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

$\forall x, x \notin \emptyset$

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

$\{ \checkmark, ☺ \}$

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

$\forall x,\ x \in \{y\}$ 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 remade?"

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

$\overset{0}{\downarrow}\overset{1}{\downarrow}\overset{2}{\downarrow}$

$\underbrace{\text{"0100001"}}_{\text{length} = 7}$ = 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())));
```

$\{\Sigma\}$

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

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

" $= B$

newList<chars>

{ Zero, One};

0	1	2	3
ε	0,	00,	000
	1,	01,	001
		10,	010
		11,	011,
			100
			101
			110
			111
2^0	2^1	2^2	2^3

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

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

lexi B 0 = ε

lexi B 1 = 0

lexi B 2 = 1

lexi B 6 = 1[illegible]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 $\forall y \in X.\ y \in A$)

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

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

$\{0\} \in$ " $\{2, 3\} \in$ $\{0, 2, 3\}$ $\downarrow$ $\underline{0}12\underline{3}$

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

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

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

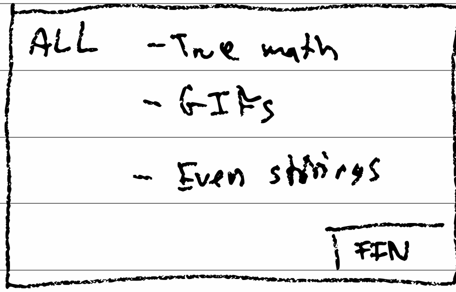
all even lengthed strings $\in P(\Sigma^*)$ $= \{\varepsilon, 00, 11, 01,\}$

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



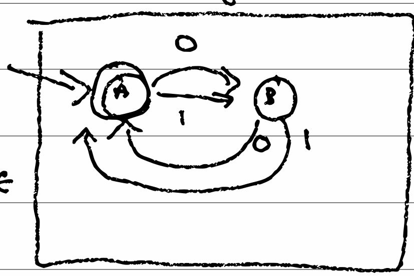
$\emptyset \in FIN$

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

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

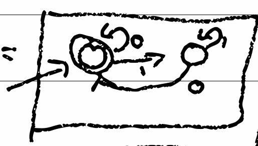
All even strings =

$x \in$



ε 01 ~~111~~

even nums =



DFA - a deterministic finite automata

○ - states $\{A, B\}$

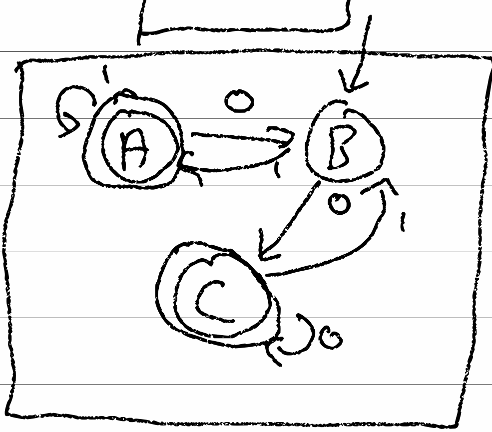
→○ - start state A

◎ - 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

states, alphabet, start, accepting

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

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

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

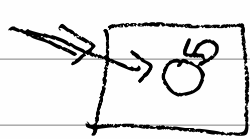
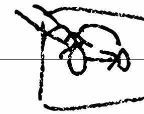
$[q_i]\,x\,y \to [q_j]\,y$ iff $\delta(q_i, x) = q_j$

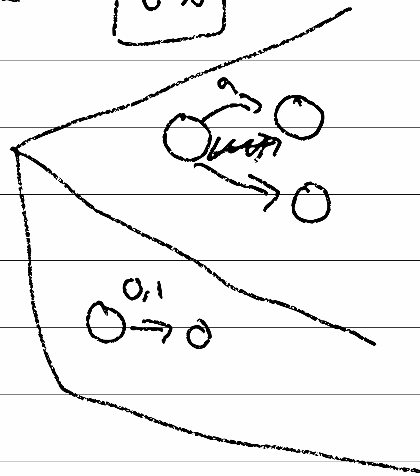
$x \in DFA$ iff $[q_0]\,x \to \to \to \to \to [q_f]\,\varepsilon$
and $q_f \in F$

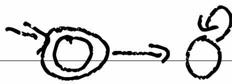
$0110 \in$ EvenLen iff $[A]0110 \to [B]110 \to [A]10$
$\to [B]0 \to [A]\,\varepsilon$ $A \in \{A\}$ ✓

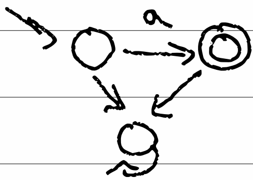
```
class DFA {
  ... Q, Σ, F, q0, δ ...
  public bool accepts (String x) {
    State qi = q0;
    while (!x.empty()) {
      qi = δ(qi, x.first());
      x = x.rest(); }
    return F.in(qi); } }
```

2-4

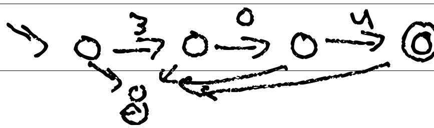
 $=$ $\emptyset$ $=$ 



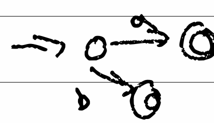
 $= \{\varepsilon\}$

 $= \{a\}$

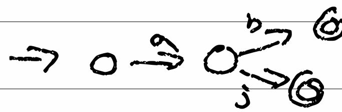
$=$ 

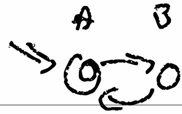
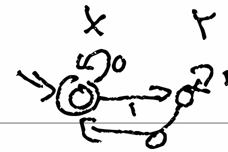
$\{304\} =$ 

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

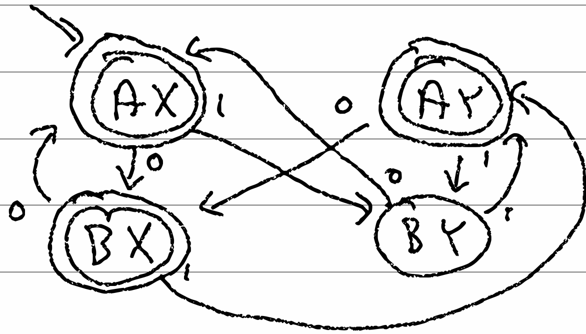
$\{a\} \cup \{b\}$ 

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



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

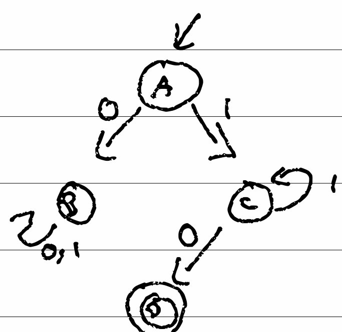
Algorithm for $X \subseteq Y$ if X & Y are DFAs

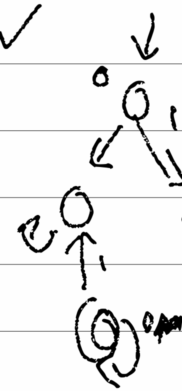
3-11 DFA $\Rightarrow$ example or false

DFA:

- Q: State $\to$ Bool
- Σ: list of characters
- q_0: state
- δ: (state $\times$ (char)) $\to$ State
- F: State $\to$ Bool

10 ✓





Q	$\emptyset$	q_0	
$\{A, B, C, D\}$	$\{\}$	$[A]$	$A \Rightarrow \varepsilon$
$\{B, C, D\}$	$\{A\}$	$[B, C]$	$B \Rightarrow A, 0$; $C \Rightarrow A, 1$
$\{C, D\}$	$\{A, B\}$	$[C]$	Yes, it's possible.'
$\{D\}$	$\{A, D, C\}$	$[D]$	$D \Rightarrow C, 0$
$\{\}$	$\{A, B, C, D\}$	$[\,]$	or if not, No!

10

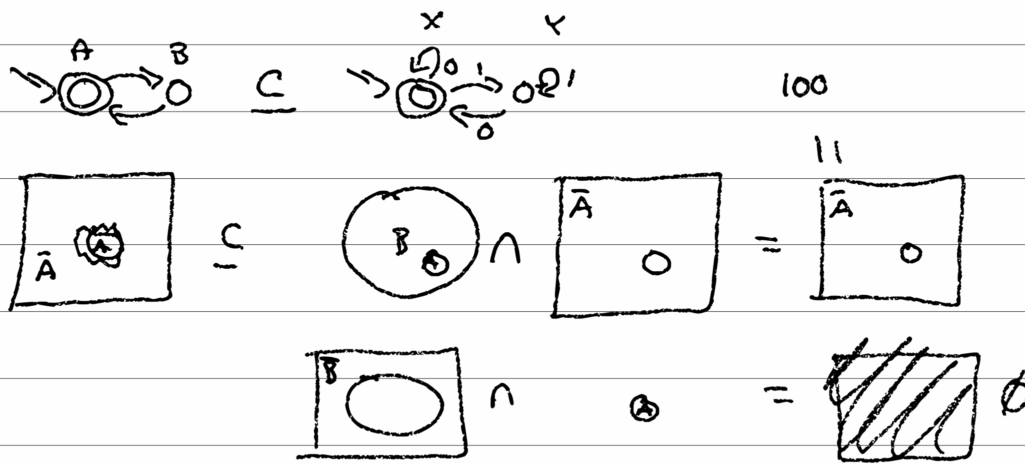
3-2/ subset

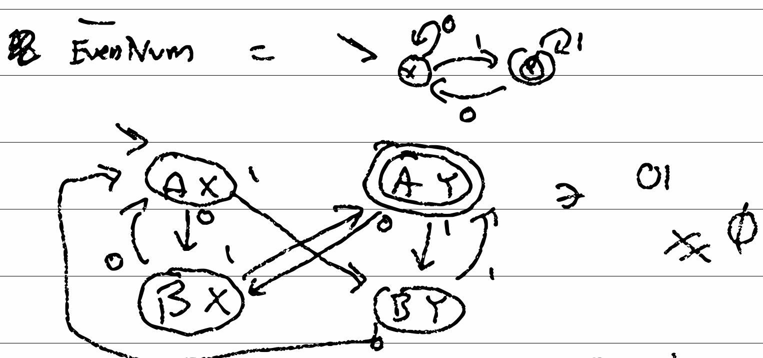
$A \subseteq B$ iff $\forall x \in U.\ x \in A \Rightarrow x \in B$

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

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

→ finite means naive works!





model checking

Soundness : model $\subseteq$ theory

completeness : theory $\subseteq$ model

model = theory

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

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

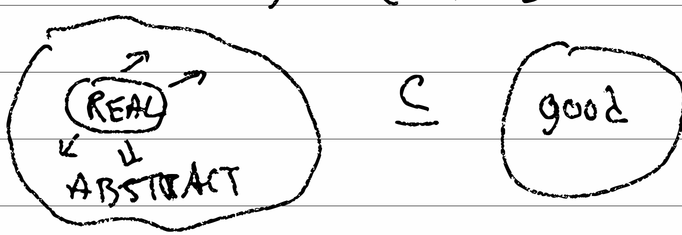
if $x > 0$ then

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

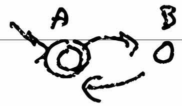
$abcd = a \circ b \circ c \circ d$

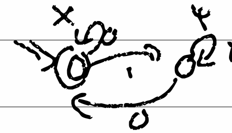
$\{j\} \circ \{m, n\} = \{jm, jn\}$

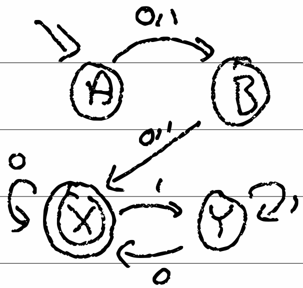
$x \in A^*$ iff $x = x_0 \circ x_1 \circ \dots \circ x_n$ for $n \in \mathbb{N}$
and $x_i \in A$

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

3-5/ Even Len $\circ$ Even Num





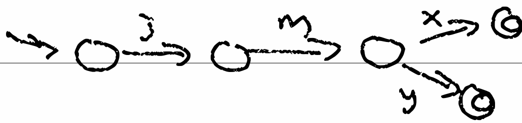


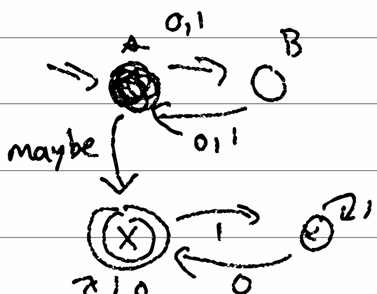
$\underbrace{00}\underbrace{110}$ ✓

0011 ×

~~[illegible]~~

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

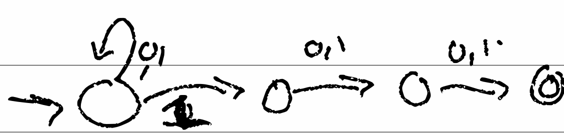




$\Sigma = \{0,1\}$

$x \in$ 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

$\varepsilon \quad \Sigma \quad e \quad \varepsilon$

4-1

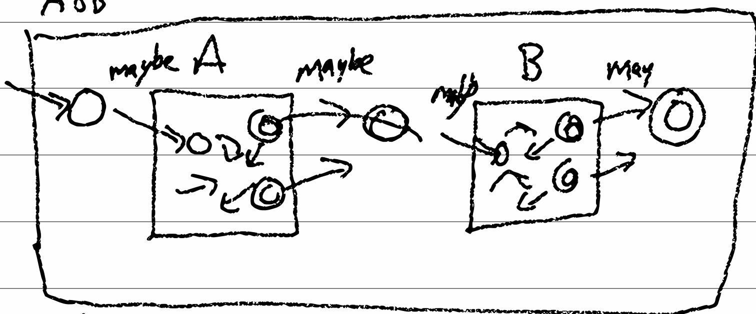
$A \circ B \in DFA$ iff $A \in DFA$

A^* $\quad$ $B \in DFA$

NFA (N - non D- deterministic)

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

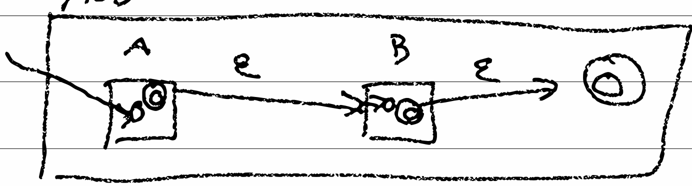
A∘B



maybe written as "ε"

What does (NFA) this mean?

A∘B



NFA ⟺ DFA

4-2 | What do NFAs mean?

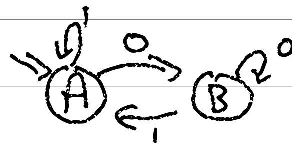
A DFA represents a set and
a set "is" a membership function

$U \to \text{Bool}$

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

config $= \Sigma^* \times Q$

$\Sigma^* \to Q^*$



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

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

Σ_e ~~make~~ $= \Sigma \cup \{\varepsilon\}$

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

$0A\ 1A\ 1A\ 0B \to N$

DFA:

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

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

Nvalid q_i ε $= Y$

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

An Oracle

4-3

$$\begin{tikzcd} \arrow[r] & A \arrow[loop above, "0,1"] \arrow[r, "1"'] & B \arrow[r, "0,1"] & C \arrow[r, "0,1"] & \boxed{D} \end{tikzcd}$$

"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

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

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

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

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

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

valid q_0 t = Y

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

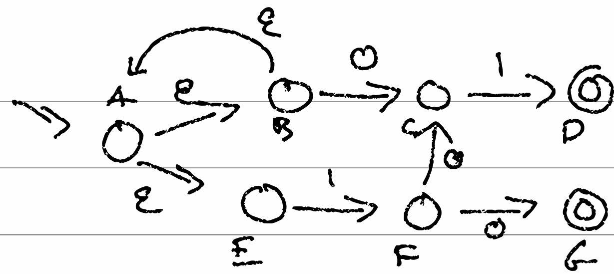
$\text{NFA-accepts} : \Sigma^* \to Y/N$

figure all possible traces

check if valid and is strings match

check if last is in $\in F$

4-4



$(\varepsilon, B)(0, C)(1, D)$ $\qquad$ $01 = \varepsilon 0 0 0 1$

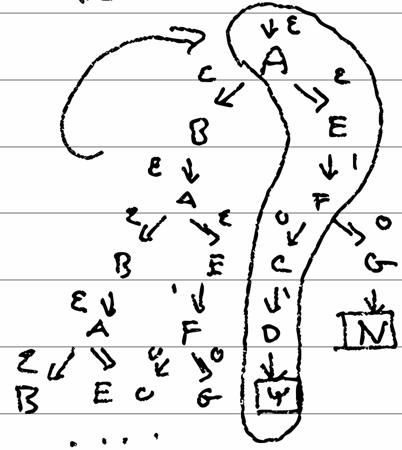
$(\varepsilon, E)(1, F)(0, C)(1, D)$ $\qquad$ $101 = \varepsilon 0 1 0 0 0 1$

$(\varepsilon, E)(1, F)(0, G)$ $\qquad$ $10 = \varepsilon 0 1 0 0$

$(\varepsilon, B)(\varepsilon, A)\ x$ where x is valid

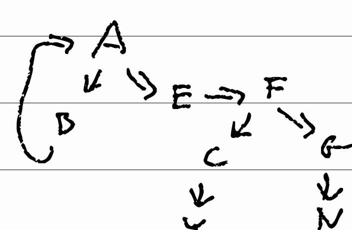
$\Rightarrow$ valid

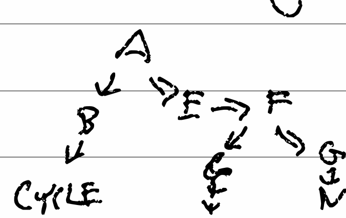
Trace Tree = Y | N | Branch (Σ, Q) (List TT)



101

$\varepsilon, E \quad 1, F \quad 0C \quad 1D$



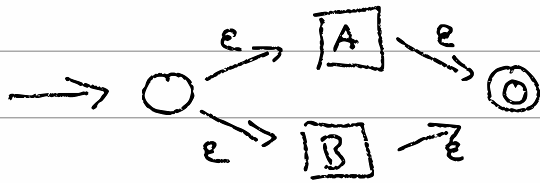


Forking model of NFAs (make TT)

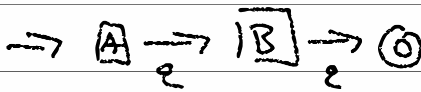
Back-tracking model (explores TT)

4-5/

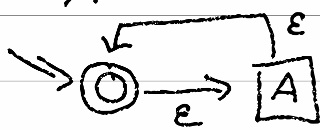
$A \cup B$

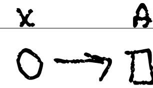


$A \circ B$



A^*





State X transitions to THE start state of A

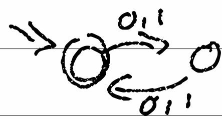
All accepting states of A transition to X

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

$\Rightarrow$

$\Leftarrow$

DFA $\Rightarrow$ NFA /



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

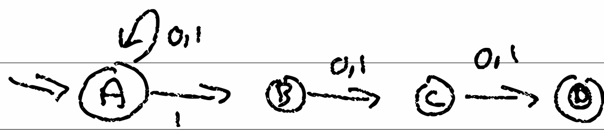
NFA $\delta': Q \times \Sigma_\varepsilon \to P(Q)$

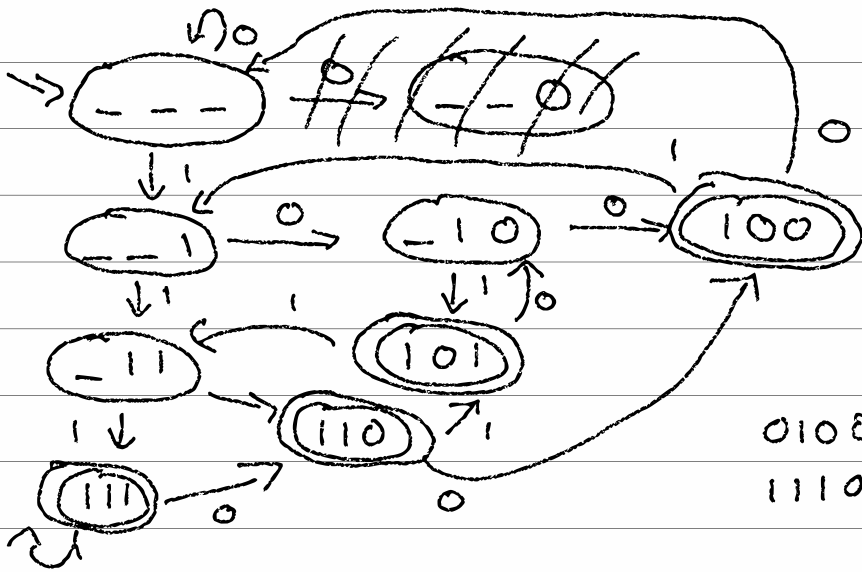
$$\delta'(q_i, \varepsilon) = \emptyset$$

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

NFA $\Rightarrow$ DFA/

3rd is 1





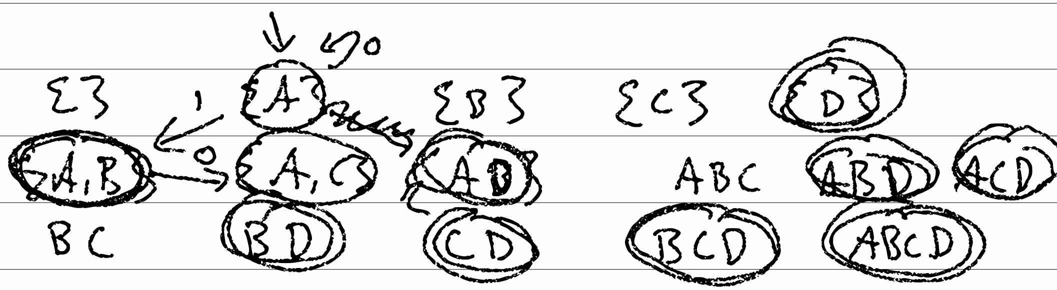
0100

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/ $A \cup B$

$\delta_A : Q_A \times \Sigma \Rightarrow Q_A$

$\delta_B : Q_B \times \Sigma \Rightarrow Q_B$

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

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

char

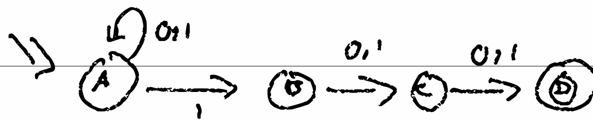
(p, c) => new Pair(deltaa(p.fst, c),
deltab(p.snd, c));

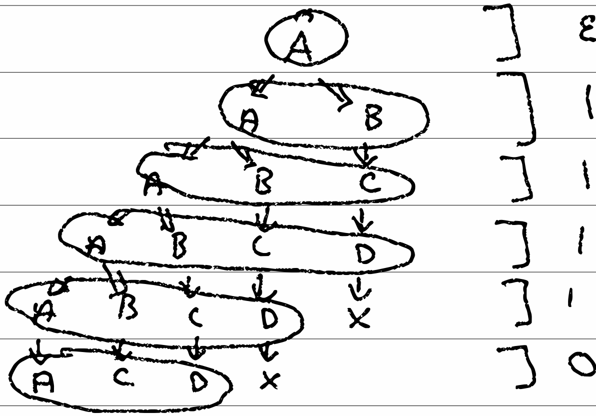
pair<State, State>

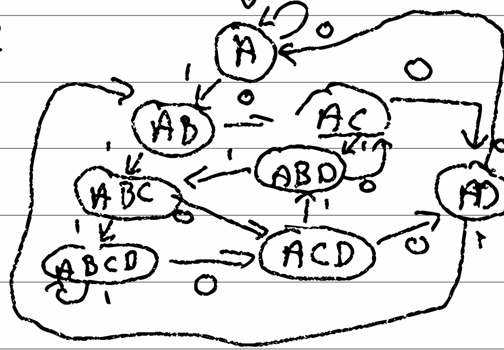
fst snd

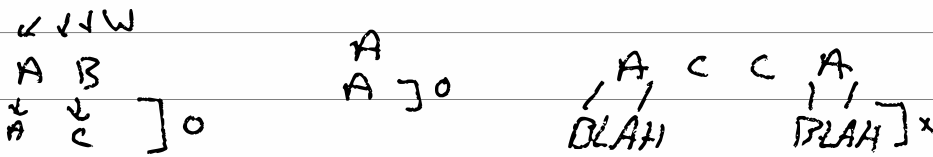
5-2/ NFA $\Rightarrow$ DFA

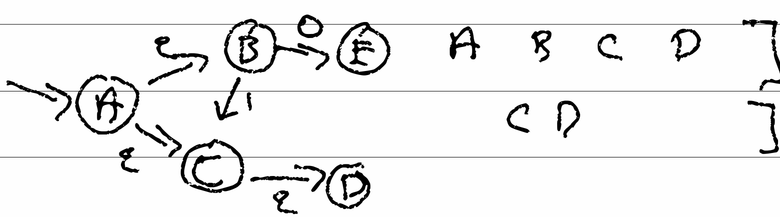
$(Q, \Sigma, q_0 \in Q, \delta : Q \times \Sigma_\varepsilon \to P(Q)), F \subseteq Q) \Rightarrow (Q', \Sigma, q_0' \in Q', \delta' : Q' \times \Sigma \to Q', F' \subseteq Q')$











5-3

$E : Q' \to Q'$ — 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 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

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

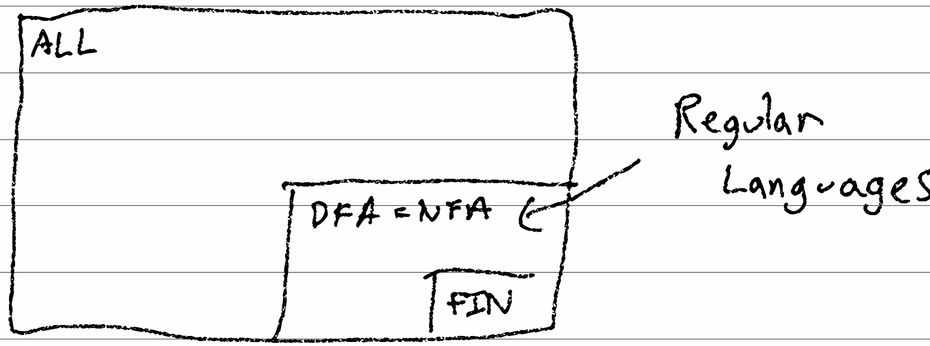
$$E^*(A) = A \cup \bigcup_{q_i \in A} \delta(q_i, \varepsilon)$$

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

$N = D$

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

$D = N$

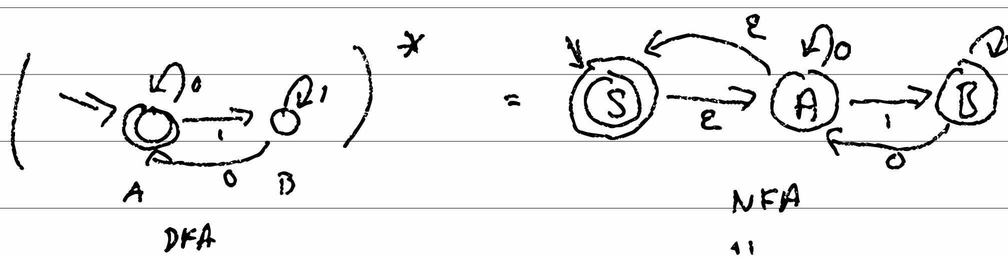


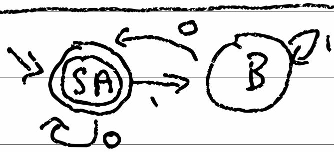
Program f ....

program g ....

"f ; g"

compositional



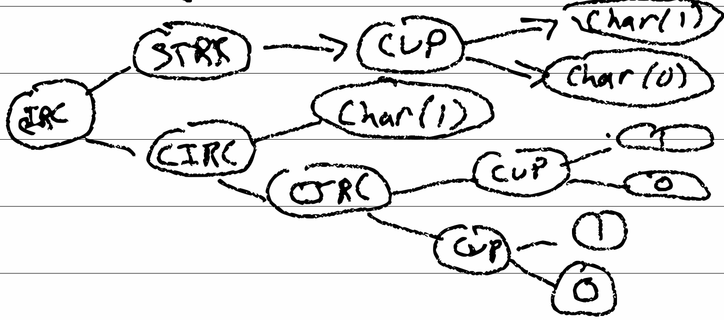


6-2 / Regular Expressions

re ::=	ε	| EMPTY
	$\emptyset$	| NULL
	c	| Char
	re $\cup$ re	| CUP
	$\star$ 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)
```

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



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

6-3 | L : RE → ALL = $P(\Sigma^*)$

(ie the language it describes)

(or some.)

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

$L(\emptyset) = \emptyset$

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

R NAME   OR

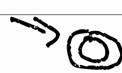
$L(r \cup r') = L(r) \cup L(r')$

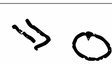
$L(r \circ r') = L(r) \circ L(r')$

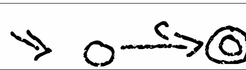
$L(r^*) = L(r)^*$

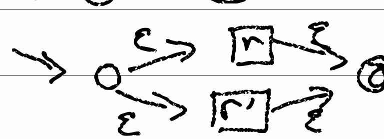
```
class Cup { L() {
        return union(lhs.L(), rhs.L())}}}
```

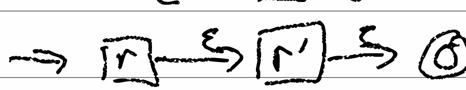
compile : RE → NFA

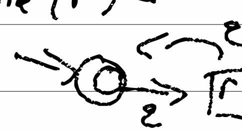
compile (ε) = 

compile ($\emptyset$) = 

compile (c) = 

compile ($r \cup r'$) = 

compile ($r \circ r'$) = 

compile (r^*) = 

6-4)

$" \ " = \varepsilon$

$\underline{\qquad} = \emptyset$

$"c" = c$

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

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

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

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

$"(a|b|c)" = \text{?} \quad (a, b, c, \in \Sigma^*)$

[012]

(zero | one | ~~two~~)

$\bullet = \Sigma$

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

6-5

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

$gen\ \varepsilon = \varepsilon$

$gen\ \emptyset = FALSE$

$gen\ c = `c'$

flip coin

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

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

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

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

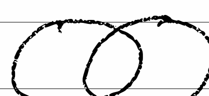
", m;s"

$equal : RE \times RE \to Bool$

$equal\ x\ y =$

NFA2DFA(compile x)

NFA2DFA(compile y) — dfa equality?



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

6-6

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

$x \bullet 1 = x$

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

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

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

6-7

$$\text{size} : RE \to Nat$$

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

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

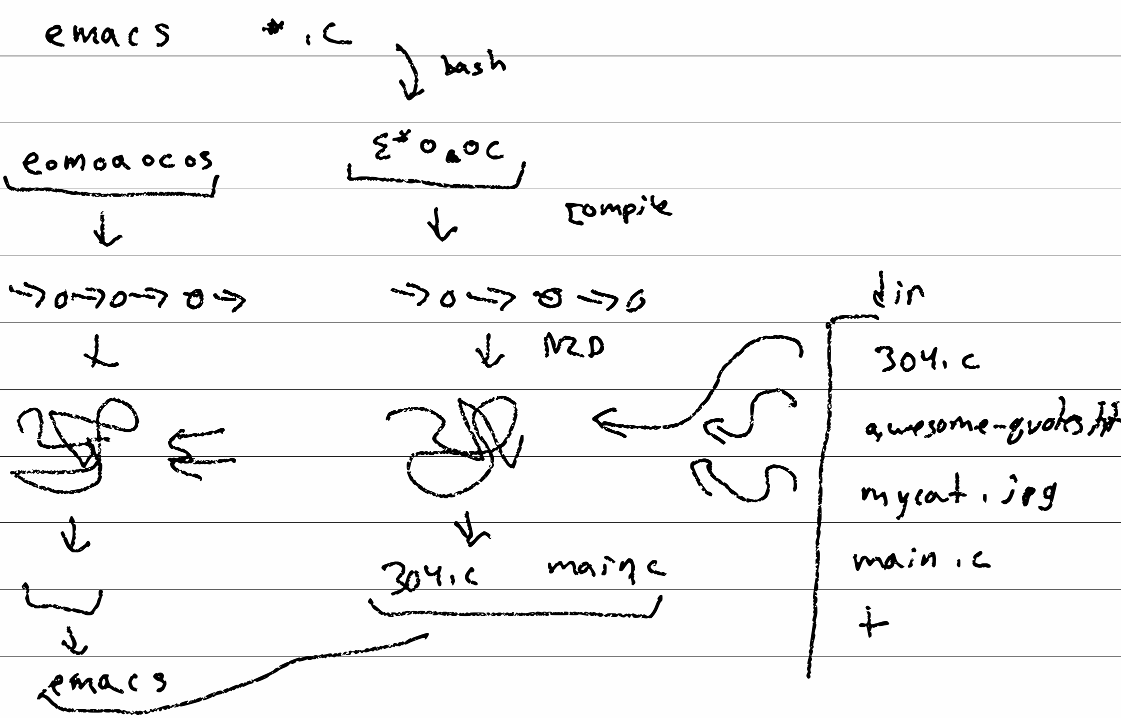
$$\text{size } c = 1$$

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

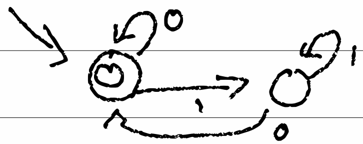
$$\text{size } (x \circ y) = sz\ x + sz\ y + 1$$

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

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

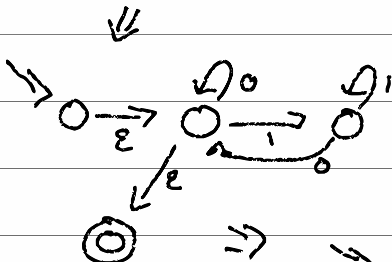


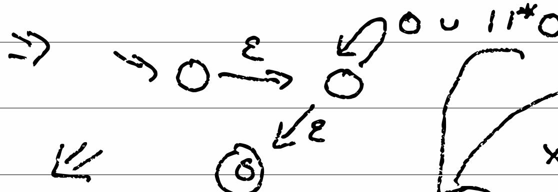
6-8 decompile : DFA (NFA or) $\rightarrow$ RE

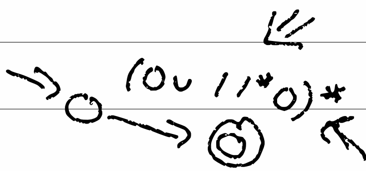


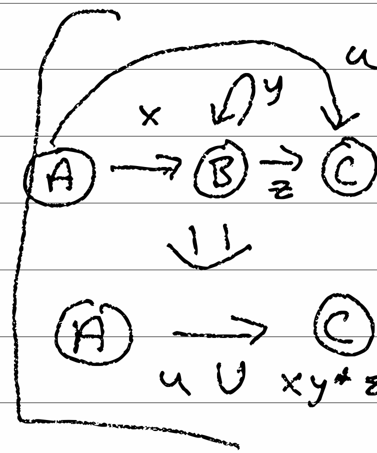
$\Sigma^* \circ 0 \cup \varepsilon$

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









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

$\varepsilon \quad 0 \quad 10 \quad 111110$

$01110 \quad 1111100$

$\underbrace{111}\underbrace{0}\underbrace{0}\underbrace{0}\underbrace{1110}\underbrace{110}\underbrace{0}\underbrace{10}\underbrace{10}\underbrace{10}$

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

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

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

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

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

OUT: 2-GNFA $\rightarrow$ RE

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

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

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

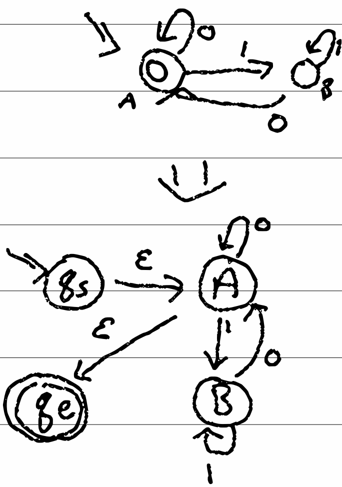
"Find r"

n- (n+2)

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/

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

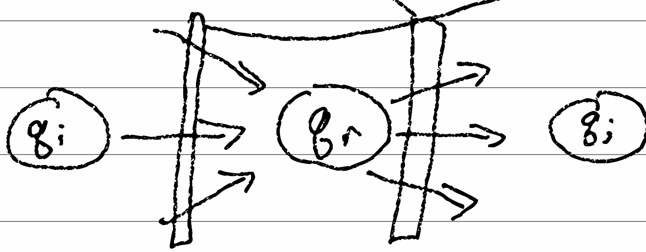
RIP = (n+1)-GNFA → n-GNFA

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

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

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

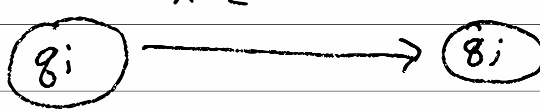
$\Delta : \underbrace{(Q - q_e)}_{q_r \in} \times \underbrace{(Q - q_s)}_{q_r \in} \longrightarrow RE$



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

$\Delta'(q_i, q_j) =$

$\cancel{x \circ z}$

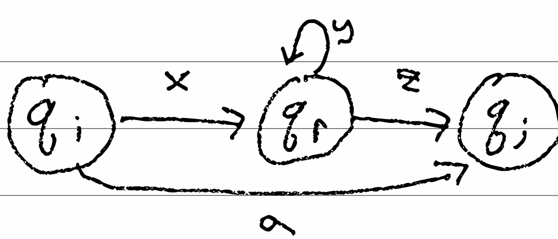


$x = \Delta(q_i, q_r) \circ$

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

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

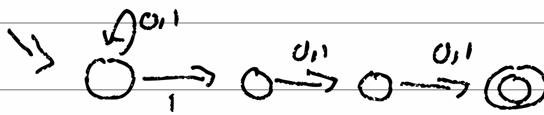
$a = \Delta(q_i, q_j)$



7-4/

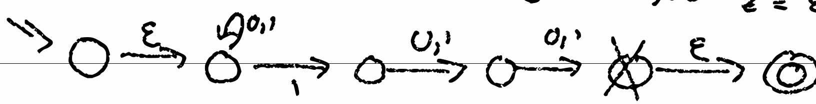
```
int main() {
    int x = 8;
    int y = f(x, z);
    ....
    int y = x + 2 * z;
```

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



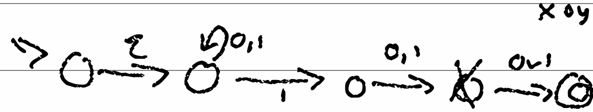
IN

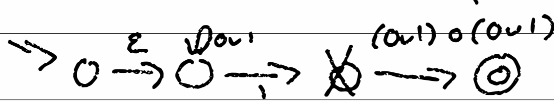
$0 \cup 1 = x \quad y = \emptyset \quad z = \varepsilon \quad a = \emptyset$

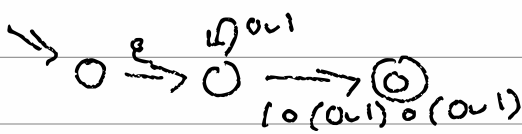


$x \circ y^* \circ z \cup a = (0 \cup 1) \circ \emptyset^* \circ \varepsilon \cup \emptyset$

$= (0 \cup 1)$

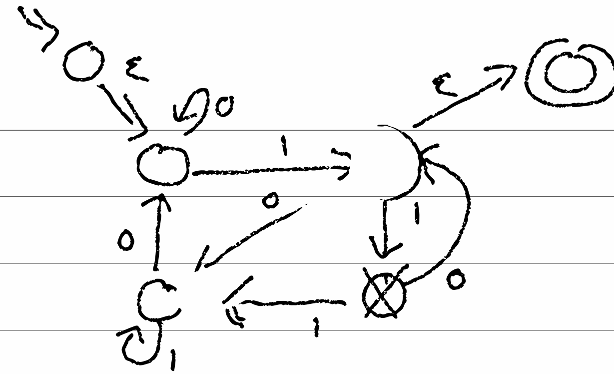


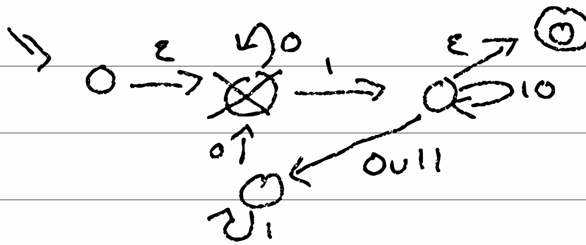


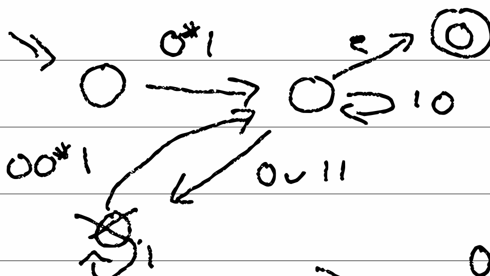


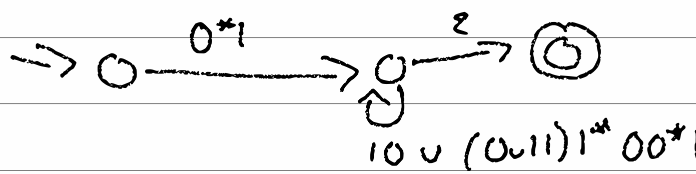
$\varepsilon \circ (0 \cup 1)^* \circ 1 \circ (0 \cup 1) \circ (0 \cup 1) = \Sigma^* 1 \Sigma \Sigma$

7-5









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

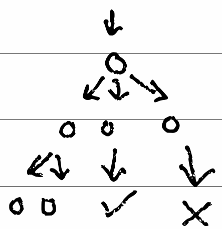
8-1 ε

Object — Char(char c)

— Epsilon()

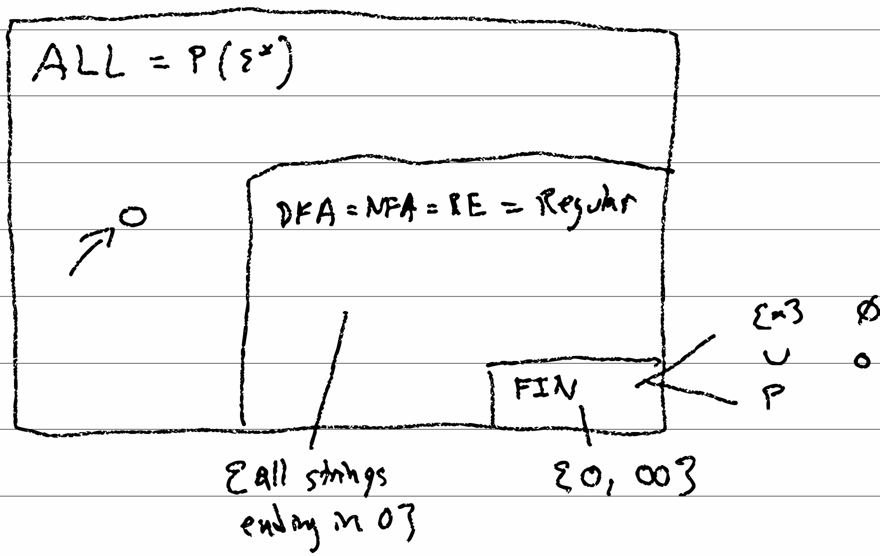
UTF-8

TT



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

$$\text{List(TT)} \to \text{Char} \times \text{TT}$$



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

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

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

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

8-2 How can know stuff about infinite sets?

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

$P : DFA \to Prop$

$\Rightarrow \quad \overbrace{q_0 \in Q}$

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

$\neg P(B)$ (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 $\exists B \in \cancel{ALL},\ \neg P(B)$

---

conclude $B \notin REG$.

$\Sigma = \ . \ . \ . \ .$

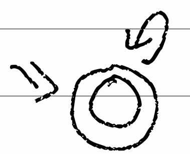
8-31

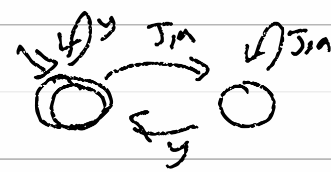
"Jay" $\in d$ $\quad \Sigma \supseteq \{J, a, y\}$

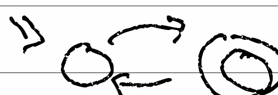
$\Rightarrow q_0 \xrightarrow{J} \xrightarrow{a} \xrightarrow{y} q_f \in F$

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

$|d.Q| = 1$







Daphne wins

pick a number of states ; 4

she picks a char

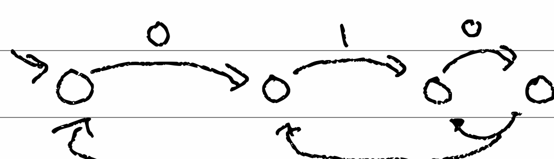
I say what state we go to

I win if I never say same state

She wins if I repeat

How many turns to win?

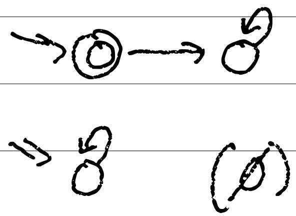
Y ⇒ N ⇒ $|Q|$

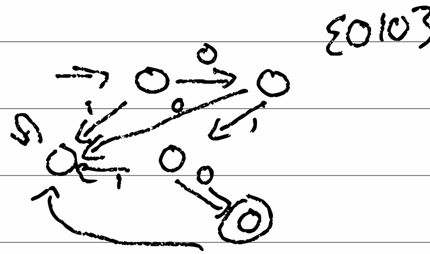


δ-y/ $\delta : Q \times \Sigma \to Q$ Total Fun

from   0, 1

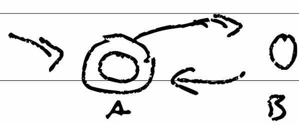
All DFAs have a loop





Some have "exciting" loops

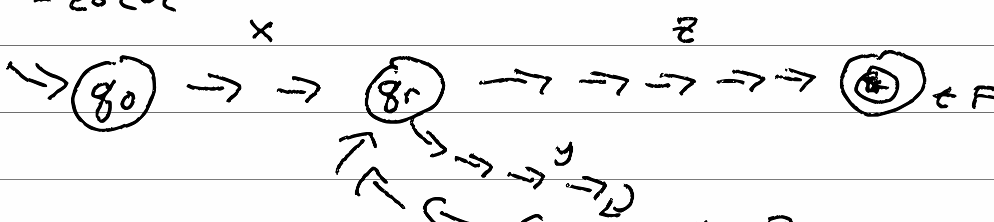
$\exists\, s \in L(\text{DFA})$. $s =$ ....

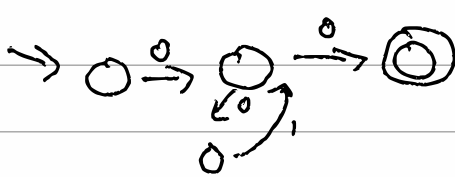


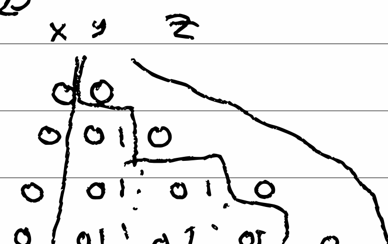
(Even lengthed)

$\varepsilon = A$ $0110 = ABABA = \varepsilon\, 0\, 01\, 0\, 10$

$= \varepsilon 0 \varepsilon 0 \varepsilon$







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

They are all finite

$L(m) \in FIN$

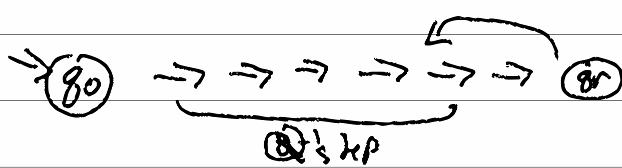
If it does have one ...

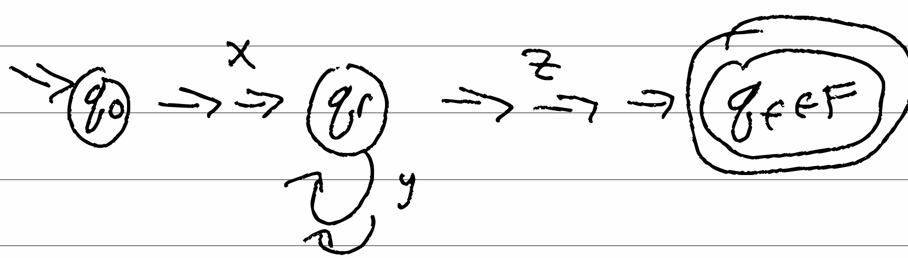
then it must be infinite (regular)

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

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

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

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

$\wedge |xy| \leq p$

$\wedge |y| > 0)$

$\forall i \in N.$

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

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

$\neg RPP(B) :=$

$\forall p \in N.$

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

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

$\exists i \in N.$

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

8-7) Need: an infinite space problem

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

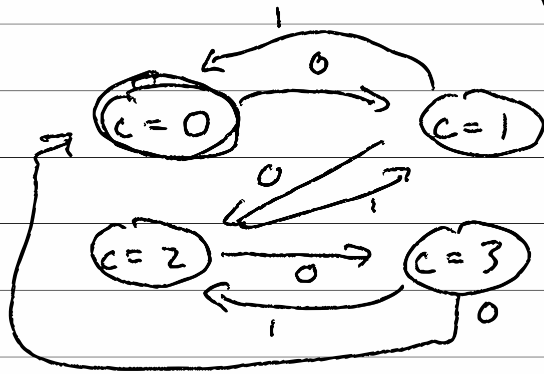
```
if (getc() != '1') { ret false }
return getc() == EOF;
```

$= 2^{32}$ 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

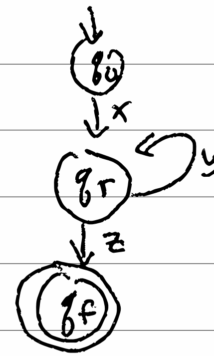
$\exists p \in \mathbb{N}.$ // $p = |Q|$

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

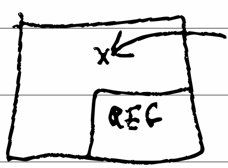
$\exists (x, y, z \in \Sigma^* \mid s = xyz$

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

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



ALL



$\neg$RPP(A) =

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

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

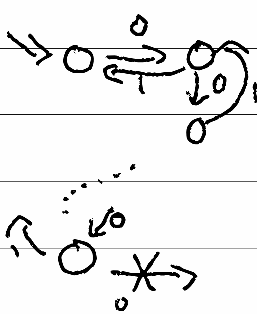
$\forall (x, y, z \in \Sigma^* \mid s = xyz \wedge |xy| < p \wedge |y| > 0)$

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

$xy^iz \notin A$

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

iff $\exists n \in \mathbb{N}.$ $x = \underbrace{00\ldots0}_{n}\underbrace{1\ldots1}_{n}$



~~int~~ long nat 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: $s = \cancel{\emptyset} = \varepsilon$

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

$s = 0^p 1^p$ $\qquad |s| = 2p \geq p$

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

$|xy| < p$ and

$|y| > 0)$

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

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

$z = 0^c 1^d$

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

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

$a + bi + c = d$

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

fun n:

if n == 0 : ret pz

else: let m = n-1

let pm = rec m

ps pm

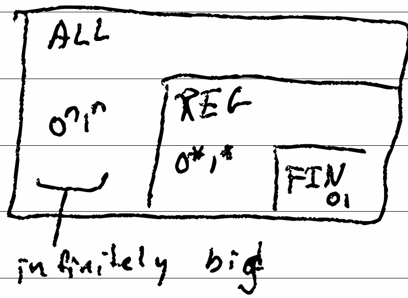
induction on N:

pz: $P\ 0$

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

$\Rightarrow \forall n : P_n$

9-3/ There is stuff outside REG!

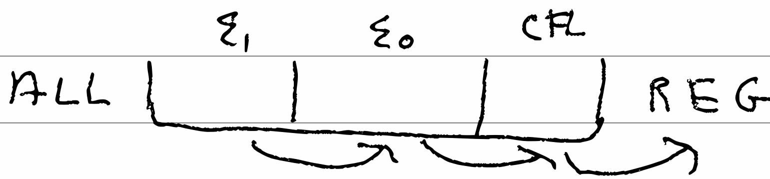


All computers are DFAs.

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) $0^x 1 0^y 1 0^{x+y}$

$01\ 0100 \Rightarrow$ "1 + 1 = 2" ✓

$001\ 0001\ 00000 \Rightarrow$ "2 + 3 = 5" ✓

$001\ 001\ 0 \Rightarrow$ "2 + 2 = 1" X

$\forall p.$

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

$\forall xyz. \quad x = 0^a \quad y = \underbrace{0^b}_{\text{change this}} \quad z = 0^c 1 0^p 1 \underbrace{0^{2p}}$

$\exists i.$ can't change

$\underbrace{2} + 2 = 4$

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

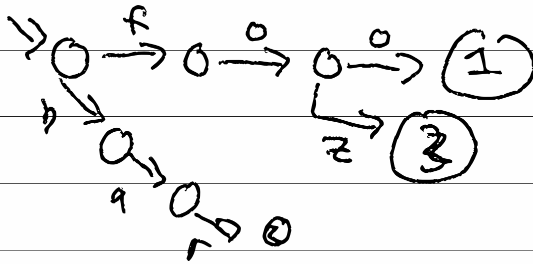
12-1 trie

foo ↦ 1

bar ↦ 2

foz ↦ 3

digital search tree



CFGs

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

start variable

A $S \to \varepsilon$

B $S \to 0S1$

← rule, productions, transitions

variables
non-terminals
symbols

rule = $\underbrace{V}\ \to\ \underbrace{\;}^{\text{rhs}}$

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

terminal

000111

$S \xrightarrow{B} 0S1 \xrightarrow{B} 00S11 \xrightarrow{B} 000S111 \xrightarrow{A} 000111$

"derivation"

$w \in G$ 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 B

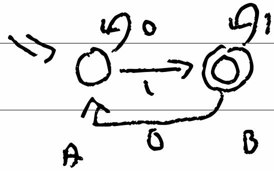
$\rightarrow$ $A \cup B$: $S \rightarrow A, S$

$S \rightarrow B, S$

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

$A \cap B$: X

DFA $\Rightarrow$ CFG



$$\begin{array}{llll} & X & Y & \\ A \rightarrow & 0A & \mid 1B & \\ B \rightarrow & 1B & \mid 0A & \mid \varepsilon \\ & Z & W & V \end{array}$$

01101

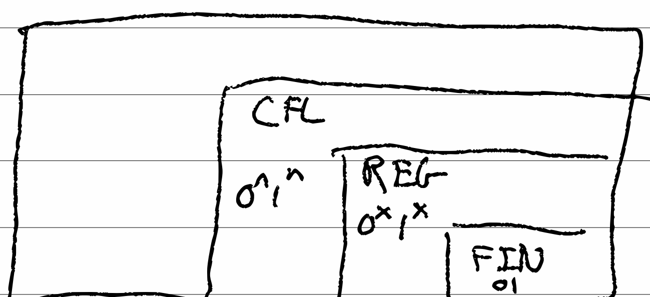
$A \xrightarrow{X} 0A \xrightarrow{Y} 01B \xrightarrow{Z} 011B$

$\xrightarrow{W} 0110A \xrightarrow{Y} 01101B \xrightarrow{V} 01101$

$S = q_0$

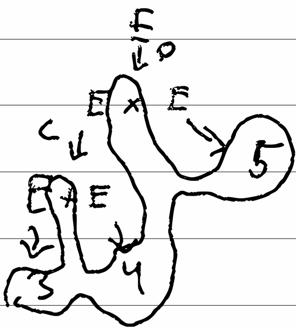
$q_i \rightarrow c\, q_j$ iff $\delta(q_i, c) = q_j$

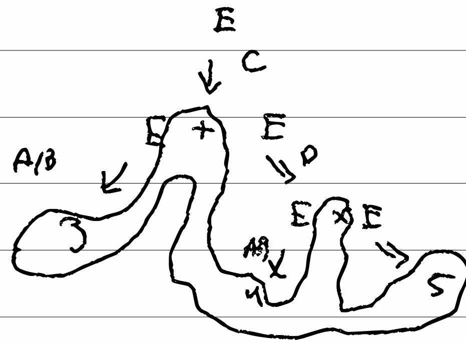
$q_f \rightarrow \varepsilon$ iff $q_f \in F$



$$\begin{array}{lccccccc} & & A & & B & & C & & D \\ \underline{12-4} / & E \to & 0 & | & 1 & | & E + E & | & E \times E \end{array}$$

$3 + 4 \times 5$





fowim $E \to n$
fowin $0 = 0$
$1 = 1$
$Plus(L, R) =$ fowim L $+$ fowim R
$Mult(L, R) =$ fowim L $\times$ fowin R

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

if cond

12-5 Nice! amb $\to$ unamb

not possible

V, Σ, R, S

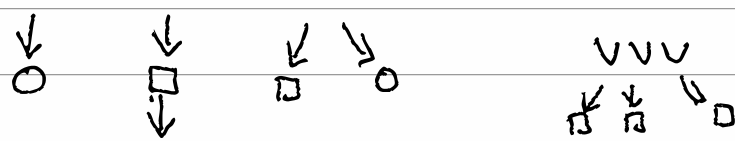
$V \to (V \cup \Sigma)^*$ $\quad\quad \varepsilon \quad\quad \Sigma\Sigma V$

$V \quad\quad VVVVV$

$V\Sigma$

$\Sigma \quad\quad V \quad\quad V\Sigma$

VVV

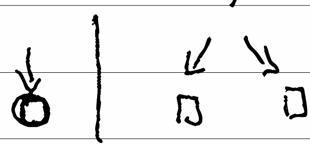


---

want: "pure" binary trees

Noam

Chomsky-Normal Form



$\Sigma \quad\quad VV$

$V \to \Sigma \quad | \quad A \to c$

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

$S \to \varepsilon$

CFG $\to$ 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 & $0^n 1^n$

$S \Rightarrow \varepsilon$ A

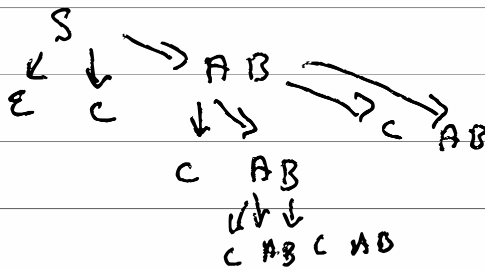
$S \Rightarrow 0S1$ B

$A \Rightarrow c$ +1

$A \Rightarrow BC$ +2

$B \Rightarrow$ +2

$A \Rightarrow$ done



$S \Rightarrow XYZ$

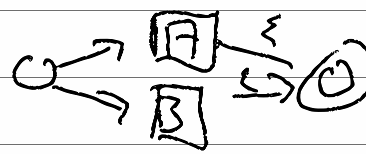
$X \Rightarrow ZZS \mid 0$

$Y \Rightarrow XYX \mid 1$

$Z \Rightarrow YSX$

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

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



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

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

unionstate = start | end | fromA qa
| fromB qb

δ (start, ε) = { fromA $A.q_0$, fromB $B.q_0$)

δ(fromA q_i, c) = fromA ($\delta_A(q_i, c)$)

12-7 new DFA("blah",

$q_i \to q_i == 0 \| q_i == 1$,

sigma, delta,

F)

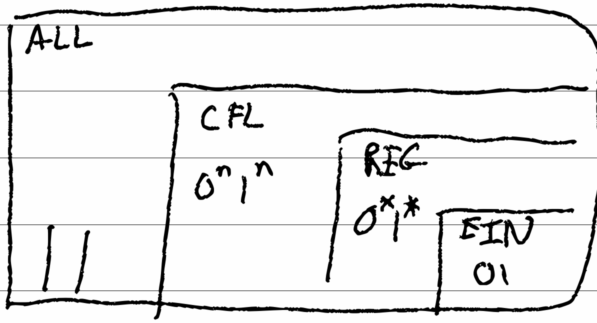
$\Downarrow$

complement (DFA d) {

new DFA(— , $q_i \to\ !\ d.F(q_i)$)

(d.Q, d.sig, d.delta,

15-1



REGULAR

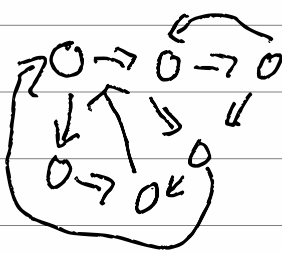
DFAs $\longleftrightarrow$ regex

$\in$ $\{\}$

CFL

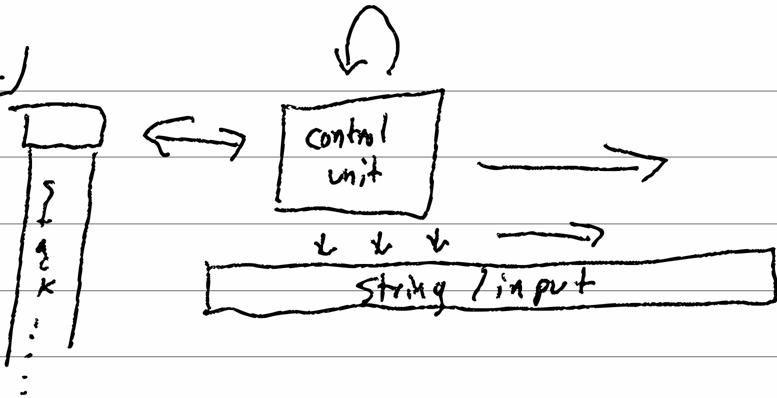
$\boxed{\text{PDA}}$ $\longleftrightarrow$ CFG

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)

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

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

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

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

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

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

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

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

let $\alpha : \beta = g$ in

let $c : w = s$ in

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

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

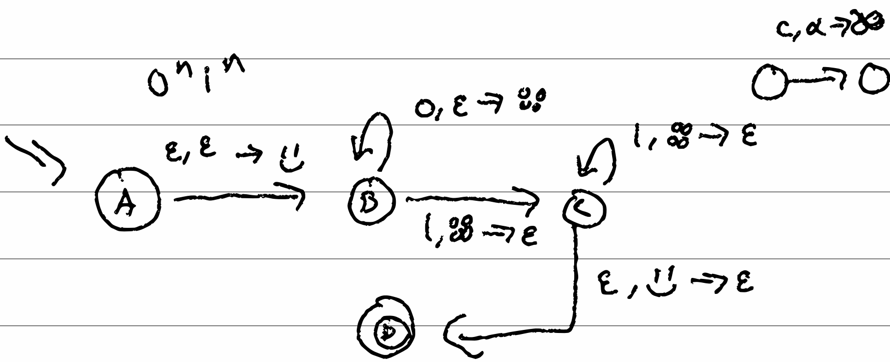
accepts : PDA × Σ^* → bool

accepts p s = while c.s ≠ ε do

c = sim p c

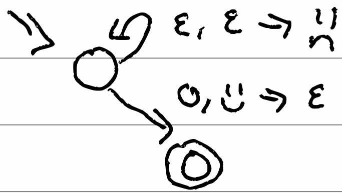
return c.q_j ∈ p.F

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/

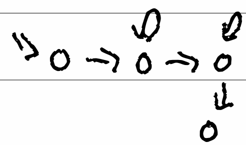


CFG → PDA

S → ε

S → 0S1

⇒



S ⇒ XYZ011SYX

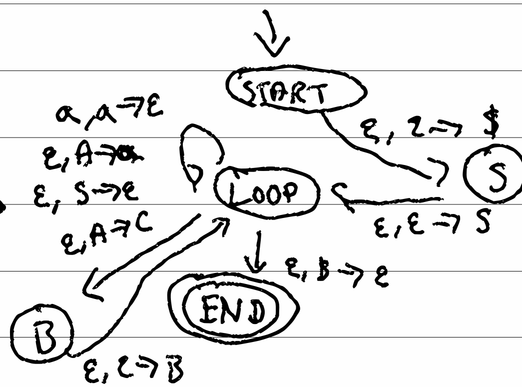
CNF → PDA

Γ = Σ ∪ V

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

⇙

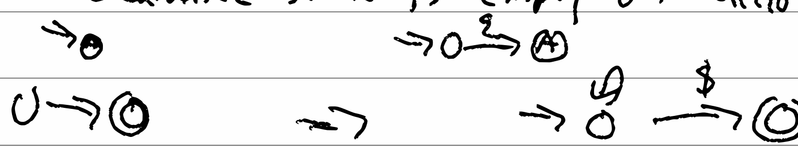
$11[L]11 → $1[L]1 → $[L] → [END] → ✓

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

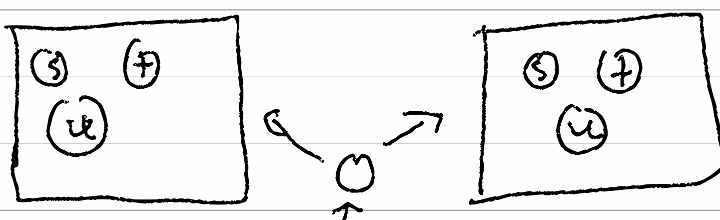
$V = (Q \times Q) \qquad S = (q_0, q_f)$

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

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

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

A <ast> B <bst>

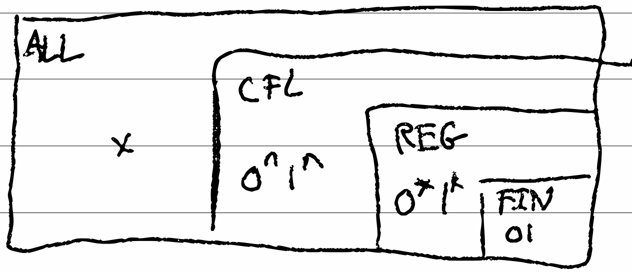


shawn <ast, bst>

$A \cup B$ < ~~[illegible]~~

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

16-1



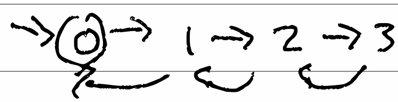
DFAs $\longleftrightarrow$ REX $\qquad 0^n 1^n \notin$ REG

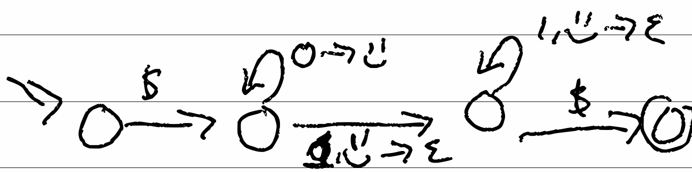
PDAs $\longleftrightarrow$ CFG $\qquad x \notin$ CFL

CFL = ALL

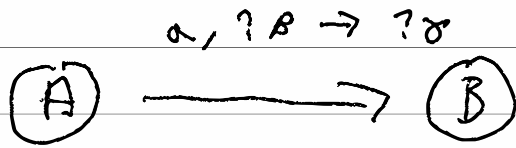
1.	P ...	$\forall r \in$ REG. $P(r)$	$\forall c \in$ CFL. $P(c)$
2.		$\exists x \in$ ALL. $\neg P(x)$	$\exists x \in$ ALL. $\neg P(x)$
3.	$\Longrightarrow$	$x \notin$ REG	$\Longrightarrow x \notin$ CFL

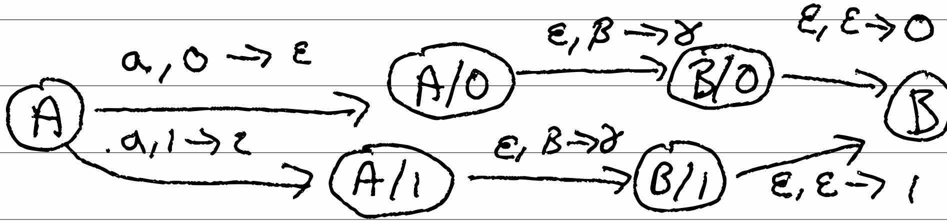
$0^n 1^n \rightarrow$





16-24 Look at 2nd thing





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

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

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

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

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

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

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

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

$\Rightarrow$ ⁘^{n} $[\,]\,0^n \Rightarrow$

16-3/ RPP

$\forall A \in REG,$

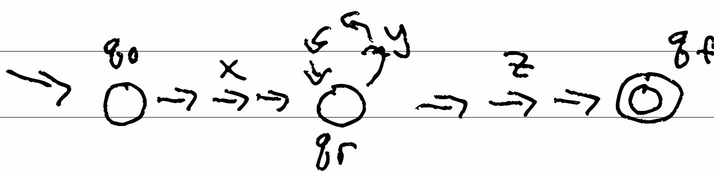
$\exists\, p \in N,$

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

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

$\forall\, i \in N,$

$xy^iz \in A.$



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

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

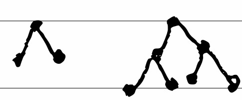
$v_i \Rightarrow v_x v_y$

$v_0 \quad v_1 \quad v_2 \quad \ldots\ldots \quad v_n \Rightarrow v_r$ $\quad v_r \in V$

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

$3^2 - 1$

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



2^2

2^3

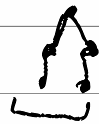
16-4/ minimum length of tree of height n

$S \Rightarrow \varepsilon$

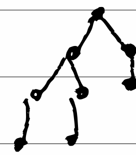
$= 0$

$S \Rightarrow a$ $= 1$

$S \to BC$



2

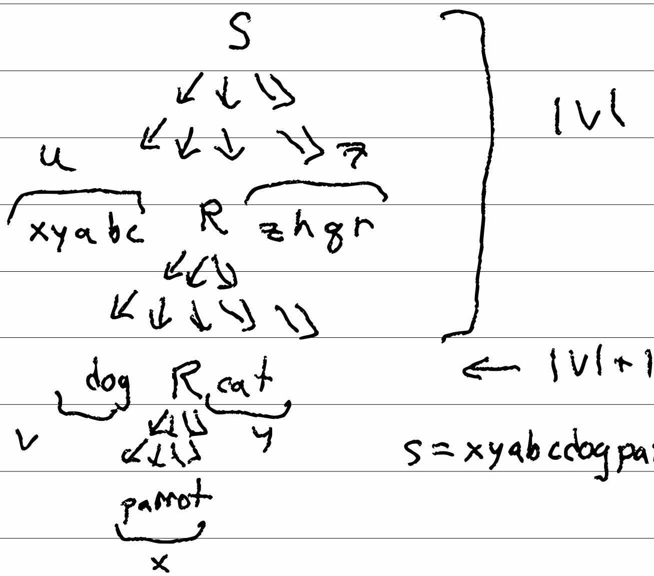


$h = 4 \Rightarrow \text{len} \geq 3$

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

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

$\forall A \in CFL,$

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

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

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

$|vxy| \leq p$

$|vy| > 0)$

$\forall i \in N,$

$uv^ixy^iz \in A$

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

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

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

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

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

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

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

given p

$0^p 1^p 0^p$

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

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

vxy has repetition

$vxy = 1$ symbol

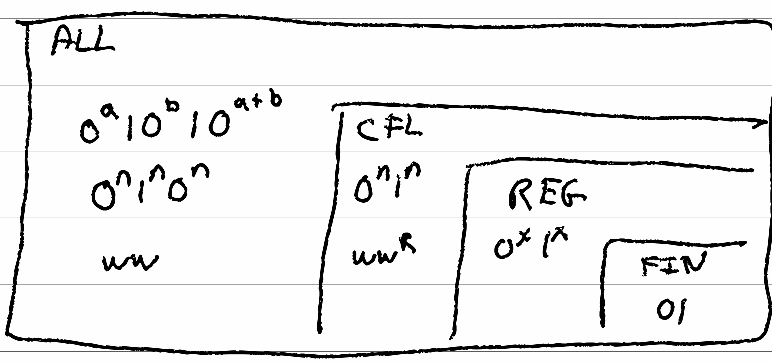
$0^p 1^p 0^p$ $vxy = 2$ symbol

3 symbols

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

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

DFAunion = Pair <X, Y> $\qquad$ NFAunion = Start
state FromLeft X ("x",X)
state From Right Y ("y",Y)
END



$\boxed{17\text{-}1}$ Finite : 0 1

Regular: $0^* 1^*$

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

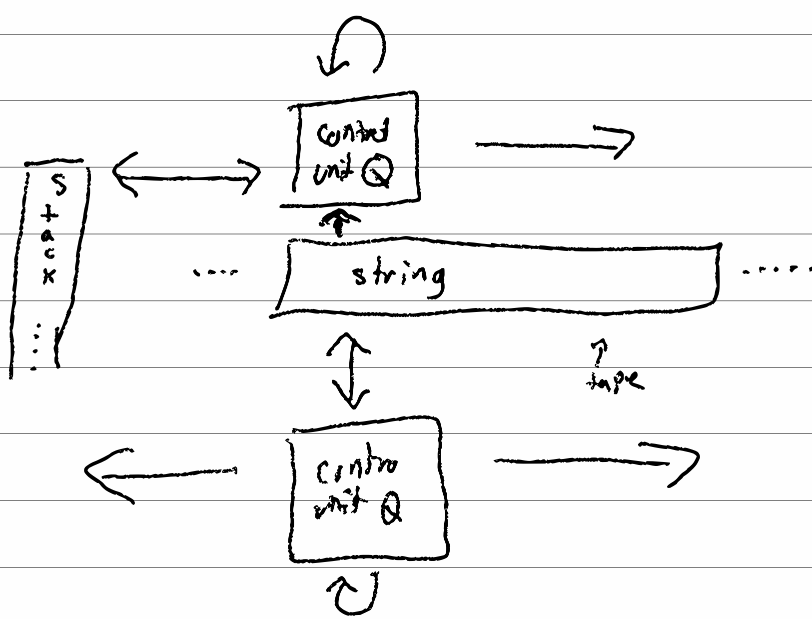
& CFPP $0^n 1^n 0^n$

context-free pumping property

$\not\Rightarrow$ CFG, PDA

Turing Machine Alan Turing

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



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 $(\Sigma \subseteq \Gamma)$ $\qquad \sqcup \in \Gamma$

$q_0 \in Q$

$q_a \in Q$ (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 $\to$ tape

left ε = $\varepsilon : \sqcup$

left $(x : c)$ = x

right : tape $\to$ tape

right $\varepsilon = \sqcup : \varepsilon$

right $(c : y) = y$

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

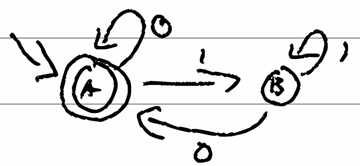
011 [q_7] 101

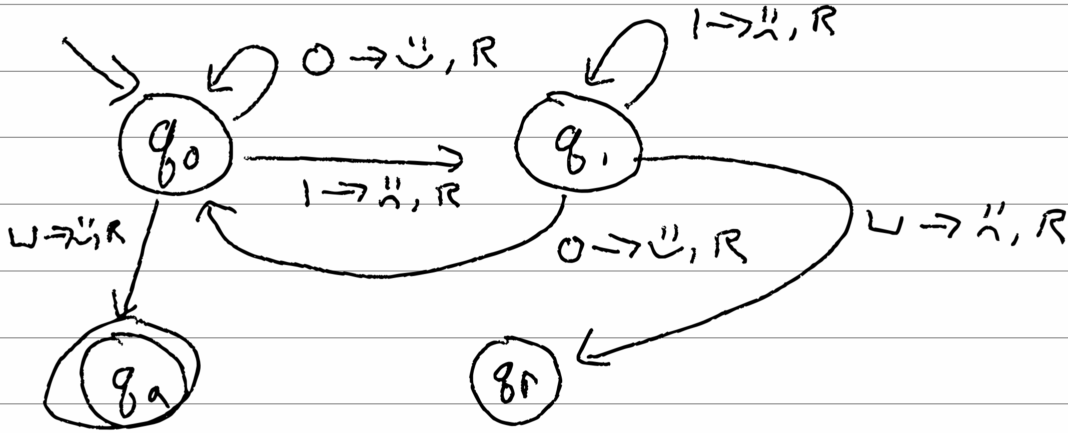
$\boxed{}\ \boxed{q_7}\ \boxed{}$

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

(DFA)

17-3





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

$\varepsilon[q_0]110 \to \ddot\frown[q_1]10 \to \ddot\frown\ddot\frown[q_1]0 \longrightarrow$
$\longrightarrow \ddot\frown\ddot\frown\ddot\smile[q_0]\varepsilon \to \sqcup\ddot\frown\ddot\frown\ddot\smile[q_0]\sqcup \to \sqcup\ddot\frown\ddot\frown\ddot\smile\ddot\smile[q_a]$

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

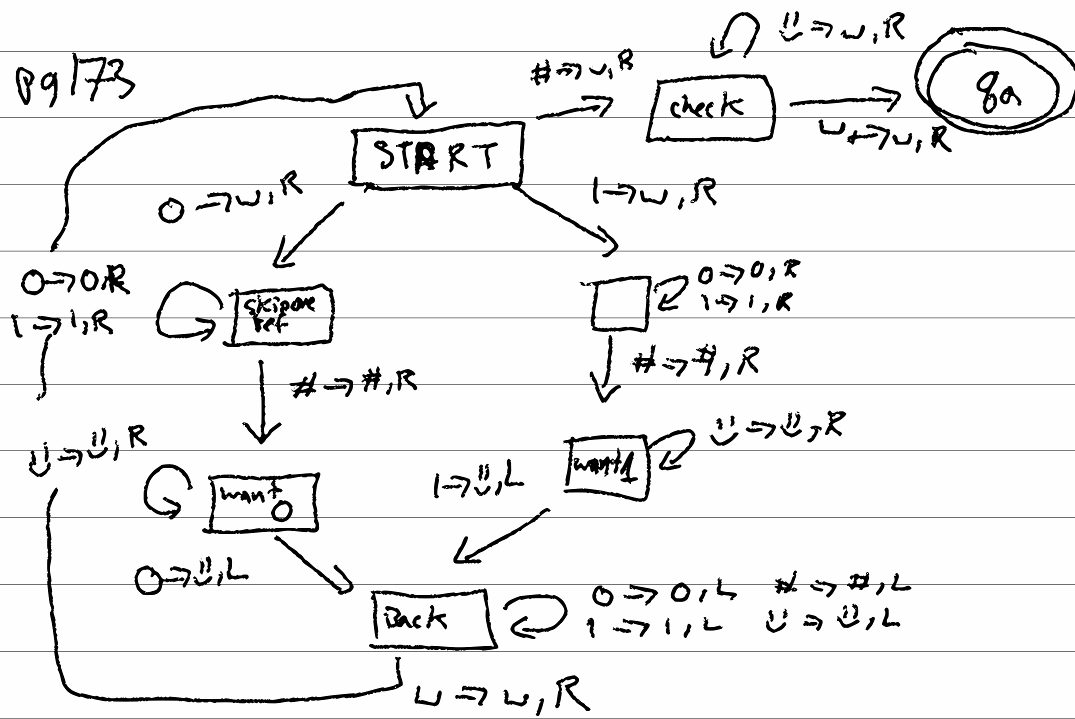
⊔# [I expect a zero] 01 → ⊔1 [I am happy] # ☺ 1

→ ⊔ [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

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

add1

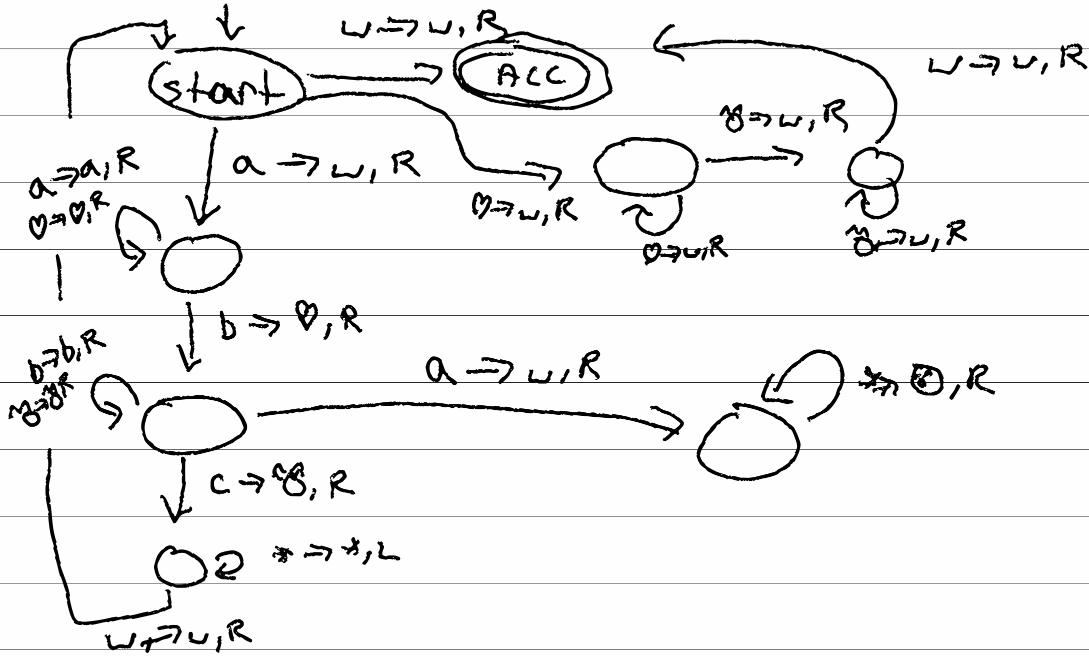
0	=	1
1	=	10
10	=	11
11	=	100

$(\supseteq \Sigma)$

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

$Q - \{q_a, q_r\}$

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



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

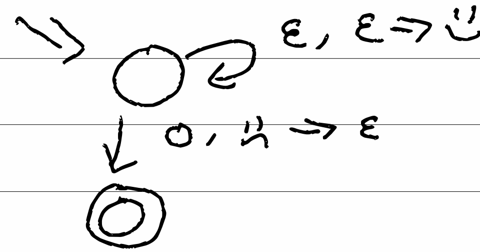
$f(0^x + 0^y) = 0^{x+y}$

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

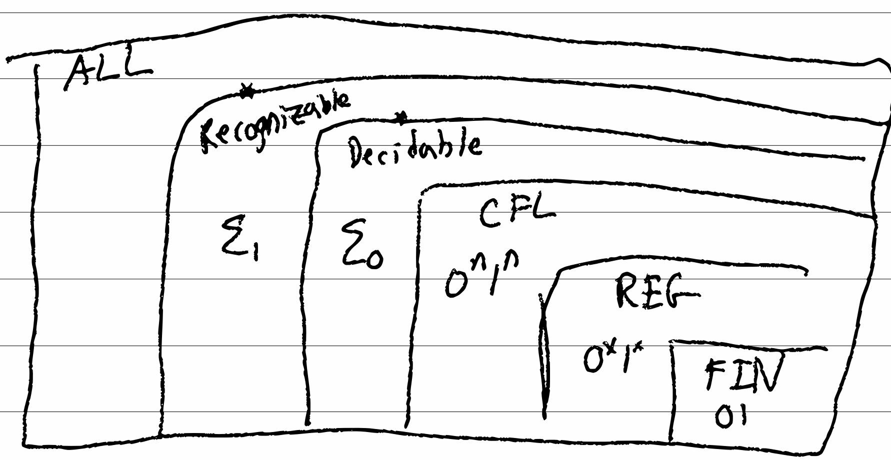
18-3/ A TM is either a

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

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

$w \in L(e)$ 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

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

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

$\Rightarrow$

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

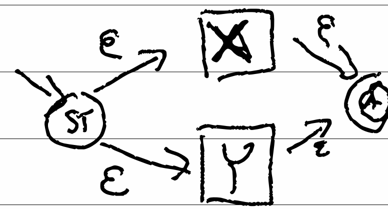
(q_{0x}, q_{0y}),

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

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

NFA union

REG $\cup$ $\to$ NFA



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

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

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

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

$c = \emptyset$

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

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

interface NFAUnionState <X, Y>

NUS_START () $\qquad$ = < nat $\to$ 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? ✓

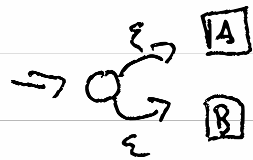
Σ_1

19-2 / Union

DFAs $Q = Q_A \times Q_B$

(did both at same time)

NFAs



TMs — can't do NFA (deterministic)

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

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

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

19-3/ Compiler correctness

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

C, assembly   asm, binary

NFA   DFA

bi-simulation

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

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

Galois connection

19-4) TM w/ "stay"

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

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

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

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

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

$u[q_i]av \Rightarrow$

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

$u[q_j]bv$

$\forall x \in \Gamma$

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

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

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

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

$$\begin{array}{l} uc \\ x \end{array}\left[q_i\right]\begin{array}{l} av \\ \alpha y \end{array} \Rightarrow \begin{array}{l} u \\ x\beta \end{array}\left[q_j\right]\begin{array}{l} cbv \\ y \end{array}$$

19-6/

MTM config $\xrightarrow{\text{compile}}$ 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 $\qquad\qquad$ 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, α)

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

(q_j)

$\Downarrow$

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

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

Σ_1 ✓