⟺ DFA

4-2 | What do NFAs mean?

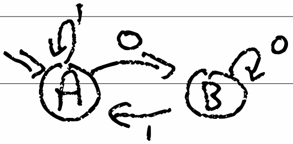
A DFA represents a set and
a set "is" a membership function

$\mathcal{U} \Rightarrow \text{Bool}$

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

config $= \Sigma^* \times Q$

$\Sigma^* \Rightarrow Q^*$



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

$\Sigma^* \Rightarrow (\Sigma, Q)^*$

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

Σ_e make $= \Sigma \cup \{\epsilon\}$

0A 1A 1A 0B → N

$\text{valid?} : Q \times (\boxed{\Sigma}, Q)^* \Rightarrow \{Y, N\}$ DFA

$\text{valid } q_i\ \varepsilon = Y$

$\text{valid } q_i\ (c, q_j) : \text{more} = \text{if } \delta(q_i, c) \boxed{==} q_j$

$\text{valid } q_j \text{ more}$

o.w. N

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

$\text{Nvalid } q_i\ \varepsilon = Y$

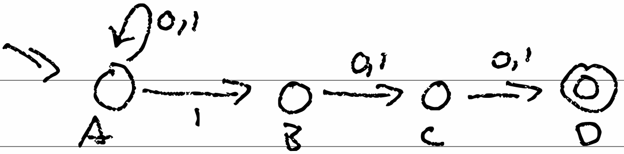
$\text{Nvalid } q_i\ (c, q_j) : \text{more} =$

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

$\text{Nvalid } q_j \text{ more}$

o.w. N

4-3



"3rd letter from end is 1"

0100 111 110100 — Y

000 1000 1011 — N

$(0,A)(1,A)(0,A)(0,A)$ ✓

$(0,A)(1,B)(0,C)(0,D)$ ✓

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

$\text{str} : (\Sigma \times Q)^* \to \Sigma^*$

$\text{str } \varepsilon = \varepsilon$

$\text{str } (c,-)::\text{move} = c \circ \text{str move}$

$\text{accepts} : \Sigma^* \to Y/N$

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

valid q_0 t = Y

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

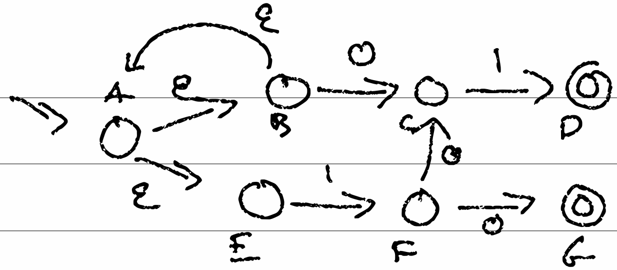
$\text{NFA-accepts} : \Sigma^* \to Y/N$

figure all possible traces

check if valid and its strings match

check if last is in $\in F$

4-4



(ε, B) (0, C) (1, D) $\quad 01 = \varepsilon \circ 0 \circ 1$

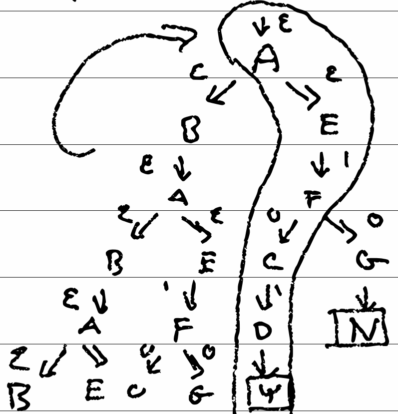
(ε, E) (1, F) (0, C) (1, D) $\quad 101 = \varepsilon \circ 1 \circ 0 \circ 1$

(ε, E) (1, F) (0, G) $\quad 10 = \varepsilon \circ 1 \circ 0$

(ε, B) (ε, A) x where x is valid

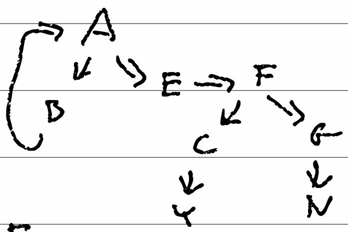
⇒ valid

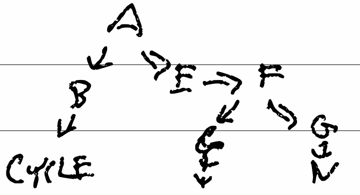
Trace Tree = Y | N | Branch (Σ, Q) (List TT)



101

ε,E 1,F 0C 1D



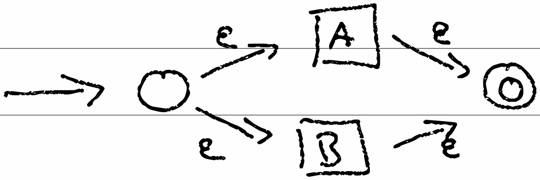


Forking model of NFAs (make TT)

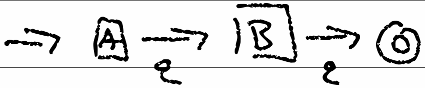
Back-tracking model (explores TT)

4-5/

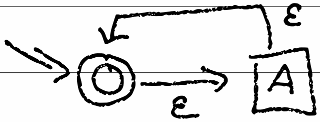
$A \cup B$

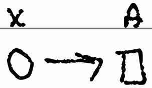


$A \circ B$



A^*





State X transitions
to THE start
state of A

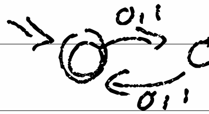
All accepting states
of A transition to X

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

$\Longrightarrow$

$\Longleftarrow$

DFA ⇒ NFA/



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

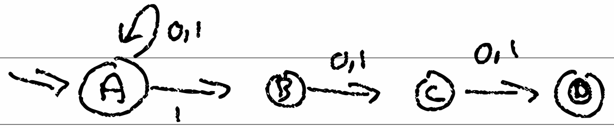
$$NFA \quad \delta': Q \times \Sigma_\varepsilon \to P(Q)$$

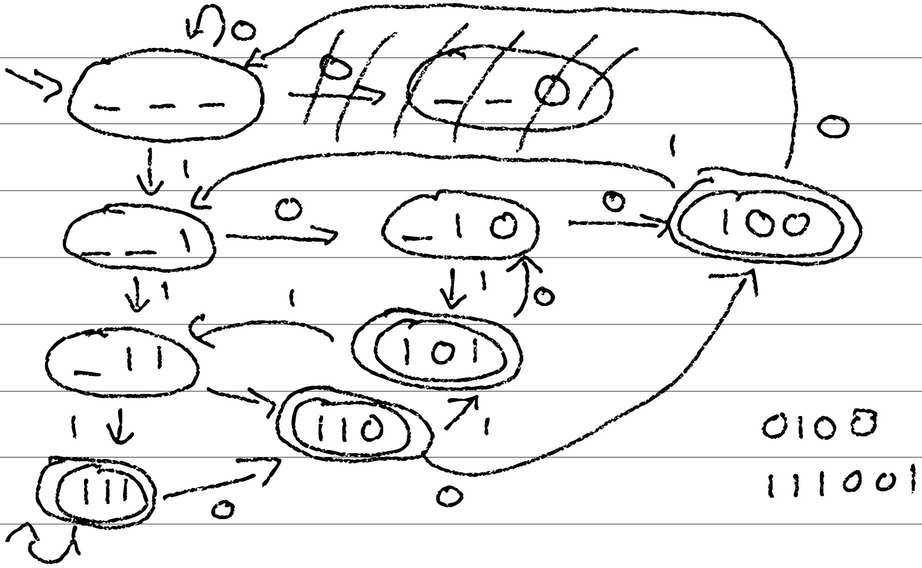
$$\delta'(q_i, \varepsilon) = \emptyset$$

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

NFA ⇒ DFA/

3rd is 1

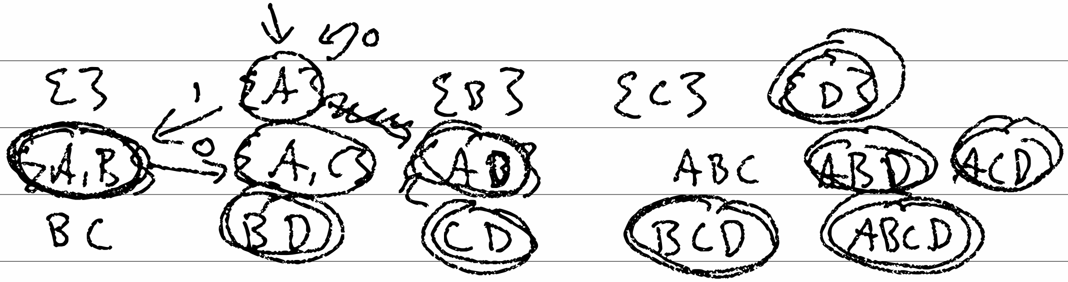




4-7/ in NFA = $(Q, \Sigma, q_0, \delta : Q \times \Sigma_\varepsilon \to P(Q), F)$

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 $\cap F \neq \emptyset$

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

5-1/ $A \cup B$ $\quad \delta_A : Q_A \times \Sigma \rightarrow Q_A$

$\delta_B : Q_B \times \Sigma \rightarrow Q_B$

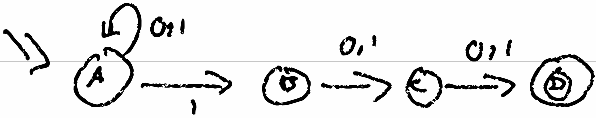
$$\delta' : \underset{(Q_A \times Q_B)}{Q} \times \Sigma \rightarrow Q$$

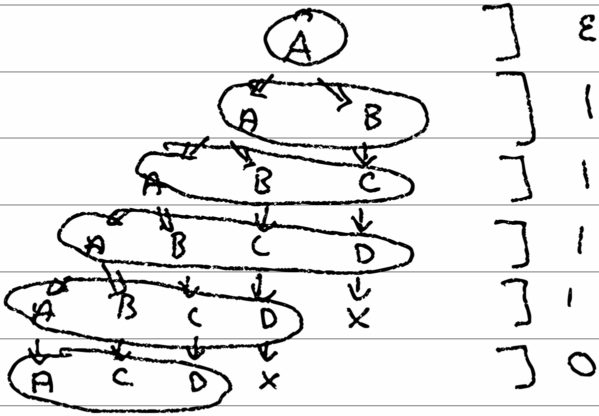
$$\delta((q_a, q_b), c) = (\delta_A(q_a, c), \delta_B(q_b, c))$$

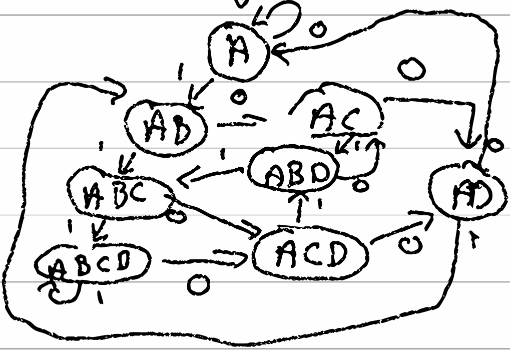
```
      char
(p, c)  =>  new Pair( deltaa(p.fst, c),
                      deltab(p.snd, c));
Pair<State, State)
     fst    snd
```

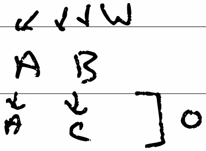
5-2/ NFA ⟹ DFA

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

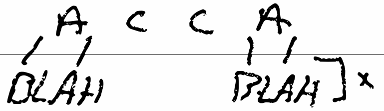


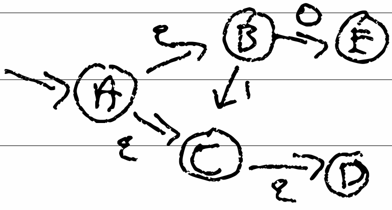






A
A] 0





A B C D]

C D] 1

5-3

$E : Q' \to Q'$ — follow all ε-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 qi in set-of-qs
        if dfa.F.apply(qi) then
          return true
    return false
```

5-4)

```
E ( set<Q> qi )
    queue<Q> next = ~~empty~~ qi
    set<Q> seen = empty
    while (nxt isn't empty)
        add next.first to seen
        δ( next.first , ε) add those
            to next unless in seen
    return seen
```

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

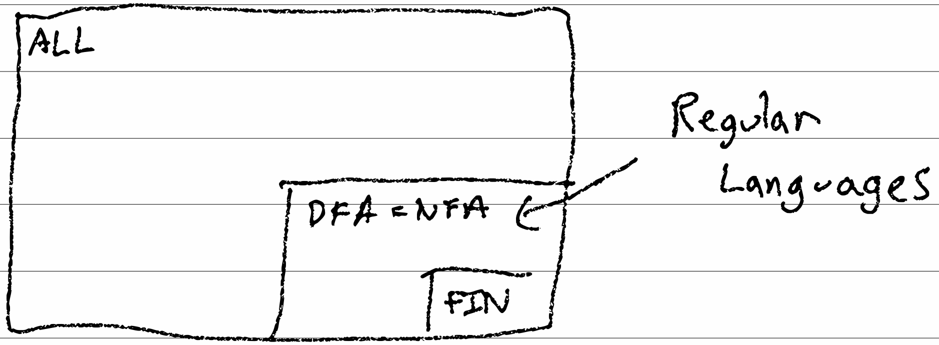
$$E^*(A) = A \cup \bigcup_{q_i \in A} \delta(q_i, \varepsilon)$$

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

$N = D$

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

$D = N$

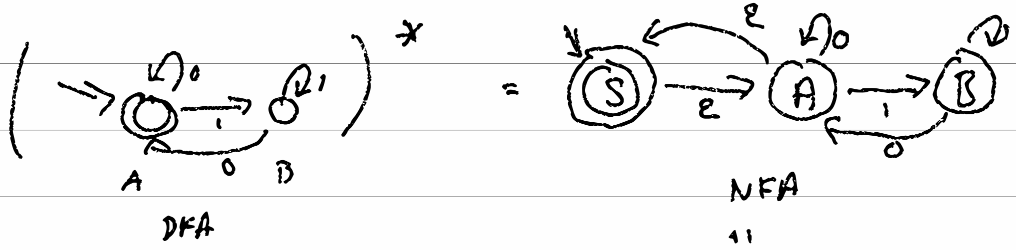


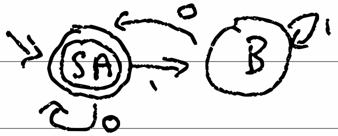
program f ....

program g ....

" f ; 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 { }

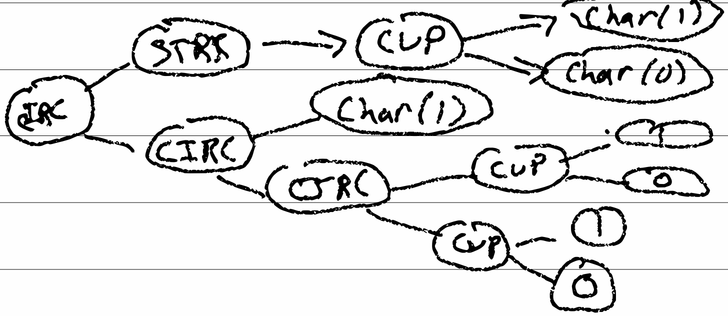
class RE_EMPTY() impl REgex {{ . . .

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.)

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

$L(\emptyset) = \emptyset$

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

$L(r \overset{\text{R NAME}}{\cup} r') = L(r) \overset{\text{OR}}{\cup} L(r')$

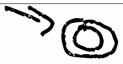
$L(r \circ r') = L(r) \circ L(r')$

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

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

compile : RE → NFA

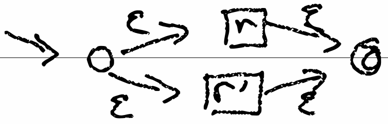
compile (ε) =



compile (∅) =

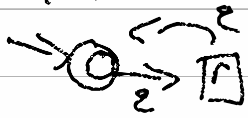
compile (c) =

compile (r ∪ r') =



compile (r ∘ r') =

compile (r*) =



6-4)

$" \ " = \varepsilon$

$\underline{\qquad} = \emptyset$

$"c" = c$

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

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

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

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

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

[012]

(zero | one | ~~two~~)

$\bullet = \Sigma$

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

6-5

$gen : RE \to \Sigma^* \text{ or false}$

$gen\ \varepsilon = \varepsilon$

$gen\ \emptyset = \text{FALSE}$

$gen\ c = \text{'c'}$

flip coin

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

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

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

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

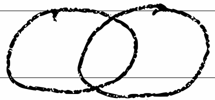
", mjs"

$equal : RE \times RE \to Bool$

$equal\ x\ y =$

NFA2DFA(compile x)

NFA2DFA(compile y) — dfa equality?



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

6-6

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

$x \bullet 1 = x$

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

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

$3x + 17 = 3x + 17$

$3x = 3x$

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

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

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

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

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

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

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

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

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

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

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

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

$\text{size}\ c = 1$

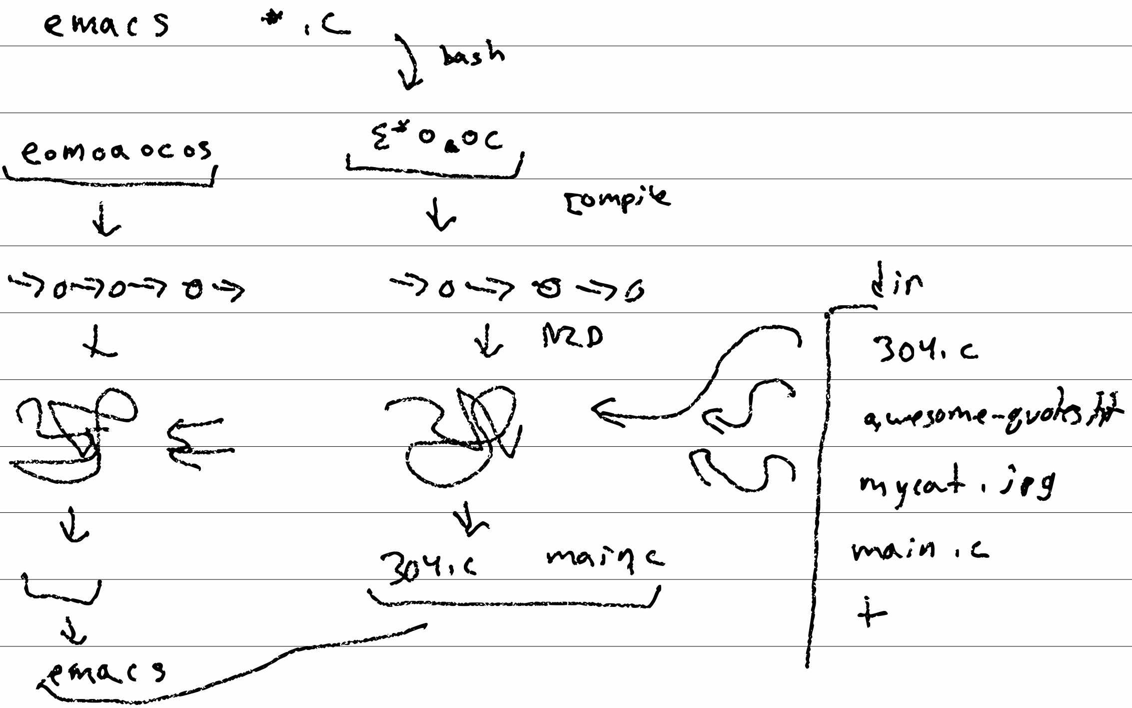
$\text{size}\ (x \cup y) = sz\ x + sz\ y + 2$

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

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

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

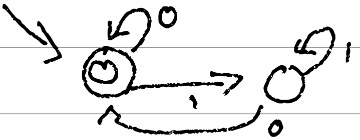
↑

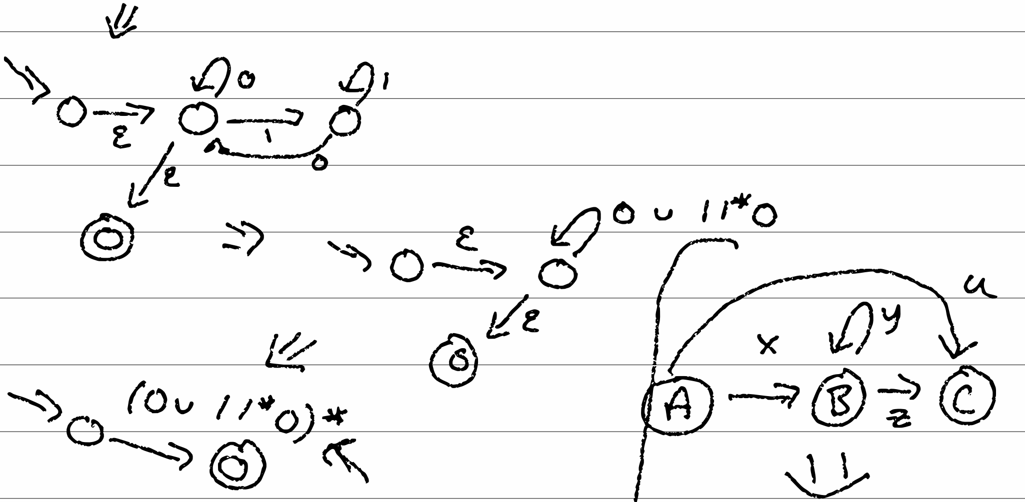


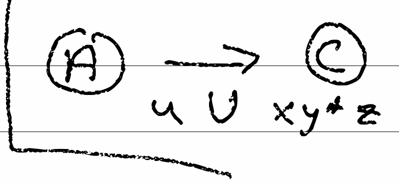
6-8 decompile: (NFA) or DFA → RE

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

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







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

ε 0 10 111110

0 1110 1111100

$\underbrace{1110}\underbrace{0}\underbrace{0}\underbrace{1110}\underbrace{1110}\underbrace{0}\underbrace{10}\underbrace{10}\underbrace{10}$

G-9/ decompile : N-state NFA
$\Rightarrow$ RE

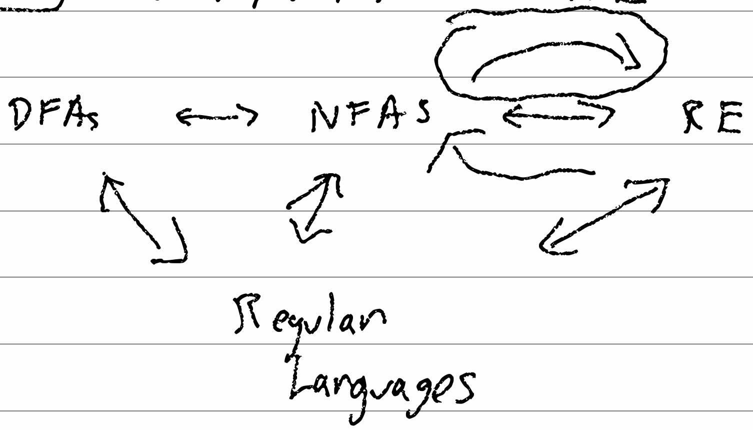
START : N-NFA $\Rightarrow$ (N+2)-GNFA

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

END : 2-GNFA $\Rightarrow$ RE

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

7-1/ DFA/NFAs → RE



NFA → RE

$\text{IN}: n\text{-NFA} \rightarrow (n+2)\text{-GNFA}$

$\text{RIP}^n: (n+1)\text{-GNFA} \rightarrow n\text{-GNFA}$

$\text{OUT}: 2\text{-GNFA} \rightarrow \text{RE}$

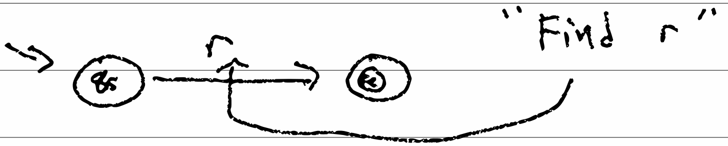
$\text{GNFA} = (Q, \Sigma, \overset{\in Q}{q_s}, \overset{\in Q}{q_e}, \Delta : (Q - q_e) \times (Q - q_s) \rightarrow \text{RE}(\Sigma))$

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

$\text{OUT}: 2\text{-GNFA} \rightarrow \text{RE}$

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

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



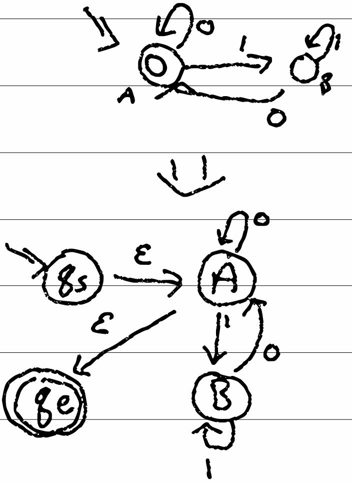
7-2)

[illegible]

IN: NFA → GNFA

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

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



$\Delta(q_i, q_j) = r$

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

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

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

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

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

7-3/ RIP = $(n+1)$-GNFA $\to$ n-GNFA

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

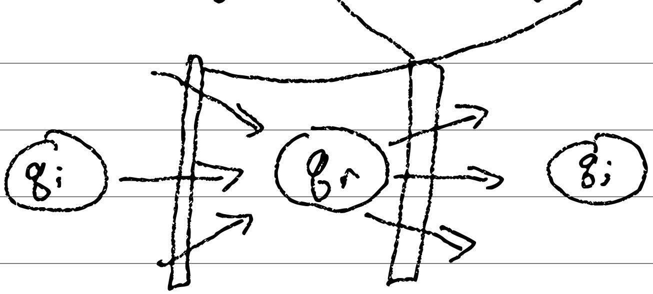
main
↓
jay
main
↓
$\boxed{R}$
↓
g
↓ jay
exit
↓
exit

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

$= q_r$

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

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

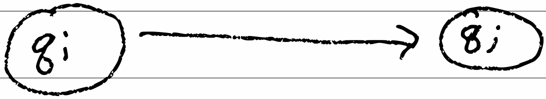


$x \circ y^* \circ z \cup a$

~~$x \circ z$~~

$\Delta'(q_i, q_j)$

$=$

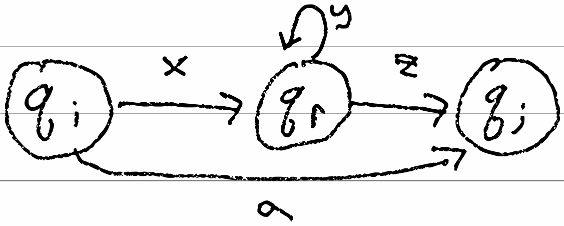


$x = \Delta(q_i, q_r) \circ$

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

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

$a = \Delta(q_i, q_j)$



7-4/ int main() {

int x = 8;

~~int y = f(x, z);~~

....

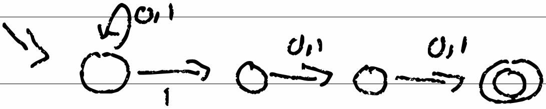
int y = x + 2 * z;

int f(int z) {

int 8

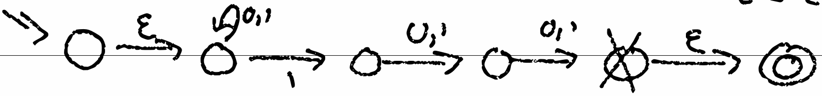
return

z + 2 * 8; }



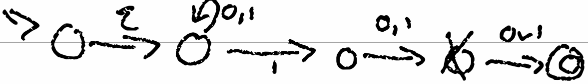
$0\cup1 = x \quad y = \emptyset \quad z = \varepsilon \quad a = \emptyset$

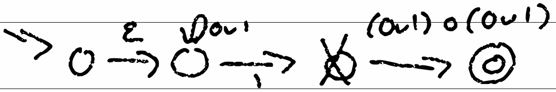
IN

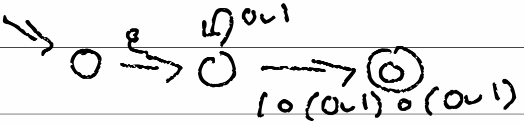


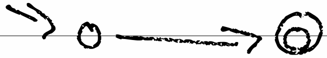
$x \circ y^* \circ z \cup a = (0\cup1)\circ\emptyset^*\circ\varepsilon\cup\emptyset$

$= (0\cup1)$



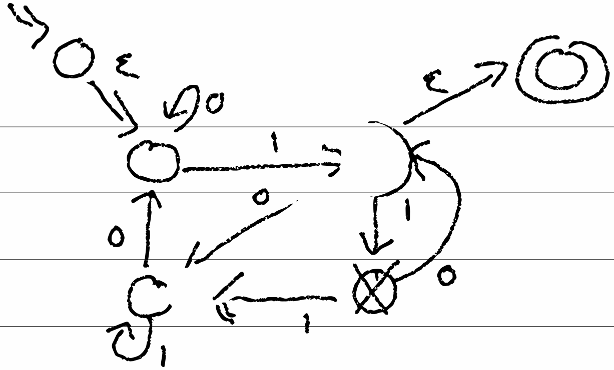


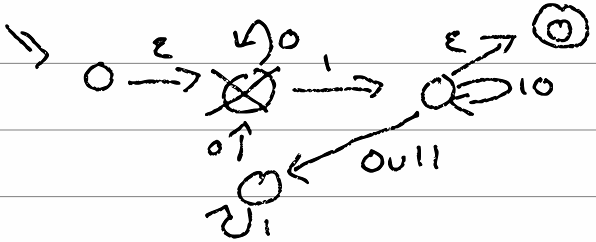


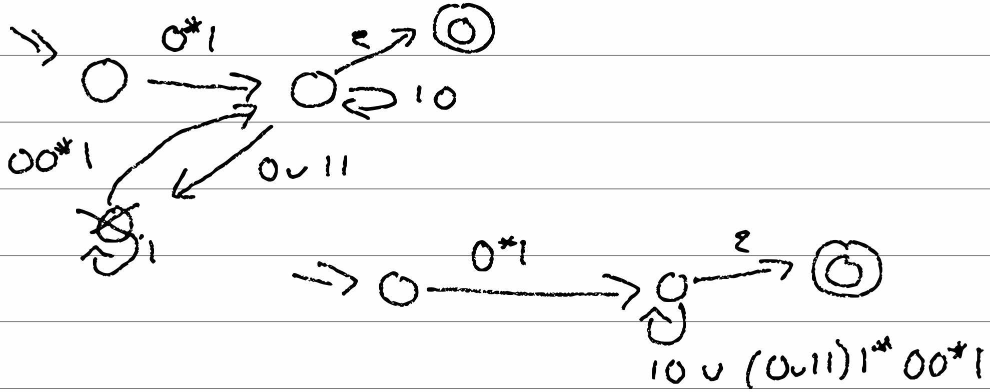


$\varepsilon\circ(0\cup1)^*\circ1\circ(0\cup1)\circ(0\cup1) = \Sigma^*1\Sigma\Sigma$

7-5





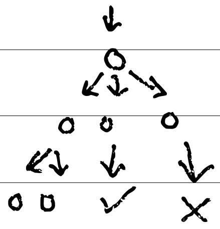


$(0^*1)(10 \cup (0\cup 11)1^*00^*1)^*$

8-1 ε Object — Char(char c)
— Epsilon()

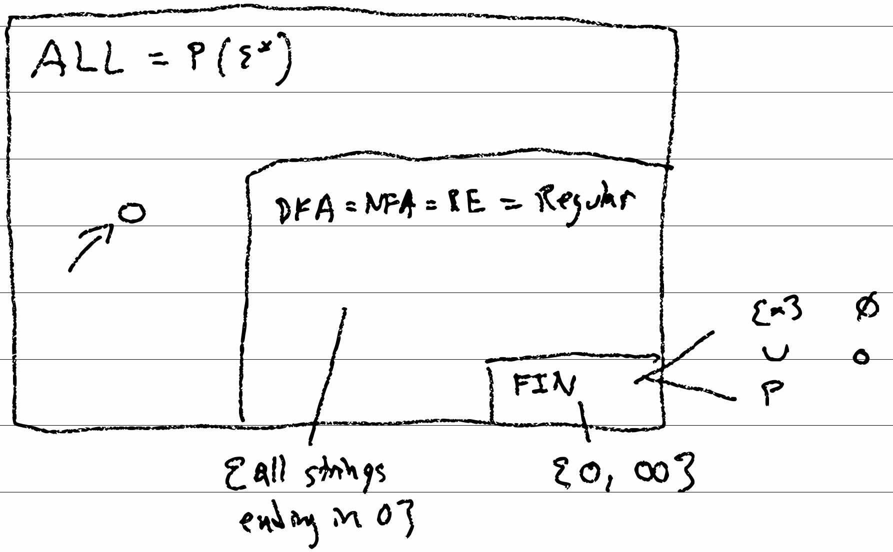
UTF-8

TT



TT = Succ | Fail
| Branch State List(TT)

List(TT) → Char × TT



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

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

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

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

8-2/ How can know stuff about infinite sets?

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

$$\Rightarrow \overbrace{q_0 \in Q}^{P : DFA \to Prop}$$

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

$\neg P(B)$ (where $B \in \mathbb{P}(\Sigma^*)$ 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 $B \notin REG$.

$\Sigma = \ldots.$

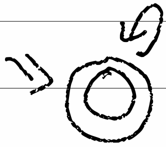
8-31

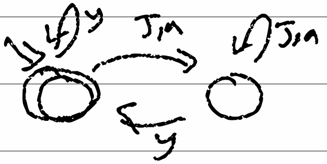
"Jay" $\in d$ $\Sigma \ni \{J, a, y\}$

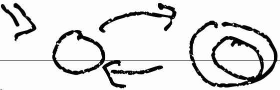
$\gg q_0 \xrightarrow{J} \xrightarrow{a} \xrightarrow{y} q_f \in F$

☒ ☑ ☑ ☑ ☒

$|d.Q| = 1$







Daphne wins

pick a number of states ; 4

she picks a char

I say what state we go to

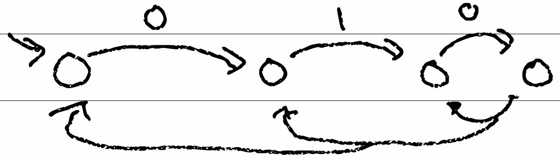
I win if I never say same state

She wins if I repeat

How many turns to win?

$4 \Rightarrow N \Rightarrow$

$|Q|$

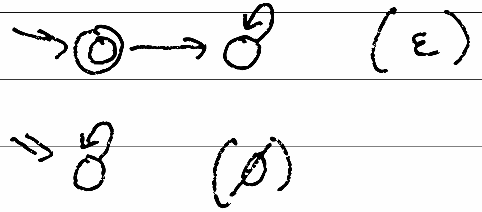


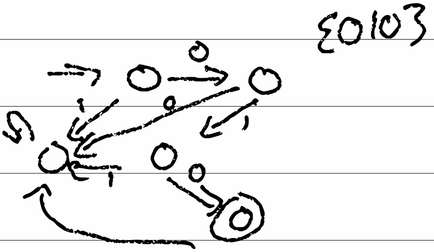
8-4

$\delta : Q \times \Sigma \to Q$ Total Fun

from 0, 1

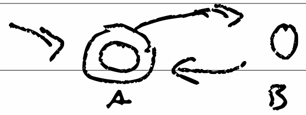
All DFAs have a loop





Some have "exciting" loops

∃ s ∈ L(DFA) . s = ...

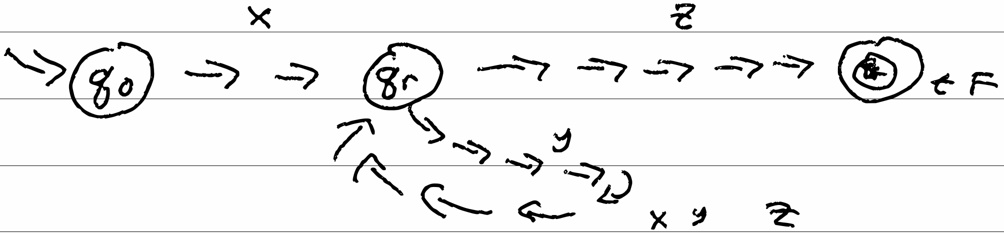


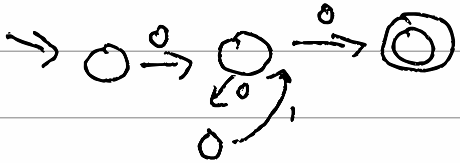
(Even lengthed)

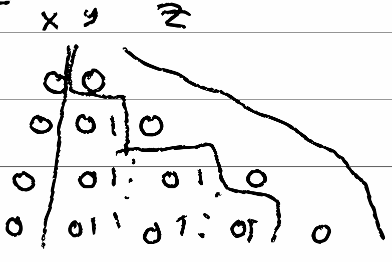
ε = A

= ε∘ε∘ε

0110 = ABABA = ε∘01∘10







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

They are all finite

$L(m) \in FIN$

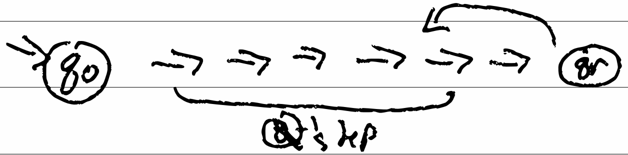
If it does have one ...

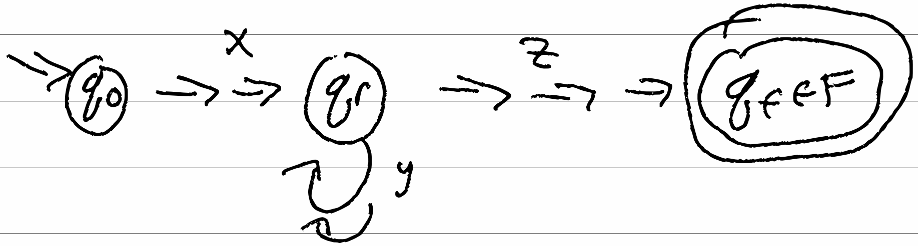
then it must be infinite (regular)

If there is an exciting loop ....

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

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





$\in$ 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

$0^* 1$ →

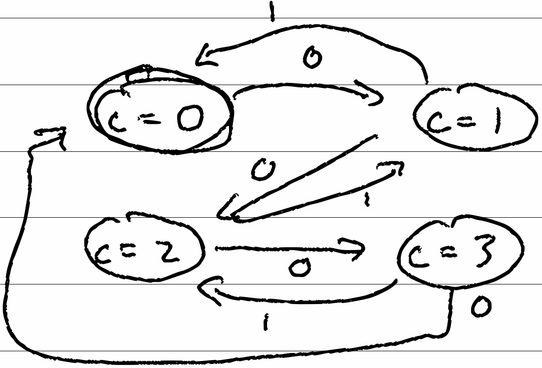
```
while (getc() == '0') {}
ungetc()
if (getc() != '1') {ret false}
return getc() == EOF;
```

$= 2^{32}$ uint2_t

$\forall n \in N,$

$0^n 1^n \in B$

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



9-1) $\text{RPP}(A) \rightarrow$ Language = $P(\Sigma^*)$ = ALL

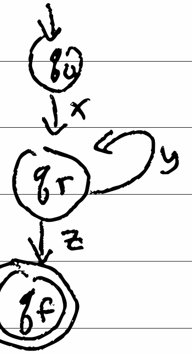
$\exists p \in \mathbb{N}.$ // $p = |Q|$

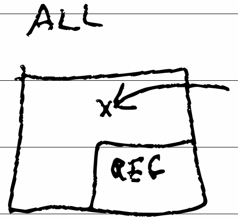
$\forall (s \in A \mid |s| > p).$

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

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

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





$\neg \text{RPP}(A) =$

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

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

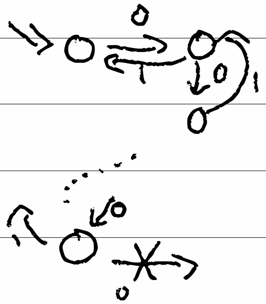
$\forall (x, y, z \in \Sigma^* \mid s = xyz \wedge |xy| < p \wedge |y| > 0)$

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

$xy^iz \notin A$

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

iff $\exists n \in \mathbb{N},$ $x = \overbrace{00\ldots0}^{n}\overbrace{1\ldots1}^{n}$



```
~~long~~ nat
~~int~~ count = 0
while ( see 0) c++
while (see 1) c--
return count == 0
```

$\underline{9\text{-}2)}$ $\neg RPP(0^n 1^n) =$

$\forall p \in N.$

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

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

$|y| > 0$)

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

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

$z = 0^c 1^d$

$\exists i \in N. \quad xy^iz \triangleq 0^n 1^n$

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

$\underset{-p}{a+bi+c} = \underset{-p}{d}$

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

```
fun n:
  if n == 0 : ret pz
  else: let m = n-1
    let pm = rec m
    ps pm
```

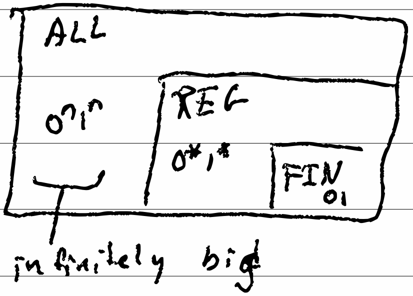
induction on N:

pz: $P\ 0$

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

$\Rightarrow \forall n : P\,n$

9-3/ There is stuff outside REG!

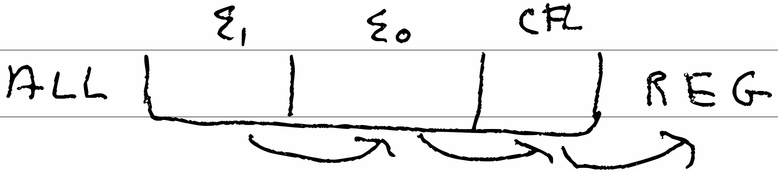


All computers are DFAs.

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 $w \in \Sigma^*$

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

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

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

01 0100 ⇒ "1 + 1 = 2" ✓

0010001 00000 ⇒ "2 + 3 = 5" ✓

001 0010 ⇒ "2 + 2 = 1" ✗

∀p.

∃s. $0^p 1 0^p 1 0^{2p}$

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

∃i.

$\underbrace{2}+2=4$

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

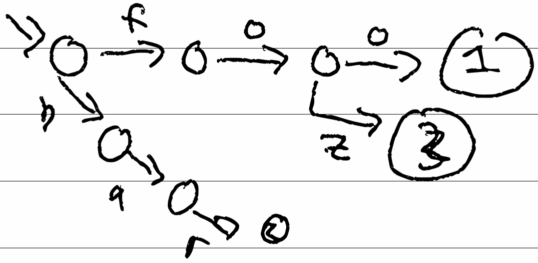
12-1 trie

foo ↦ 1

bar ↦ 2

foz ↦ 3

digital
search
tree



CFGs

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

start variable

A $\boxed{S} \to \varepsilon$

B $S \to 0S1$

← rule, productions, transitions

rule = $\underbrace{V} \to \overset{\text{rhs}}{\underbrace{\quad\quad}}$

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 $\mu\left(\begin{matrix} S \to 01S \\ S \to SS \end{matrix}\right) = \emptyset$

Context-free grammar

$0101 \in$ ↑

alphabet ↗

(V, Σ, R, S)

↓ 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 B

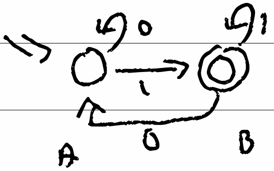
→ $A \cup B$: $S \to A, S$

$S \to B, S$

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

$A \cap B$: X

DFA → CFG



$$\begin{array}{llll} & x & y & \\ A \to & 0A & \mid 1B & \\ B \to & 1B & \mid 0A & \mid \varepsilon \\ & z & w & v \end{array}$$

01101

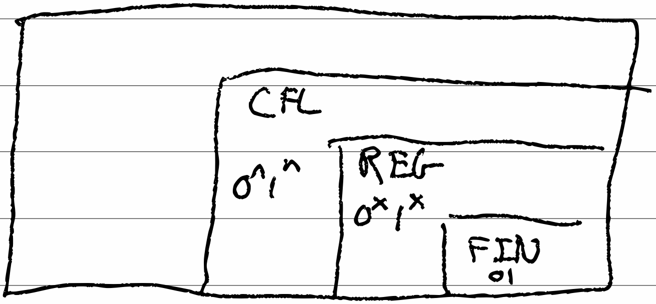
$S = q_0$

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

$\xrightarrow{w} 0110A \xrightarrow{y} 01101B \xrightarrow{v} 0$[illegible]

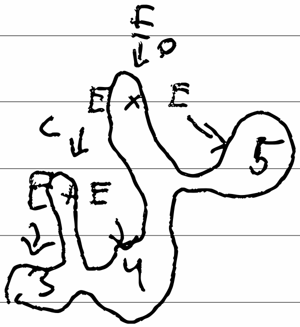
$q_i \to c\, q_j$ iff $\delta(q_i, c) = q_j$

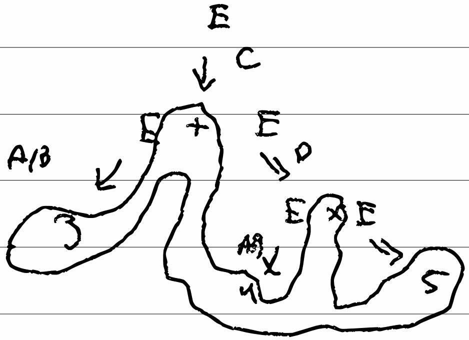
$q_f \to \varepsilon$ iff $q_f \in F$



$$\begin{array}{lllll} & A & B & C & D \\ \text{12-4/} \quad E \Rightarrow & 0 & \mid 1 & \mid E + E & \mid E \times E \end{array}$$

3 + 4 × 5





fowim E → n

fowin 0 = 0

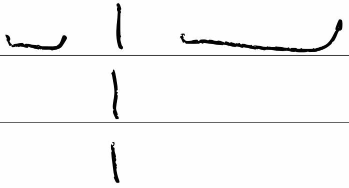
1 = 1

Plus(L, R) = fowim L + fowim R

Mult(L, R) = fowim L × fowin R

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

if cond



12-5) Nice! amb → unamb

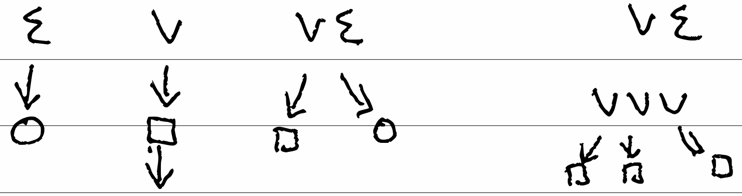
not possible

V, Σ, R, S

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

ε $\Sigma\Sigma V$

V $VVVVV$

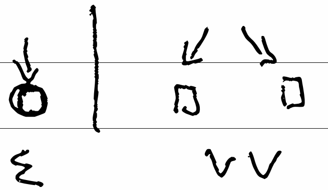


---

want: "pure" binary trees

Noam

Chomsky-Normal Form



$V \Rightarrow \Sigma$ | $A \Rightarrow c$

$V \Rightarrow VV$ | $A \Rightarrow \underset{\neq S}{\underbrace{B}\underbrace{C}}$

$S \Rightarrow \varepsilon$

GFG → 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^n1^n

$S \Rightarrow \varepsilon$

$S \Rightarrow 0S1$

A

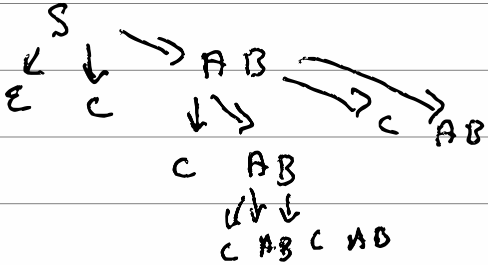
B

$A \Rightarrow c$ +1

$A \Rightarrow BC$ +2

$B \rightarrow$ +2

$A \Rightarrow$ done



$S \Rightarrow XYZ$

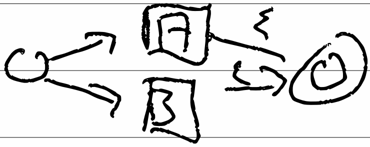
$X \Rightarrow ZZS \mid 0$

$Y \Rightarrow XYX \mid 1$

$Z \Rightarrow YSX$

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

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



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

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

unionstate = start | end | fromA qa

| fromB qb

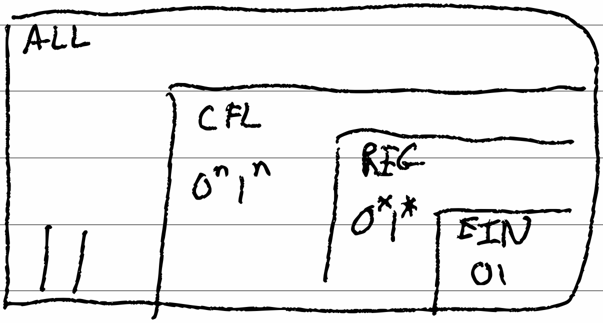
$\delta(\text{start}, \varepsilon) = \{\text{fromA } A.q_0, \text{ fromB } B.q_0)$

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

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,
```

15-1



REGULAR

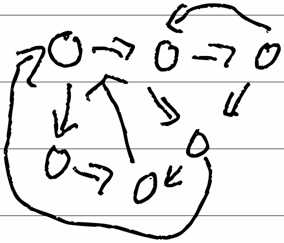
DFAs ⟷ regex

ϵ $\{\}$

CFL

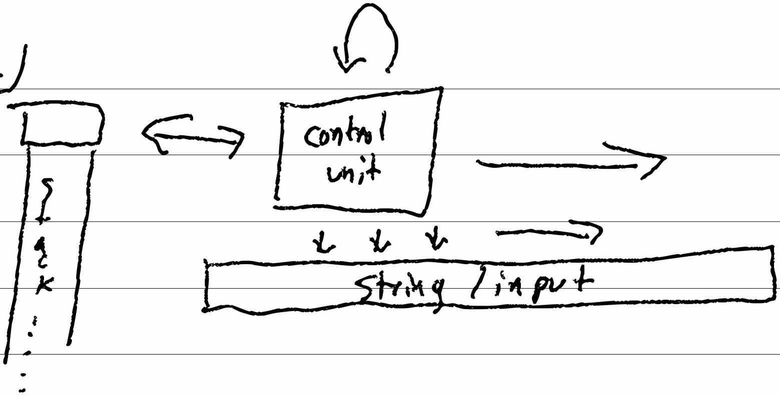
$\boxed{\text{PDA}}$ ⟷ CFG

push-down automata



$\Sigma^*1\Sigma\Sigma$

15-2



$$\text{DFA}: \quad Q \times \Sigma \to Q$$

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

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

$Q \times \Sigma \times \varepsilon \to Q \times \varepsilon$ (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 × $\overset{\text{config}}{\cancel{\;\;}}$ → config

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

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

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

$\text{let}\ (q_j, \gamma) \in \delta(q_i, c, \alpha)\ \text{in}$

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

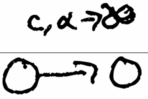
$\text{accepts} : \text{PDA} \times \Sigma^* \to \text{bool}$

$\text{accepts}\ p\ s = \text{while}\ c.s \neq \varepsilon\ \text{do}$

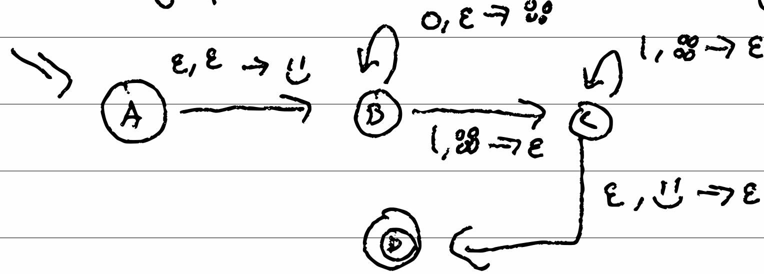
$c = \text{sim}\ p\ c$

$\text{return}\ c.q_j \in p.F$

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

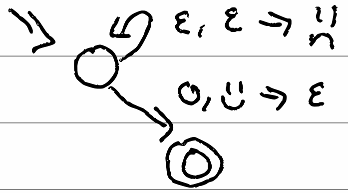


$0^n 1^n$



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

15-4/

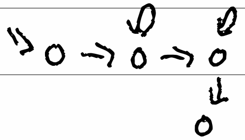


CFG → PDA

S → ε

S → 0S1

⇒



S ⇒ XYZ 011 SYX

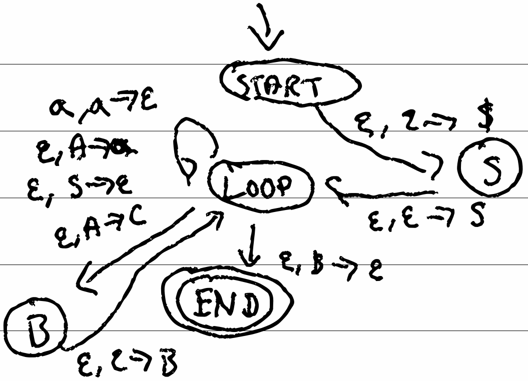
Γ = Σ ∪ V

CNF → PDA

A → BC (B, C ≠ S)

S → ε

A → a



A[LOOP] → C[B] → CB[LOOP]

$[L]0011

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

↓

$1S[L]11 ← $11S0[L]011 ← $1S[L]011

⇙

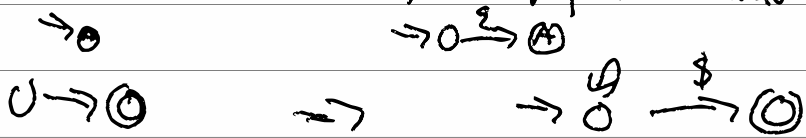
$11[L]11 → $1[L]1 → $[L] → [END] → ✓

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

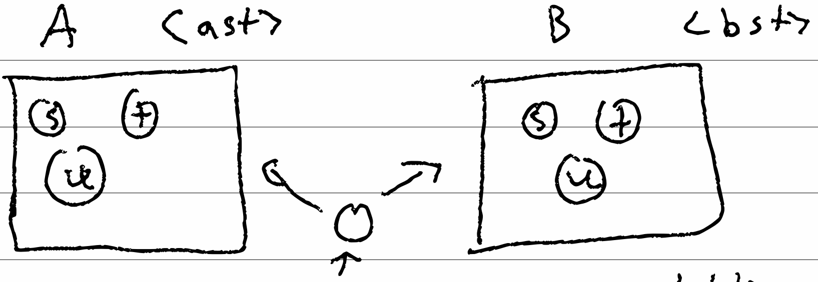
X replace : Γ , Γ

Every symbol pushed is eventually popped

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

$$\frac{\begin{array}{ccc}(r, t) & \in & \delta(p, a, \varepsilon) \\ (q, \varepsilon) & \in & (s, b, t)\end{array}}{(p, q) \rightarrow \underset{\in \Sigma}{a} \; \underset{\in V}{(r, s)} \; \underset{\in \Sigma}{b}}$$

$$(p, p) \rightarrow \varepsilon$$
$$(p, q) \rightarrow (p, r)\,(r, q) \quad \forall p, q, r$$

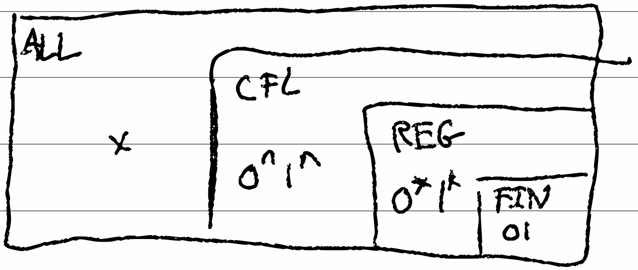


shawn <ast, bst>

$A \cup B$ < [illegible]

shawn <x, y> = start
| anA x
| aD y

16-1



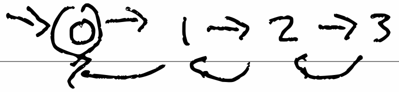
DFAs ⟷ REX $0^n1^n \notin$ REG

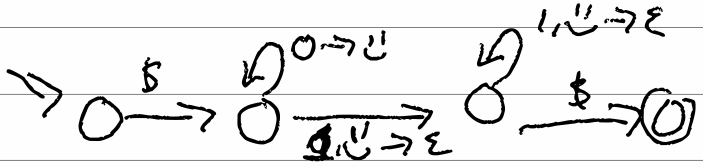
PDAs ⟷ CFG $x \notin$ CFL

CFL = ALL

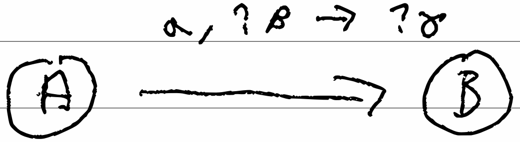
1. P …	$\forall r \in$ REG. $P(r)$	$\forall c \in$ CFL. $P(c)$
2.	$\exists x \in$ ALL. $\neg P(x)$	$\exists x \in$ ALL. $\neg P(x)$
3. ⟹	$x \notin$ REG	⟹ $x \notin$ CFL

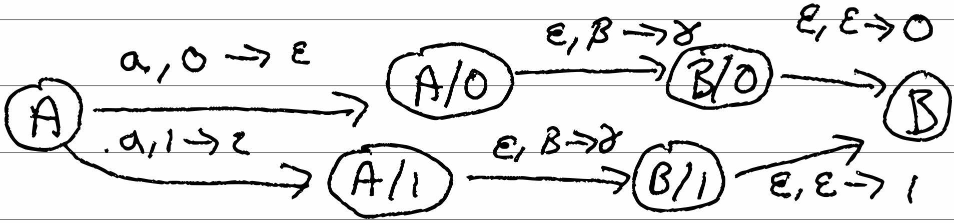
0^n1^n →





16-24 Look at 2nd thing





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

$0^n 1^n$ $\quad$ $0^n 1^* 0^n$ $\quad$ $0^n 1^n 0^n$

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

ε[]$0^n 1^n 0^n$

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

☺^{n}[]$0^n \to \varepsilon$[]ε $\quad$ ε[]0^n

ε[]$0^n 1^n 0^n$ $\Rightarrow$ ☺^{n}[]$1^n 0^n$ $\to$ ☺$^{n-1}$∷^{1}[]$1^{n}0^{n}$

$\Rightarrow$ ∷^{n}[]0^n $\Rightarrow$

16-3/ RPP

$\forall A \in REG,$

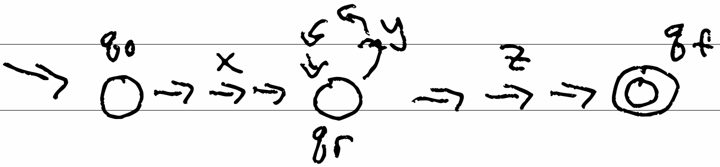
$\exists p \in N,$

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

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

$\forall i \in N,$

$xy^i z \in A.$



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

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

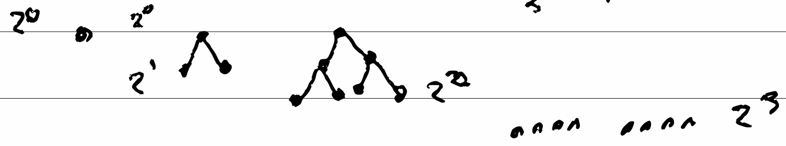
$v_i \Rightarrow v_x v_y$

$$\begin{array}{ccccccc} v_0 & v_1 & v_2 & \ldots & v_n & \Rightarrow & v_r \\ & v_1 & v_2 & & \vdots & & \vdots \\ & & v_2 & & & & \\ & & v_2 & & v_n & & v_r \end{array}$$

$v_r \in V$

$3^3 - 1$

$2^{n+1} - 1$



16-4/ minimum length of tree of height h

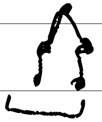
$S \Rightarrow \varepsilon$

$= 0$

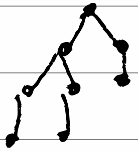
$S \Rightarrow a$

$= 1$

$S \to BC$



2



$h = 4$

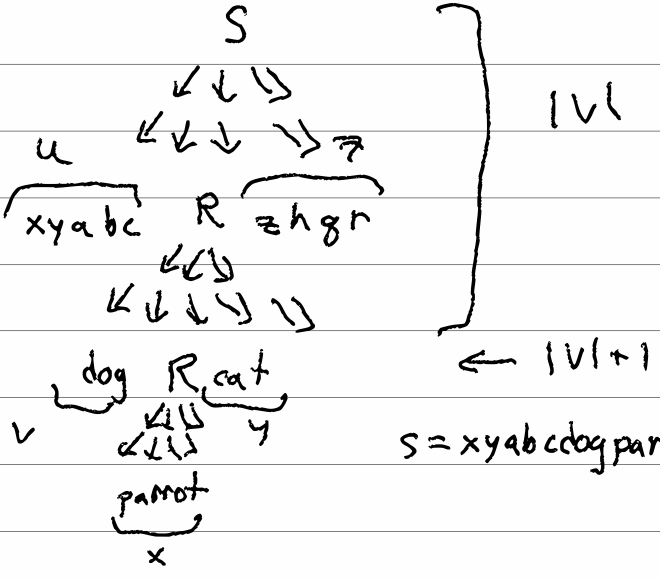
$\Rightarrow$

len ≥ 3

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

$s \in A$

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



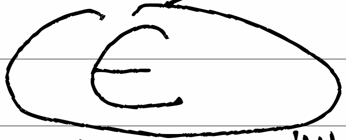
$s = xyabcdogparrotcatzhgr$

$S \Rightarrow uRz$

$R \Rightarrow vRy$

$R \Rightarrow x$

16-5/ Context-free pumping property



$\forall A \in CFL,$

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

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

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

$|vxy| \geq p$

$|vy| > 0)$

$\forall i \in N,$

$uv^ixy^iz \in A$

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

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

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

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

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

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

$\underline{16\text{-}6}$ $0^n 1^n 0^n = A$

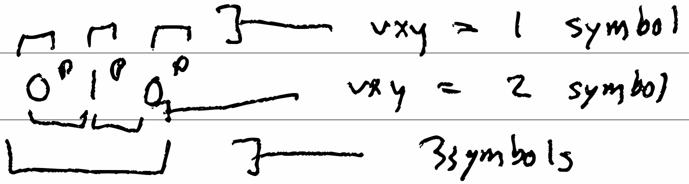
given p

$0^p 1^p 0^p$

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

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

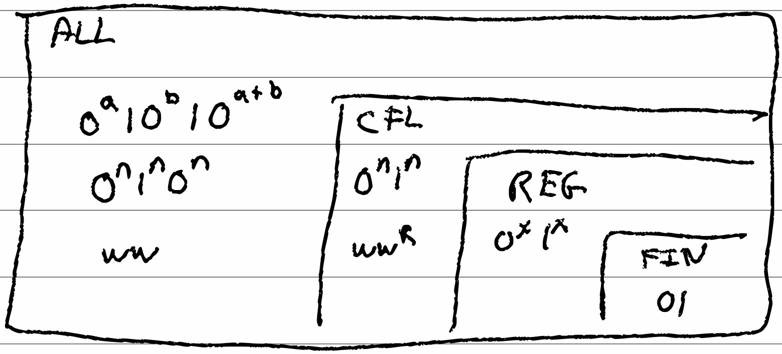
vxy has repetition



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

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

DFAunion = Pair <X, Y> $\qquad$ NFAunion = Start
stateFromLeft X ("x", X)
stateFromRight Y ("y", Y)
END



17-1 Finite : 01

Regular: $0^* 1^*$

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

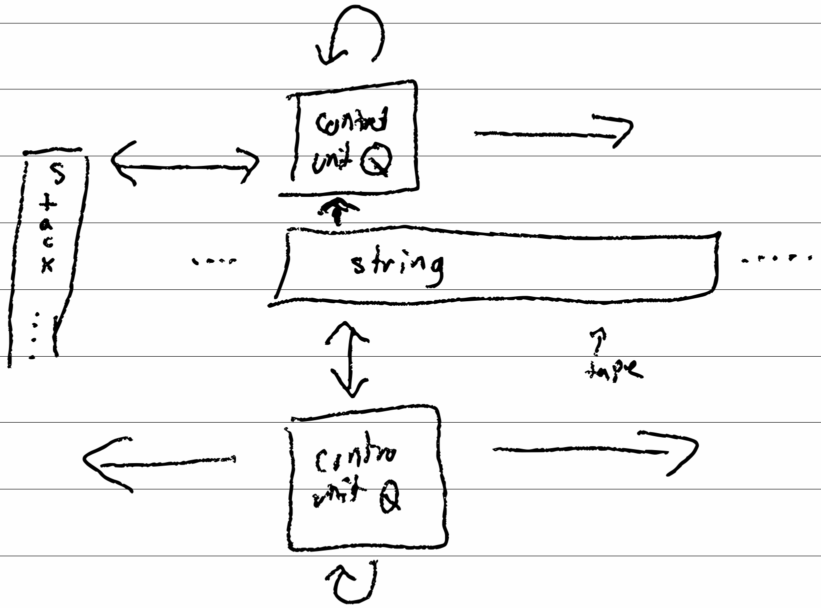
$\notin$ CFPP $0^n 1^n 0^n$

context-free pumping property

$\nRightarrow$ CFG, PDA

Turing Machine Alan Turing

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



17-2

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

states : Q

input : Σ $\qquad \sqcup \notin \Sigma$

tape : Γ $\quad (\Sigma \subseteq \Gamma)$ $\qquad \sqcup \in \Gamma$

$q_0 \in Q$

$q_a \in Q$ (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 → tape

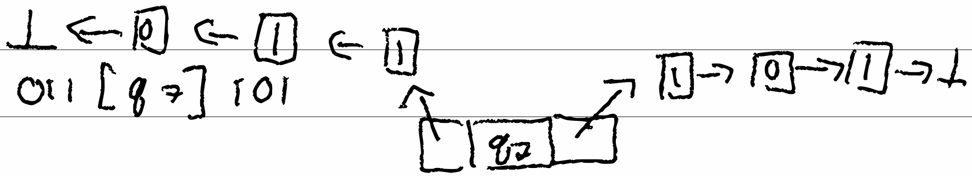
left ε = $\varepsilon : \sqcup$

left $(x : c)$ = x

right : tape → tape

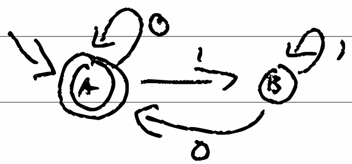
right ε = $\sqcup : \varepsilon$

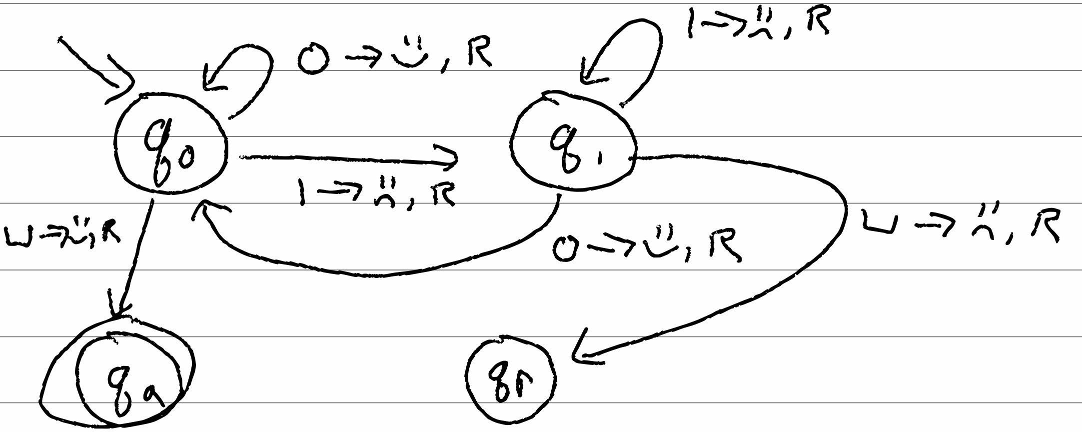
right $(c : y)$ = y



(DFA)

17-3





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

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

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

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

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 $q_i \notin F$

Σ = {0, 1, #}

17-4/ w # w where w ∈ {0,1}*

I saw a zero

[] 01#01 → ␣[]1#01 → ␣1[]#01

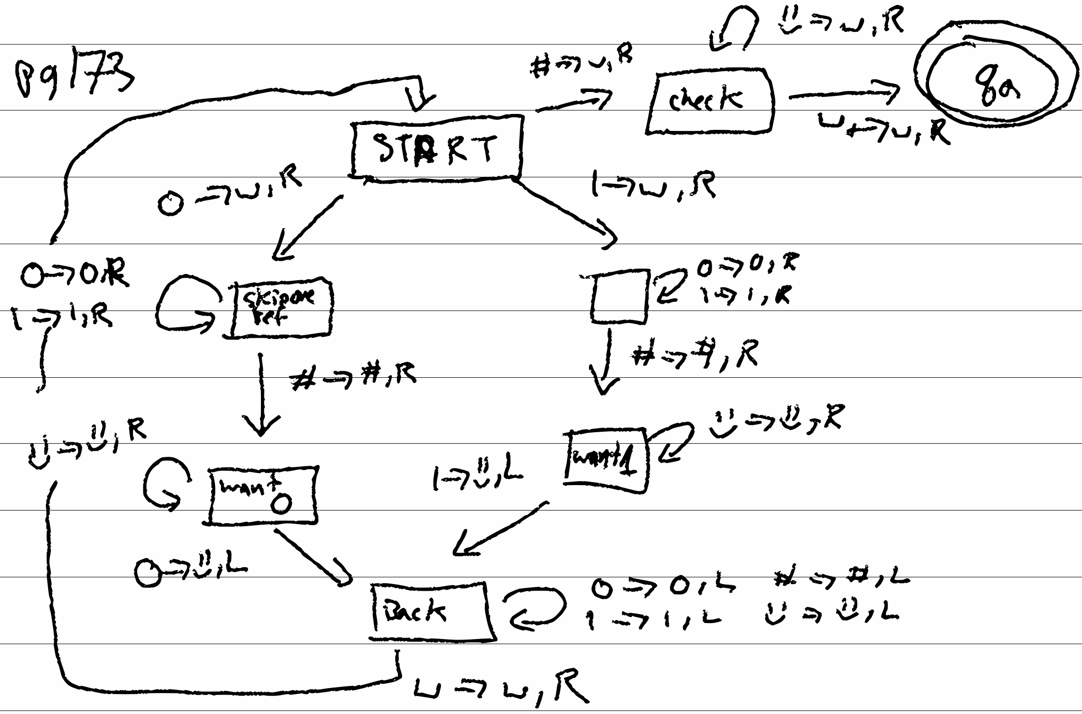
␣1# [I expect a zero] 01 → ␣1 [I am happy] # ☺ 1

→ ␣ [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 $0 = 1$

$1 = 10$

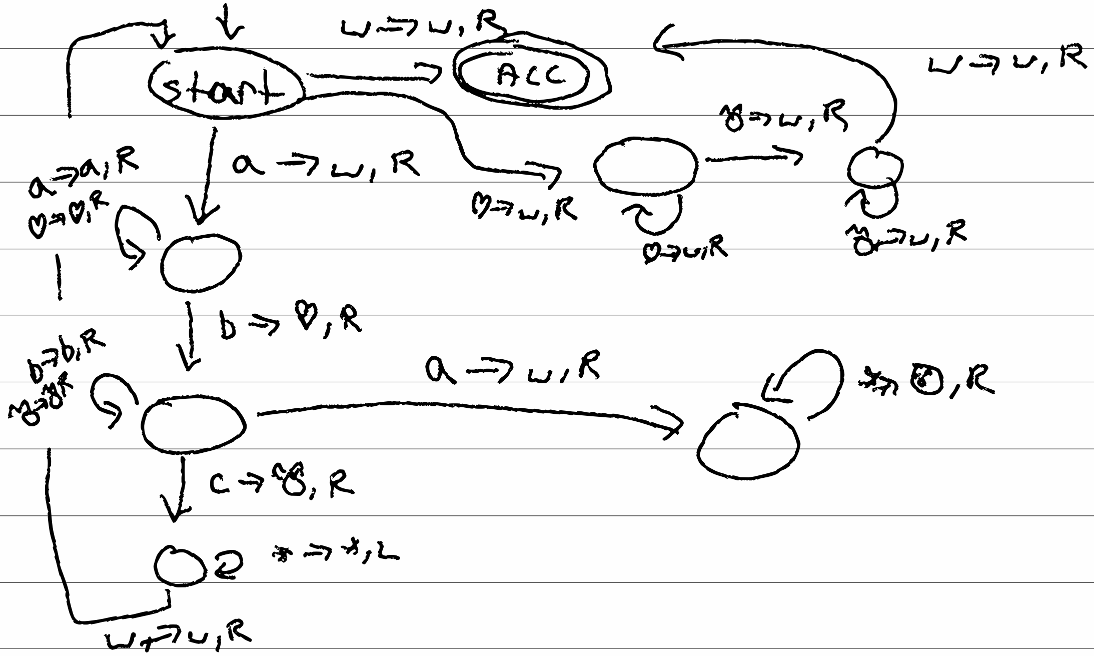
$10 = 11$

$11 = 100$

$$\overset{(\supseteq \Sigma)}{}$$

18-1) $\delta: \underset{Q - \{q_a, q_r\}}{Q} \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

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



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

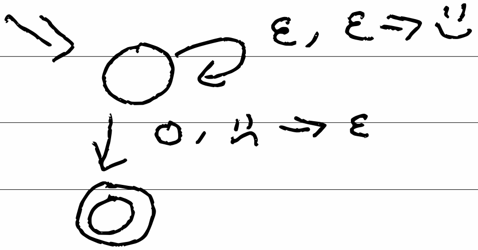
$$f(0^x + 0^y) = 0^{x+y}$$

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

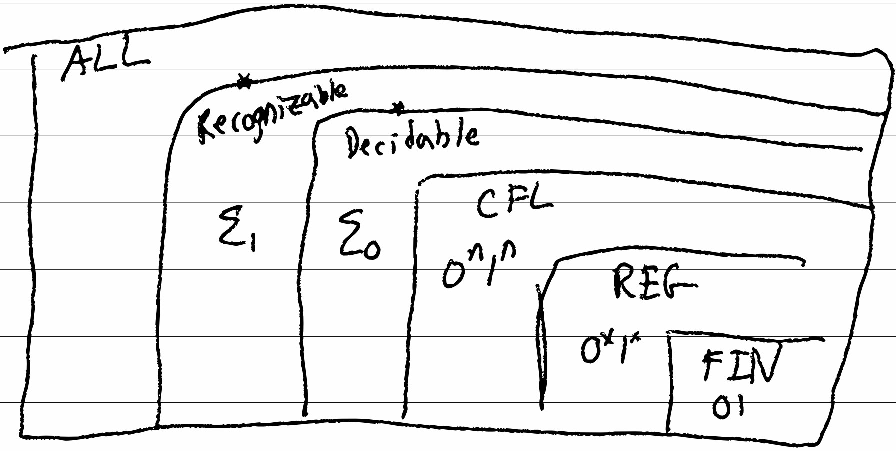
18-3/ A TM is either a

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

$$(Q, \Sigma, \Gamma, q_0, \delta : \underbrace{Q}_{\substack{\text{really } Q \\ \text{not } Q - \{q_p\}}} \times \Gamma \Rightarrow Q \times \Gamma \times \{L, R\}, q_p \in Q)$$

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

"recognizer" enumerator ⇒ Outback

"decider" enumerator ⇒ shortest-to-longest (lexicographic)

18-5/ DFA union

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

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

$\Rightarrow$

$(\overbrace{Q_x \times Q_y}^{\text{Pair}<Q_x, Q_y>}, \Sigma$

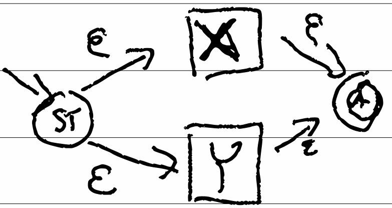
$(q_{0x}, q_{0y}),$

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

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

NFA union

REG ∪
$\hookrightarrow$ NFA



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

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

$Q = \{START, ACC\}$
$\cup \{0\} \times X$
$\cup \{1\} \times Y$

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

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

$\delta((0, q_x), c) = \{0\} \times \delta_x(q_x, c) \cup$
(if $q_x \in F_x$ then $\{Acc\}$ if $c = \varepsilon$)

interface NFAUnionState <X, Y>

NUS_START ()

= < nat → X or Y >

NUS_ACC ()

NUS_FromX (X)

NUS_FromY (Y)

19-1/ Closure properties

A set A

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

"A is closed under f"

The regular languages are closed under

C, $\cup$, $\cap$, $\circ$, $*$

The CFLs

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

What are Σ_0 (decidable)

Σ_1 (recognizable) languages

closed under?

Is Σ_0 closed under complement? ✓

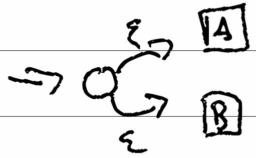
Σ_1

19-2/ Union

DFAs $Q = Q_A \times Q_B$

(did both at same time)

NFAs



TMs — can't do NFA (deterministic)

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

$Q \rightarrow Q_A \times Q_B$ tape ⟶ two tapes

TMs — simple (like assembly) DFAs

↑ ↑ ↑

MTMs C NFA

↑

↑

↑

↑

x86

19-3/ Compiler correctness

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

C, assembly — asm, binary

NFA — DFA

bi-simulation

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

$$\begin{tikzcd}
A \arrow[r] \arrow[dr, "m"'] & B \arrow[d, "m"] \\
& M
\end{tikzcd}$$

Galois connection

19-4) TM w/ "stay"

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

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

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

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

$\Rightarrow$

$(q_K, c, L/R)$

where $\delta(q_j, b) = ?$

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

$u[q_i]av \rightsquigarrow$

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

$\swarrow$

$u[q_j]bv$

$\forall x \in \Gamma$

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

19-5/ MTMs (exist to do $\cup / \cap$)

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

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

$\delta_{A \cup B}((q_a, q_b), (t_a, t_b)) =$

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

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

$$\frac{\delta(q_i, a, \alpha) = (q_j, (b, L), (\beta, R))}{\begin{array}{rcl} uc & \multirow{2}{*}{$[q_i]$} & av \\ x & & \alpha y \end{array} \Rightarrow \begin{array}{rcl} u & \multirow{2}{*}{$[q_j]$} & cbv \\ x\beta & & y \end{array}}$$

19-6/

$$\begin{tikzcd} \text{MTM config} \arrow[r, "\text{compile}"] \arrow[d, "\text{eval step} \Rightarrow"'] & \text{STM config} \arrow[d, Rightarrow] \\ \text{new config} & \text{new S config} \end{tikzcd}$$

$$\Gamma = \{\#\} \cup \Gamma_A \cup (\Gamma_A \times \{X\}) \cup \Gamma_B \cup (\Gamma_B \times \{X\})$$

$$\begin{tikzcd} \begin{matrix} u\,c & [q_i] & a\,v \\ x & & \alpha\,y \end{matrix} \arrow[r] \arrow[d, Rightarrow] & {[q_i]}\,u\,c\,\overset{X}{a}\,v\,\#\,x\,\overset{X}{\alpha}\,y \arrow[d] \\ \begin{matrix} u & [q_j] & c\,b\,v \\ x\,B & & \gamma\,y \end{matrix} & {[q_j]}\,u\,\overset{X}{c}\,b\,v\,\#\,x\,B\,\overset{X}{\gamma}\,y \arrow[l] \end{tikzcd}$$

(q_i, a) $\quad$ (q_i, a, α)

$(q_i, (b, L))$ $\quad$ $(q_j, (B, L), (B, R))$

(q_j)

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

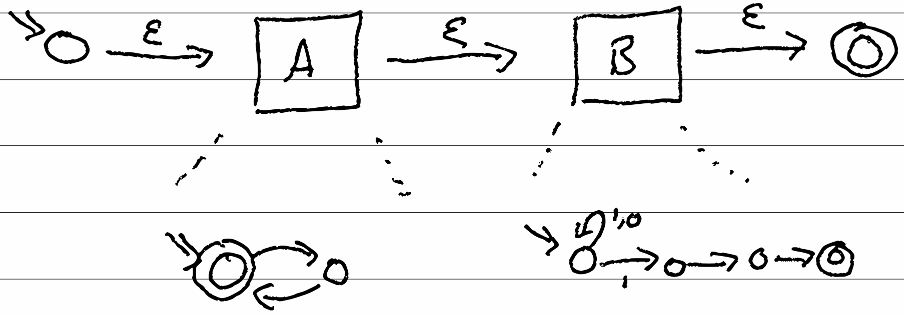
Σ_1 ✓

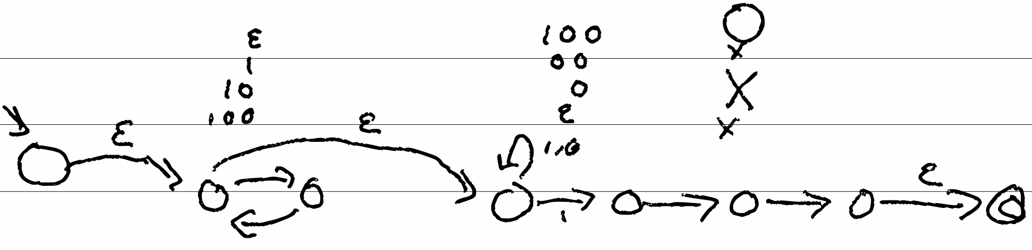
2041) Σ_0 : $\subset, \cap, \cup, \quad \circ \quad , *$

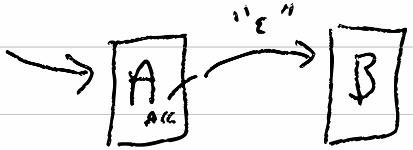
Σ_1 : $\square, \cap, \cup,$

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

f(g(x)) is comp







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

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

$\overset{\varepsilon}{0\ 11\#}\ 011\ 001$ 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 ~~[illegible]~~ try next)

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

ret ACC if both ACC

ret REJ

ε abc – REJ

a bc – M.LOOPS

ab c – N.LOOPS

abc ε – ACC

Σ_0 : $0 \rightarrow$ ✓

Σ_1 : $0 \rightarrow$ ✓

20-3/ Non-deterministic Turing Machine

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

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

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

(L)
(W)
(R)

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

$\Rightarrow$: rel(config)

P(config, config)

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

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

$=$: rel (num-sentence)

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

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

"oracle" -model

⟸ ⟸ ⟵

reject if

$\Rightarrow^* [\ ]$

20-4 | Forking semantics

acc if

$\Rightarrow^* [``x[q_a]y'', c_2 \ldots c_n]$

F-config = Sequence of config

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

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

$$\frac{}{[``x[q_r]y'', c_2 \ldots c_n] \Rightarrow [c_2 \ldots c_n]}$$

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

$$\begin{tikzcd}
\text{Fconfig}_0 \arrow[r, Rightarrow] \arrow[d, "\text{run}"] & \text{config} \arrow[d, dashed] \\
\text{Fconfig}_1 & \text{config} \arrow[l, Leftarrow]
\end{tikzcd}$$

$[c_1, \ldots c_n]$

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

$\overset{\wedge}{x[q_i]y} \Rightarrow x \overset{\nearrow Q}{q_i} y$

$\Gamma' = \Gamma \cup Q$

$$\begin{tikzcd}
O \arrow[r] & ND \arrow[r] & 2ND \arrow[r] \arrow[rd] & BT \\
 & & & \text{Forking} \arrow[r] & \checkmark
\end{tikzcd}$$

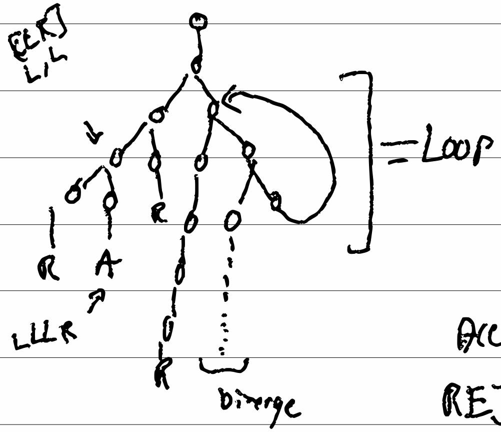
20-5

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 $2^k + 2^{k-1} + 2^{k-2} + \dots = \sum_{i=0}^{k} 2^i$

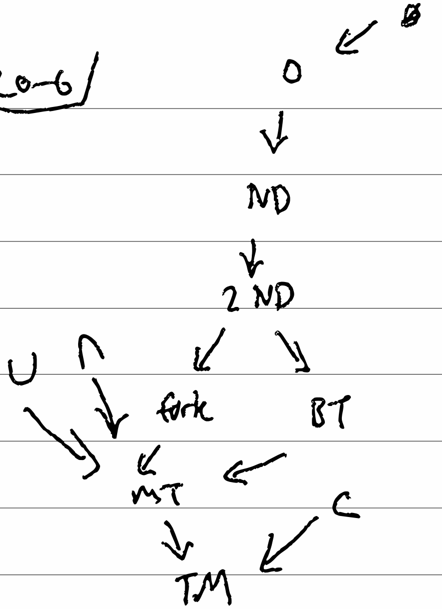
$k2^k$

tape 0: w

tape 1: current map → add 1

tape 2: simulation tape

20-6/



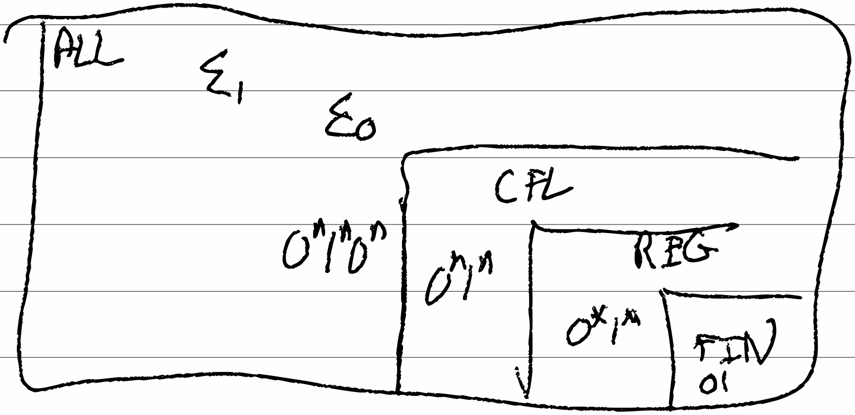
$w \in A^*$ iff

$w = w_1 \ldots w_n$

where $w_i \in A$

21-1/ Σ_0 (decidable - Y/N)   c, ∪, ∩, ∘, *

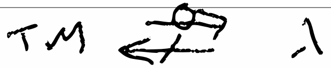
Σ_1 (recognizable - Y/N/L) □ ∪, ∩, ∘, *

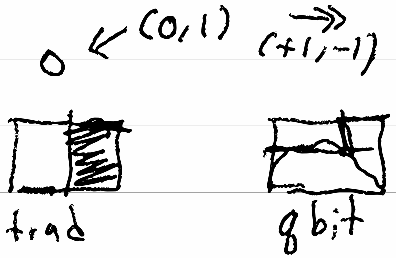


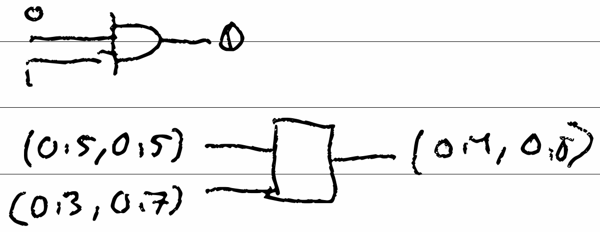
Church-Turing Thesis

"Algorithm means Turing Machine or
λ-calculus term"

Human = Math₁ = Math₂







21-2/ Humans counting means $N(0, +1)$

Love means OkCupid / Tinder Match.com

Movie taste means Netflix advice

$$\begin{tikzcd}
\text{Algorithm} & = & \text{TM} \arrow[rr, shift left] \arrow[d, leftrightarrow] & & \lambda\text{-calculus} \arrow[ll, shift left] \arrow[d, leftrightarrow] & \\
& & \text{ND} \arrow[d, leftrightarrow] & & \text{JSWIM} \arrow[d, leftrightarrow] & \\
& & \text{Stack} \arrow[d, leftrightarrow] & & \text{Algol} \arrow[d, leftrightarrow] & \\
& & \text{Reg} \arrow[d, leftrightarrow] & & \text{C} \arrow[d] & \\
& & \text{x86} \arrow[d, leftrightarrow] & & \text{Java} \arrow[d] & \\
& & \text{Jay's phone} & & \text{Racket} \arrow[r, Leftrightarrow] & \text{"Software"} \\
& & \text{"Hardware"} & & &
\end{tikzcd}$$

21-3/ 10th : "Devise an effective method to find the integer roots of any polynomial of any number of variables"

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

$dx^3 + \dots$

$ex^4 + \dots$

"one variable"

$k = 2$

$c_0 = c$ $c_1 = b$

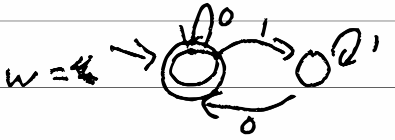
$c_2 = a$

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

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

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

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



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

$\{\}, \{\}, e, \{\}, \{\}$

010110 # ⟨Q⟩

$\langle\ \rangle \curvearrowright$ 011111000100110

DFA(REG) $\subseteq$ TM(Σ_0)

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

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

$A_{DFA} \ni \langle M, w \rangle$ 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 → Σ_0 yes ⇒ Σ_1

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

= should = Reject (Σ_0)

reject or loop (Σ_1)

$A_{TM} \in \Sigma_1$

$A_{TM} \notin \Sigma_0$

$\Sigma_1 \neq \Sigma_0$

preten $\in \Sigma_0$

Make a new TM D(<M>) =

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

Output the opposite of A_{TM}."

D<M> = accept if M doesnotaccept <M>

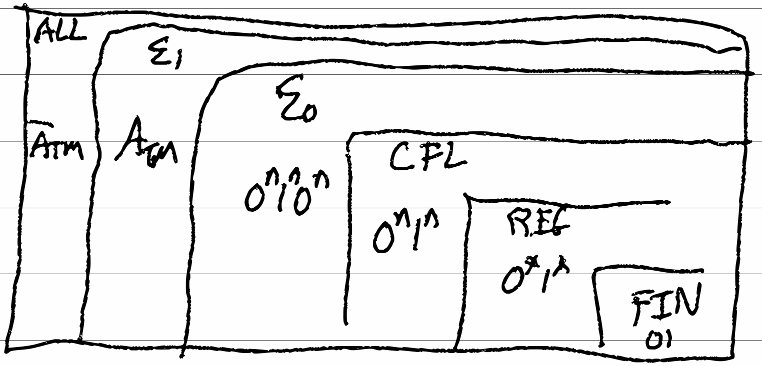
reject if M doest accept <M>

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 $x \notin A$

N or Loops o.w. $\qquad$ N or Loops o.w.

$$M(x) = \begin{cases} Y & \text{if } x \in A \\ N & \text{if } x \notin A \end{cases}$$

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

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

222/ $A_{TM} \ni \langle M, w \rangle$ 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)

$\hat{A}\mathrm{TM}$ o $0^x 1^*$

22-4/ Sizes of infinity

$\aleph_0$

$x_1, \ldots x_{\aleph_0}$

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

$\rightarrow \mathbb{R}^{\infty} = \mathbb{R} \cup \{+\infty, -\infty\}$

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

$|0^* 1^*|$

Peano

$N = 0 \mid S\ N$

$\Downarrow$

$\emptyset \quad P$

$f : X \Rightarrow Y$

one-to-one

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

$\Rightarrow a = b$

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

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

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

$\{\{\emptyset\}\},$

onto:

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

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

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

$\Rightarrow$

$|A| = |B|$

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

$|N|$

$=$

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

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

$f(a) = 2 \times a$

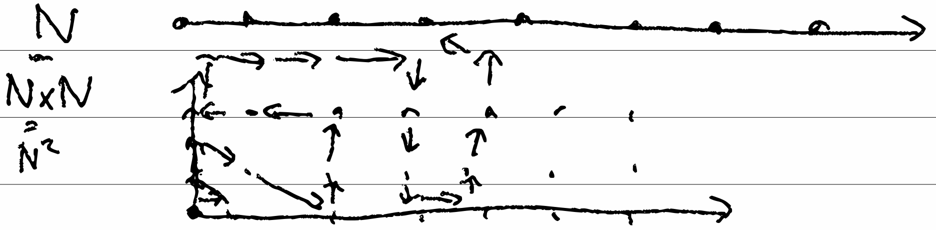
$2 \times a = 2 \times b$

$\Rightarrow a = b$

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

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

"countable"



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

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

22-6) $|N| = |N \times N|$

$|N| = |N^k|$ $(k \geq 1)$

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

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

g(before ..., last) =
cantor(f(before), last)

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

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

Fair Enumeration

22-7, $|\Sigma_0|$ and $|\Sigma_1|$ and $|\text{the TMs}| = |N|$

$|\Sigma_0| \leq |\Sigma_1| \leq |TMs| \leq |N|$

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

$\times q_a \times q_r$

$= (k : N) \times (g : N) \times \{0, \dots, k\} \times (\{0 .. k\} \times \{0 .. g\} \times$

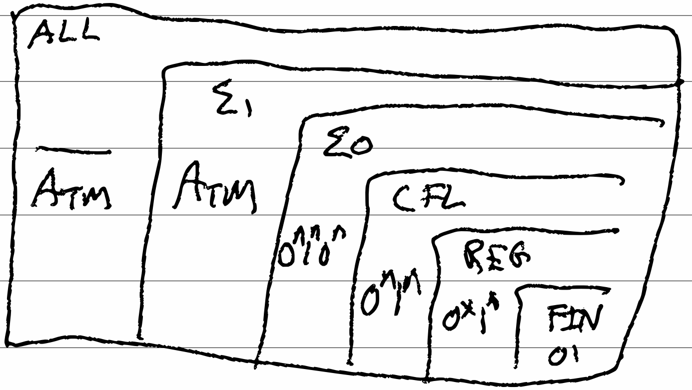
$\{0 .. k\} \times \{0 \cdots g\} \times \{L, R\})$

$\times \{0 \cdots k\} \times \{0 \dots k\}$

$\leq N \times N \times N$

$\simeq N$

23-1



$N \simeq N \times N \simeq N^k \geq TM > \Sigma \geq \Sigma_0$

<

[illegible] $R > IBS \overset{?}{\simeq} ALL$

Real numbers, R, "numbers with decimals"

1.2357 $\frac{12357}{10000} \in Q$

"weird numbers, like pi"

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

vector space of bases

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

$j = (0,1,0)$ $v = (-1,0,3)$ Cauchy sequence

$k = (0,0,1)$ $w = (-1,-1,0)$ $Q, Q, Q, Q, \dots, R$

Dedekind cut

23-2

"infinitely many digits"

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

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

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

$N \to$ digit $\{0, 1, 2, , 3$

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

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

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

$(= |N|)$

onto:

$\forall x, y.$

$f\ x = f y$

$\to x = y$

one-to-one:

$\forall x \overset{IBS}{\in} \exists y \in N$

$f\ y = x$

i	$f(i)$
0	$0,\underline{1}011001\ldots$
1	$0,0\underline{1}10111\ldots$
2	$0,01\underline{1}1110\ldots$
3	$0,111\underline{1}000\ldots$
4	$0,1010\underline{1}100\ldots$

$0.00000\ldots$

$\exists f : IBS \to N.$

onto and one-to-one

$\exists f : N \to IBS$

onto-and onetoone

claim:

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

$\forall x \in IBS.$

$\exists y \in N.$

$f\ y = x)$

$\leftrightarrow$

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

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

$\forall y \in N.$

$f\ y \neq x$

$\leftrightarrow$

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

$x(a) = \neg f(a)(a)$ ✓

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

$\exists x : N \to B.$

$\forall y \in N.$

$\exists i \in N, \quad f(y)(i) \neq x(i) \qquad i = y \qquad f(y)(y) \neq \underset{\neg f(y)(y)}{x(y)}$

23-4 IBS is the same size as ALL

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

$$ALL = P(\Sigma^*)$$

$$= P(\{\overset{0}{\varepsilon}, \overset{1}{0}, \overset{2}{1}, \overset{3}{00}, \overset{4}{01}, \overset{5}{10}, \overset{6}{11}, \overset{7}{000}, \overset{8}{001}, \ldots\})$$

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

$$\overset{\{0,1\}}{\{\varepsilon, 0\}}, \{\varepsilon, 011001\}, \ldots$$

$$\overset{\{0,2,4,6,8,\ldots\}}{\{\varepsilon, 1, 01, 11, 001, \ldots\}}$$

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

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

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

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

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

$f :$ ~~$IBS \to ALL$~~ $\wedge$ onto(f) $\wedge$ one-to-one(f)

$$f(A)(i) = \text{lexi}(i) \in A$$

$$ALL \approx IBS$$

<u>23-5/</u>	IBS	$>$	N		
	$\geq$		$=$		
	$R[0,1)$		$N \times N$		
	[illegible]		$=$		
	R		N^k		
	$=$		$\geq$		
	ALL		TM		
	$\aleph_1$		$\geq$		
			Σ_1	$=$	$\aleph_0$
			$\geq$		
			Σ_0		

23-6/ Reducibility

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

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

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

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

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

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

$A_{TM} \le_m E_{TM}$ $\exists f.\ \langle M, w \rangle \longrightarrow \langle M' \rangle$

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

$M'(x)$ = rum M on w if yes → 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 $L(M_1) = L(M_2)$

$E_{TM} \geq_m EQ_{TM}$

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

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

$M_1 = M$ $M_2(x) = \text{reject}$