COMP 147

alphabet $\Sigma = \{ \dots \}$ finite set of symbols used in languages

string — finite sequence of symbols

infinite combinations

length not limited unless told

empty symbol ε — different from $\in$ (element of)

- length = 0, $|\{\varepsilon\}| = 1$ valid

language = set of strings

$L_1 = \{ \text{(binary representation of even numbers)} \}$

$= \{ w \mid w = 0^n 1^n, n \geq 1 \}$

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

└ infinite combinations of 0 and 1

building machines that recognise a language and will accept / reject strings based on their pattern

ex. washing machine (75 cents) (quarters)

$\{ , \{ \} \; 0 \dots 9 \} \in$ symbols to use

25 ×

25 25 ×

25 25 25 ✓

→ 111

25 25 25 25

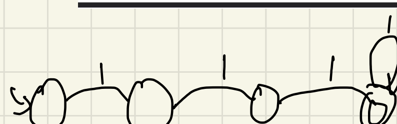
→ 1111 ....

$\Sigma = \{1\}$

$L_1 = \{111, 1111, 11111 \dots\}$

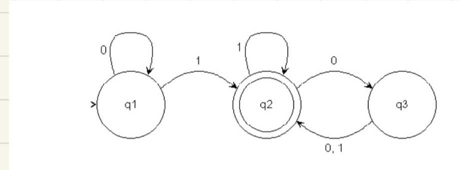
$= \{ w \mid w = 1^n, n \in \mathbb{N}, n \geq 3 \}$

$= \{ w \mid \lambda(w) \geq 3 \}$



initial state

loops if more coins are entered after accepting state

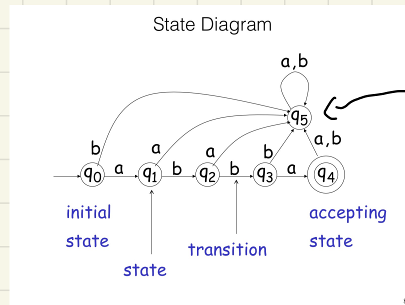


	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

$\lambda(q_1, 0) = q_1$

$\lambda(q_2, 0) = q_3$

, . . .



another state after accepting state, meaning there's a limit to the strings it'll accept

$L_1 = \{abba\}$

if ε, ends on start up since no input was taken

Ex. polynomial, integral root? $\Sigma = \{0, 1, 2 \dots 9, ",", -, "."\}$

$x^2 - 2x + 1$	$x^4 + 1$	$x^3 - 3x^2 + 3x - 1$	$x^2 - x - 0.75$
$x = 1$ ✓	none ✗	$x = 1$ ✓	$x = 1.5, x = -0.5$ ✗
1, -2, 1	1, 0, 0, 0, 1	1, -3, 3, -1	1, -1, -0.75

$L_1 = \{1,-2,1 \;,\; 1,-3,3,-1, \dots\}$

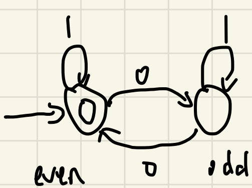
$= \{w \mid w$ represents a poly. with int root$\}$

binary string, even # of 0? $\Sigma = \{0, 1\}$

0001	01	100	ε	111	11
✓	✗	✓	✓	✓	✓

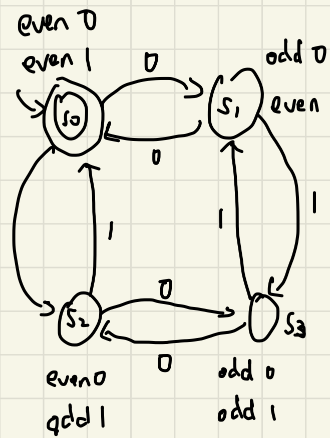
($\frac{0}{2}$ remand = 0)

$L_2 = \{w \mid w$ has even # of 0$\}$



$L_3 = \{w \mid w$ has even 0's and even 1's$\}$

0011 ✓, 1111 ✓, 011 ✗, ε ✓



(OR odd 0 odd 1

make S_3 final

then $F = \{S_0, S_3\}$)

if no final state

$L_4 = \{\}$ empty

(every state has trans.)

DFA req

set of states 1) $S = \{S_0, S_1, S_2, S_3\}$

set of symbol to use 2) $\Sigma = \{0, 1\}$

start 3) S_0

set of final states 4) $F = \{S_0\}$

edges 5)

	0	1
S_0	S_1	S_2
S_1	S_0	S_3
S_2	S_3	S_0
S_3	S_2	S_1

$F = \{\dots\}$

*4) set of states that are final

image $\delta(S_0, 0) = S_1$

$\delta(S_0, 1) = S_2$

⋮

regular language

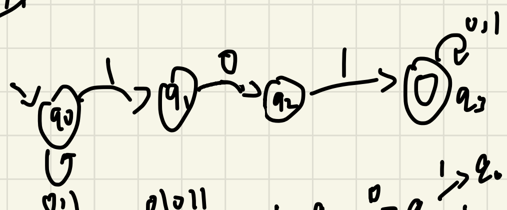
→ stuff that can make an DFA out of it

$L_1 = w \mid w = 0^n 1^n,\ n \geq 1 = \{01, 0011, 000111 \ldots$

no DFA can be designed for this because this is infinite, for it keeps continue the pattern in the language L_1

DFA - finite and deterministic (every state has transitions)

NFA - finite but multiple route (dead end included)



0,1   0101   0101 (first two entries)

01011

$q_0 \xrightarrow{0} q_0$

$\xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0$ (not accepted)

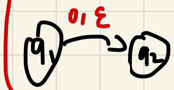
$\xrightarrow{1} q_0 \xrightarrow{1} q_1$

$\xrightarrow{1} q_1 \xrightarrow{1}$ X

$\xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_3$ ✓

$\varepsilon \to q_2 \xrightarrow{0}$ X

}



if a transition has ε, q_1 can move to q_2 without taking any input

10

$q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0$

$\searrow q_1 \xrightarrow{0} q_2$

$\varepsilon \downarrow$

$q_2 \xrightarrow{0}$ X

01011

$q_0 \xrightarrow{0} q_0$

$\xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0$ X

$\nearrow q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{1} q_3$ ✓

$\nearrow q_1 \xrightarrow{\varepsilon} q_2$ X

$\xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_3$ ✓

$\varepsilon \to q_2 \xrightarrow{0}$ X

1110

$q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0$ X

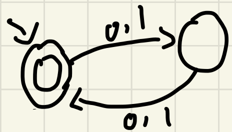
$\searrow q_1 \quad \searrow q_1 \quad \searrow q_1 \xrightarrow{0} q_2$ X

$\xrightarrow{1}$ X $\xrightarrow{1}$ X the edges in NFA will be a set of states that are connected by transitions

	0	1	(ε)
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$(\{\})$
q_1	$\{\}$	$\{q_2\}$	$(\{\})$
q_2	$\{\}$	$\{\}$	$(\{\})$

w|w length is even

DFA / NFA



BY DEFINITION

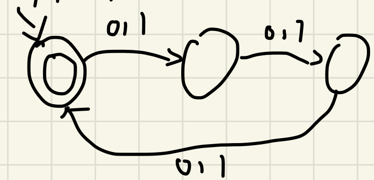
DFA = NFA

but NFA ≠ DFA

w|w length is multiple of 3

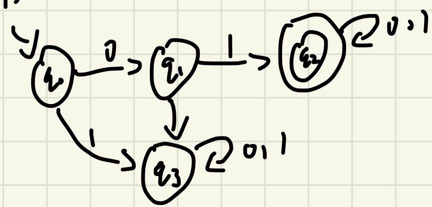
$= \{w \mid \lambda(w) = 3k,\ k \in \mathbb{N}\}$

DFA / NFA



w|w starts with 01

DFA



DFA $\delta: S \times \Sigma = S$

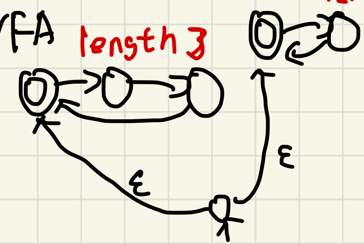
NFA $\delta: S \times \Sigma = P(S)$

w|w length is multiple of 2 OR ... of 3

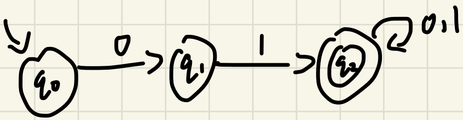
NFA

length 3

length 2



NFA



	0	1	ε
q_0	$\{q_1\}$	$\{\}$	$\{\}$
q_1	$\{\}$	$\{q_2\}$	$\{\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{\}$

$\delta(q_1, 1) = \{q_2\}$

$\delta(q_0, 1) = \{\}$

⋮

$L_1 = \lambda(w) = 3k$

$L_2 = \lambda(w) = 2k$

$L_3 = \lambda(w) = 3k$ OR $\lambda(w) = 2k$

$= L_1 \cup L_2$

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

Σ finite

0. 1.

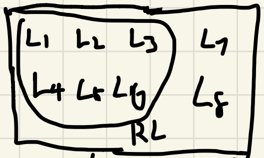
RL regular language

all strings

1110 0 1
ε 01

infinite

all language



DFA = NFA

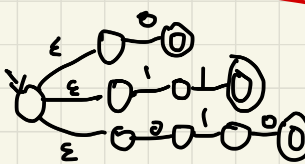
if DFA of L exists, then NFA of L exists
if NFA of L exists, then DFA of L exists

NFA ≠ DFA

L_1 = w | w has even # of 0's

L_2 = w | w starts with 01

$L_3 = \{0, 1, 010\}$ – NFA design:



L_4 = w | w is a binary string

$L_5 = \{\}$ $L_6 = \{\varepsilon\}$

↳ does not have accept state

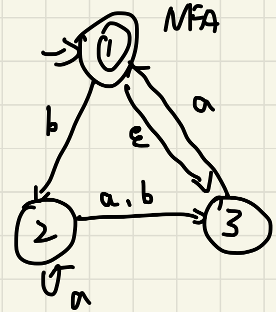
$L_7 = w \mid w = 0^n1^n, n \geq 0$

state depends on the numbers of 0's it has, so it's not finite

L_8 = w | w is a palindrome
$= \{0, 1, 00, 010, \varepsilon \ldots\}$

no DFA because of there exist infinitely # of palindromes

NFA



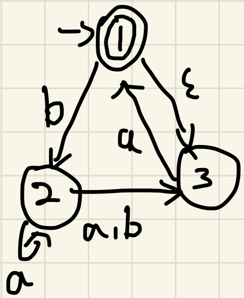
$S = \{1, 2, 3\}$

$P(S) = \{\{\phi\}, \{1\}, \{2\}, \{3\}$
$\{1, 2\}, \{1, 3\}$
$\{2, 3\}, \{1, 2, 3\}\}$

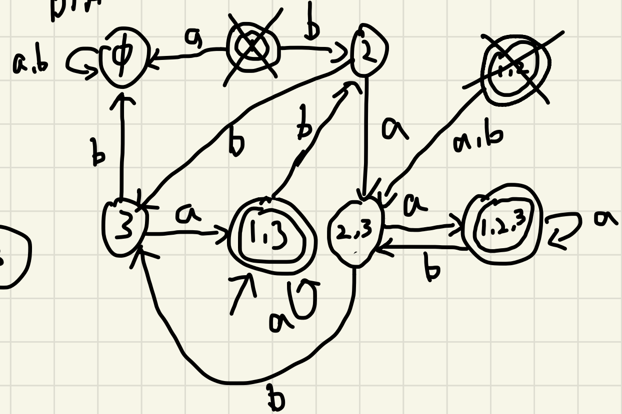
states of DFA = subsets of NFA

DFA next page

NFA



DFA



$S_{DFA} = P(S_{NFA})$

$\Sigma = \{a, b\}$

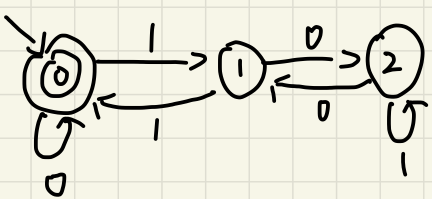
δ

start: $\{1, 3\}$

final $F = \{\{1,3\}, \{1,2,3\}, \{1,2\}, \{1\}\}$

(any sub state that has main final {1})

$\{w \mid w$ divisible by 3



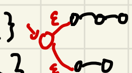
11 0 = 2(3k)
11 1 = 2(3k)+1
111 0 = 6k+2
111 1 = 2(3k+1)+1
= 6k+3
= rem. 0

101 0 = 6k+4 (3+1)
101 1 = 6k+5 (3+2)

regular language are closed under:

1) $\overline{L}$ compliment $L_2 = \overline{L_1} = \{w \mid w \notin L_1\}$

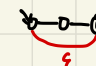
2) $\cup$ nion $L_1 \cup L_2 = \{w \mid w \in L_1$ OR $w \in L_2\}$



3) $\cap$ intersection $L_1 \cap L_2 = \{w \mid w \in L_1$ AND $w \in L_2\}$

4) concat $L_1 \cdot L_2 = \{w \mid w = u \cdot v,\ u \in L_1, v \in L_2\}$

5) * star $(L_1)^* = \{w \mid w = (u)^*,\ u \in L_1\}$



if ε exists, have extra start state that accepts ε

set of int are closed under +, −, × (but not /)

1) 4) if L_1 is regular, then $\overline{L_1}$ / L_1^* is regular

2) to 4) if L_1 and L_2 are regular, then the product is regular

3): $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$ De Morgan's NEW!!

$= \overline{\overline{L_1}} \cap \overline{\overline{L_2}} = L_1 \cap L_2$

NFA compliment next page

being regular language means it has DFA/NFA

4) ε transitions amount
= L_1 final states amount

NFA compliment

- don't flip states (work with DFA)
- make NFA → DFA then flip states
  LONG PROCESS

intersect

$L_1 \cap L_2$ final state should be accepted in both languages

ex. L_1 $F=\{a\}$ L_2 $F=\{e\}$

$L_1 \cup L_2$ $F=\{a,b\},\{a,c\}\ldots\{a,e\},\{b,e\}$ $L_1 \cap L_2$ $F=\{a,e\}$

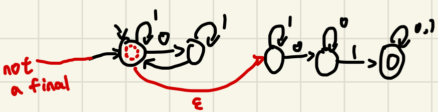
can use $L_1 \cup L_2$ for same diagram

$L_1 \cup L_2$ F = {any state that contains 1 final from $L_{..}$}

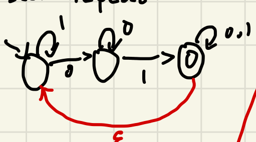
$L_1 \cap L_2$ F = {the sub state that has both final}

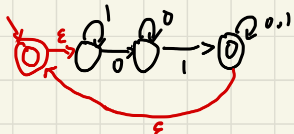
concat

$L_1 \cdot L_2$ L_1 final → L_2 start



L_2^* star repeat (loop back to start, from final states)





correct version because L_2^* may not have 1 repetition

Regular Expressions

ex. $(0 \cup 1)^*$ = Language

→ 0 or more repetitions of 0/1

→ ε, 0, 1, 00, 01 ...

precedence order: $\Sigma=\{0,1\}$

$*$, $\cdot$, $\cup$ like math

$(0 \cup 1)\cdot 0^* = \{1, 0, 10, 00, 100 \ldots\}$

= w | w is arbitrary # of 0's or 1 followed by arbitrary # of 0's

$1^*01^* = \{0, 10, 101 \ldots\}$

= w | w has exactly one 0

$R_1 = 0 \to L(R_1) = \{0\}$

$R_2 = 1 \to L(R_2) = \{1\}$

$R_3 = \varepsilon \to L(R_3) = \{\varepsilon\}$ by definition

$R_4 = \phi \to L(R_4) = \{\}$ null

R_5 = if R_1 is reg and R_2 is reg, then $R_1 \cup R_2$ is reg

$= R_1 \cup R_2$

$L(R_5) = L(R_1) \cup L(R_2) = \{0, 1\}$

concat

$L(R_6) = L(R_1) \cdot L(R_2)$

$= 0 \cdot 1 = 01$

$L(\underbrace{0 \cdot 1}_{01}(\underbrace{\varepsilon \cup 1}_{\varepsilon,1}))$

$= \{01, 011\}$

$L((0 \cup \phi)\cdot 11)$

$\{0\}$ $\{11\}$ $\{\}$ ✗ ✓ $\{\varepsilon\}$

$= \{011\}$ $\{11\}$

if R_1 is reg and R_2 is reg, then $R_1 \cdot R_2$ is reg

if R_1 is reg, then $(R_1)^*$ is reg

$L[R_7] = (L(R_1))^* = (\{0\})^*$

$= \{\varepsilon, 0, 00, 000 \ldots\}$

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

$= \{\varepsilon, 0, 00 \ldots\} \cup \{1\}$

$(0 \cup 1)^* = \{0,1\}^* = (L(0 \cup 1))^*$

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

$\{0,1\} \cdot \{0,1\} \cdot \{0,1\}$

$L[(1^*01^*01^*)^*] = \{\varepsilon, 00, 0010 \ldots\}$

even # of 0

$(1^*01^*01^*)\cdot(1^*01^*01^*)$

0 0 1 0 0

$L_1 = \{w \mid w \text{ contains one } 0\}$

$= 1^*01^*$

$L_2 = \{w \mid w \text{ has at least one } 1\}$ two 1

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

$L_3 = \{w \mid w \text{ has } 010 \text{ as substring}\}$

$= (0\cup 1)^* 010 (0\cup 1)^*$

$L_4 = \{w \mid \text{every } 0 \text{ in } w \text{ followed at least one } 1\}$

$= 1^*(01^+)^*$

$L_5 : \{w \mid w \text{ has even length}\}$

$= [(0\cup 1)(0\cup 1)]^* \cdot (0\cup 1)$ if odd

$(0\cup 1) = \Sigma \rightarrow \Sigma = \{0,1\}$

$R_1 = [0(0\cup 1)^*0] \cup [1\cdot(0\cup 1)^*1] \cup [0] \cup [1]$

$= \{w \mid$ begin/end with 0 OR begin/end with 1$\}$

$= \{w \mid w$ starts/end with same symbol$\}$

if L_1 is the language of reg expression L_1 has NFA

Convert RE to NFA

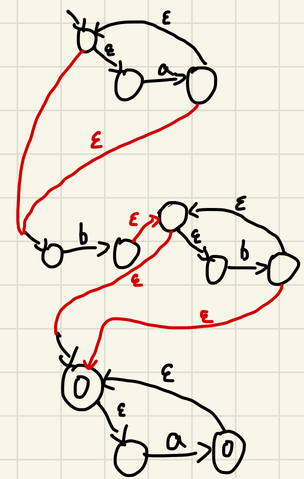
$a^*b^+a^*$

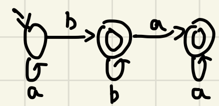
a^*

b^+

$b \cdot b^*$

a^*



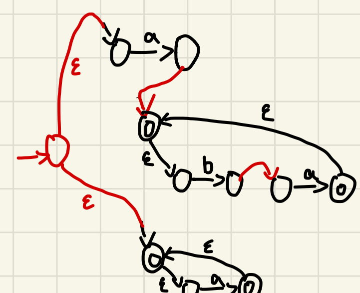


$a(ba)^* \cup a^*$

a

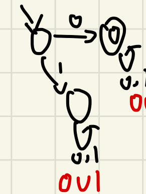
$(ba)^*$

a^*

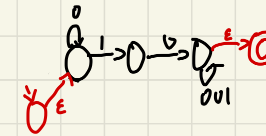


RE on transition (GNFA)

$w \mid w$ starts with 0



substring 10

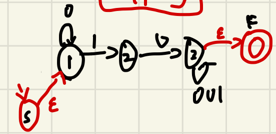


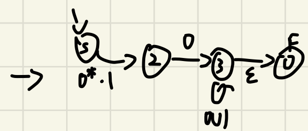
no input on start state

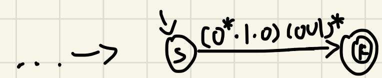
no output on final state

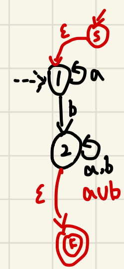
multiple transitions → RE

ripping

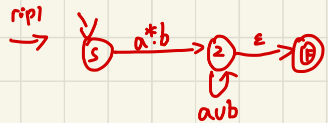




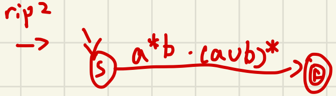


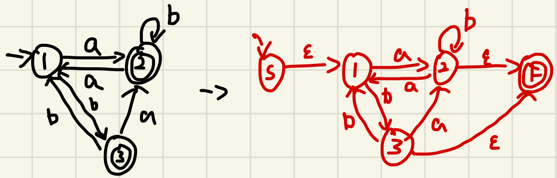


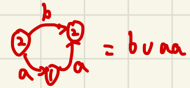
rip1



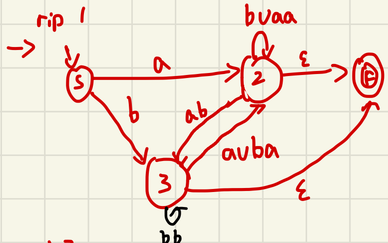
rip2



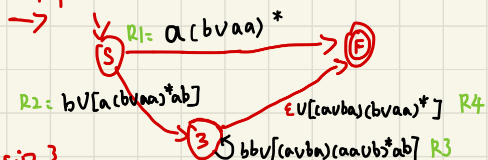




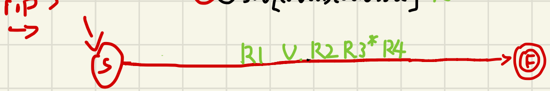
rip 1



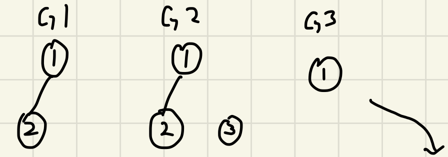
rip 2



rip 3

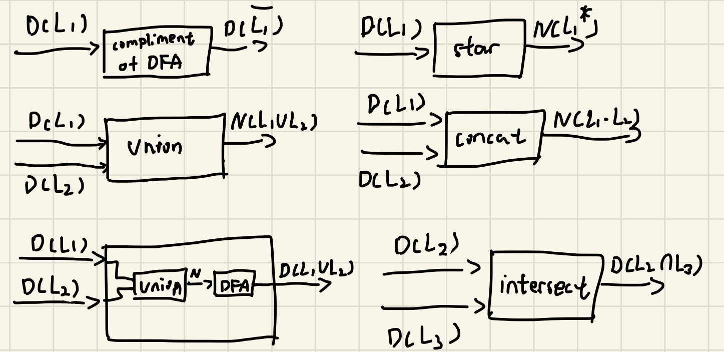


given any graph G, is G connected? review ?



L:{⟨1,2⟩, ⟨{1,2}⟩, ~~⟨1,2,3⟩, ⟨{1,2}⟩~~, ⟨1⟩, ⟨{1}⟩ ...}

✓ ✗ ✓



$\to \overline{L_2} \cup \overline{L_3} \to \overline{(\overline{L_2} \cup \overline{L_3})} \to L_2 \cap L_3$

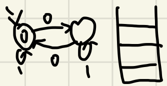
[later]

Push-down automata

Finite state machine that can handle a stack

DFA with a stack

L_2 = w| w even 0's



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

(including RL) -> context free language

w| w is even length

$S \to \varepsilon \mid SS \mid 00 \mid 01 \mid 10 \mid 11$

or $S \to \varepsilon \mid 00S \mid 01S \mid 10S \mid 11S$

or $S \to BBS \mid \varepsilon$

$B \to 0 \mid 1$

w| w has substring 0110

$S \to A0110A$

$A \to 0A \mid 1A \mid \varepsilon$

w| w has exact 2 0's

$S \to A0A0A$

$A \to 1A \mid \varepsilon$

Context free grammars

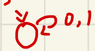
$L_5 : \{w \mid w = 0^n1^n, n \in N, n \geq 1\} = \{01, 0011, 000111 \dots\}$

1) S tart symbol $\to 0S1$, $n \geq 0$ $= \{\varepsilon, 01, 0011 \dots\}$

2) $S \to 01$ base

3) $S = \varepsilon$

$L(S) = \{\}$



$0S1 \to 00S11 \xrightarrow{(3)} 0011$

$S \to 0S1 \to 0011$

$S \to 0S1 \xrightarrow{(1)} 00S11 \to 000111$

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

$L_7 = \{w \mid w = 0^n\#1^n, n \geq N, n \geq 0\} = \{0\#1, 00\#11, 000\#111, \#\dots$

1) $S \to 0S1$

2) $S \to \#$ / $S \to A$

3) $A \to \#$: $L(A) = \{\#\}$, $L(S) = L_7$

$S \xrightarrow{1} 0S1 \xrightarrow{2} 0\#1$

$S \xrightarrow{1} 0S1 \xrightarrow{1} 00S11 \xrightarrow{2} 00\#11$

$\Sigma = \{(,)\}$

$L_8 = \{w \mid w$ is a balanced parenthesis expression$\}$

(())
aaabbb

()()()()
aababbab

1) $S \to aSb$

2) $S \to ab$

3) $S \to SS$

$\to S \to aSb \mid ab \mid SS$

$S \xrightarrow{3} SS \xrightarrow{2} Sab \xrightarrow{1} aSbab \xrightarrow{3} aSSbab$

$\xrightarrow{2} aSabbab$

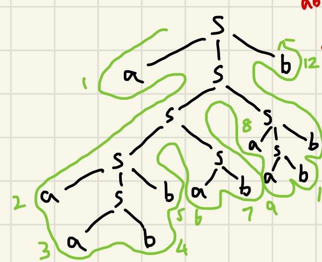
$\xrightarrow{2} aababbab$

aaabbabaabbb

$S \to aSb \to aSSb \to aSSSb \to aaSbSSb$

$\to aaabbSSb \to aaabbabSb$

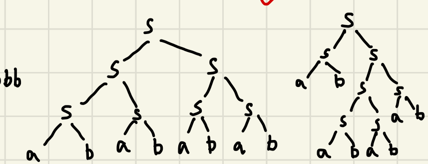
$\to aaabbabaSbb \to aaabbabaabbb$



if a string in grammar is ambiguous, then the grammar is ambiguous

2 ways to make 1 string

abababab

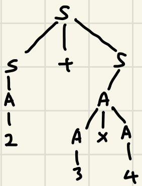


$S \to S\times S \mid S+S \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \ldots \mid 9$ ambiguous because S+S and S×S

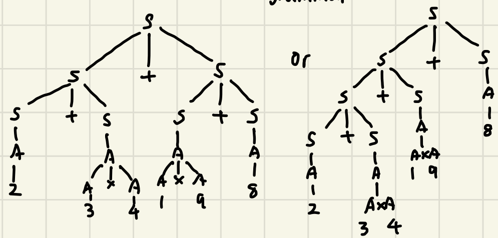
$S \to S+S \mid A$

$A \to A\times A \mid 1 \mid 2 \mid 3 \ldots \mid 9$

2+3×4



$2+\underline{3\times 4}+\underline{1\times 9}+8$ ← ambiguous grammar

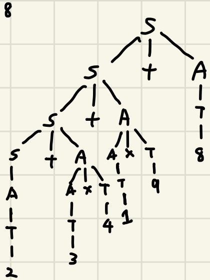


$S \to S+A \mid A$

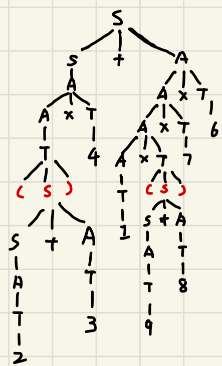
$A \to A\times T \mid T$

$T \to 1 \mid 2 \mid 3 \ldots \mid 9$

$2+\underline{3\times 4}+\underline{1\times 9}+8$

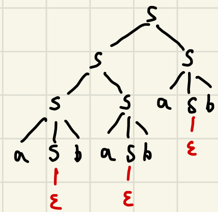


$\underbrace{(2+3)\times 4}_{S} + \underbrace{1\times(9+8)\times 7\times 6}_{A}$



$S \to aSb \mid \cancel{ab} \mid SS \mid \varepsilon$

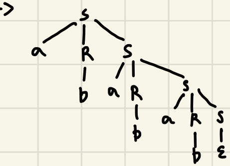
ababab



$S \to aRS \mid \varepsilon$

$R \to b \mid aRR$

ababab →



R = balanced () with extra)

ex. ())

$aR \to$ (()) balanced

$aRR \to$ (()))

$L_1 = \Sigma^* = (0\cup 1)^*$ — w is binary string

$S \to 0S \mid 1S \mid \varepsilon$

LL(1)   00011 0 →

$L_2 = \{\}$ $L_3 = \{\varepsilon\}$

$S \to S$ $S \to \varepsilon$

$L_4 = 0^*$

$S \to 0S \mid \varepsilon$

$L_5 = 00^*11^*$

$S \to 0S \mid 0A$

$A \to 1A \mid 1$

$L_6 = 0^*10^*1$

$S \to A1A1$

$A \to 0A \mid \varepsilon$

$L_7 = (01 \cup 00)^*1$ LL(2)

$S \to A1$

$A \to 01A \mid 00A \mid \varepsilon$

L_8 = w has equal # of a's and b's

balanced $S \to aM \mid bN \mid \varepsilon$

extra b $M \to bS \mid aMM \mid b$

extra a $N \to aS \mid bNN \mid a$

aM bN balanced

S: strings with equal a's and b's

M: strings with extra b

N: strings with extra a

L_9 = w | w has even # of 0's

$S \to \varepsilon \mid 0M \mid 1S$

odd 0 $M \to 0 \mid 0S \mid 1M$

L_{10} = w | w length 5 and third symbol is 0

$T \to MM0MM$

$M \to 0 \mid 1$

L = w | starts and ends with same symbol

$K \to 0 \mid 1 \mid 0M0 \mid 1M1$

$M \to 0M \mid 1M \mid \varepsilon$

L_U = w | start/end same symbol OR length 5 and third symbol = 0

$S \to F \mid T$
$F \to 0 \mid 1 \mid 0M0 \mid 1M1$
$M \to 0M \mid 1M \mid \varepsilon$
$T \to AA0AA$
$A \to 0 \mid 1$

L_P = w is a palindrome

$S \to 0S0 \mid 1S1 \mid \varepsilon \mid 0 \mid 1$

$L = \{w \mid w = 0^n 1^n,\ n \geq 1\} \neq 0^m 1^n,\ m \geq 1, n \geq 1$
$S \to 01 \mid 0S1$ $\quad 0^+ 1^+$
not regular

$L_E = \{w \mid w = a^i b^{i+k} c^k,\ i \geq 1, k \geq 1\} \Rightarrow S \to TF$
$\{abbc, aabbbc, \dots\}$

$L_T = \{w \mid w = a^i b^i,\ i \geq 1\}$ $\quad L_{bc} = w \mid w = b^k c^k,\ k \geq 1$
$T \to aTb \mid ab$ $\quad F \to bFc \mid bc$
ab, aabb, ... $\quad L_E = L_T \cdot L_{bc}$ $\quad$ bc, bbcc, ...

Chomsky Normal Form

ex. $S \to AB$, $A \to CD$, $B \to CA$
or $A \to a$, $C \to c$, $D \to d$

no transitions to ε
variable → 2 variables
variable → single terminal symbol

every grammar S with language L, there is a CNF grammar N with language L, except ε

× $N \to \varepsilon$ ×

$S \to aSb \mid ab$
if $aSb \mid \varepsilon$ same CNF because no ε and same strings

$N \to AB \mid BC$
$C \to NB$
$A \to a$
$B \to b$

$N \to AB \to aB \to ab$
$N \to AC \to ANB \to AABB \xrightarrow{\dots} aabb$ $\quad 2(2)-1=3$ $\quad 2(4)-1=7$
$N \to AC \to ANB \to AACB \to AANBB \to AAABBB \xrightarrow{*} aaabbb$
...

length of string n derived into 2n − 1
start to fully expand n − 1
fully expanded to actual string n
AABB $\quad$ aabb

$S \to AB$, $A \to aA \mid a$, $B \to AB$ → $S \to A\cancel{B}$, $A \to aA \mid a$, ~~$B \to AB$~~

3) eliminate unneccessary variables
- a variable that doesn't generate a terminal string
- a variable that can't be reached from start

1) eliminate ε productions

$S \to AB \mid B \mid A \mid \cancel{\varepsilon}$
$A \to aA \mid \cancel{\varepsilon} \mid a$
$B \to bB \mid A \mid \cancel{\varepsilon} \mid b$

delete ε trasitions, make cases so that A is not reached in same transitions

2) eliminate unit production

because C is unreachable
$S \to \cancel{ABC} \mid \cancel{BC} \mid \cancel{AC} \mid \cancel{C} \mid AB \mid B \mid A$
$A \to aA \mid a$
$B \to bB \mid b$

$S \to ABC$
$A \to aA \mid \varepsilon$
$B \to bB \mid \varepsilon$
~~$C \to \varepsilon$~~

→ $S \to ABC \mid AB$
$A \to aA \mid \varepsilon$
$B \to bB \mid \cancel{\varepsilon}$

$S \to ABC \mid AC \mid AB \mid A$
$A \to ab \mid \cancel{\varepsilon}$
$B \to bB \mid b$

2) eliminate unit production

$S \to AB \mid Aa$
$A \to cD \mid aa$
$B \to aCb \mid \cancel{\varepsilon} \mid \cancel{A}$ → $B \to cD \mid aa$ → $B \to a \mid b \mid c$
$\cancel{C \to D}$ → $C \to a \mid b \mid c$
$D \to a \mid b \mid c$

$A \to bb \mid aa$

$\cancel{B \to A}$ $B \to bb \mid aa$

$B \to C$

$L_1 = \{w \mid w = 0^n 1^n, n \in \mathbb{N}, n \geq 1\}$

$S \to 0S1 \mid 01$

1) skip
2) skip
3) skip

ε transitions

* Unit production: $S \to A$
variable to variable

useless transitions
- unreachable
- no terminal transitions.
ex. $S \to 0S1$

to CNF: $S \to AB \mid S \to a$

$\Rightarrow S \to AB \mid CB$ (ASB)
$A \to 0$
$B \to 1$
$C \to AS$

grammar → convert to CNF → N

$S \to ASA \mid aB \mid CD$
$A \to B \mid S$
$B \to b \mid \cancel{\varepsilon}$
$C \to ab$
$D \to aDb$

1) ε transitions
$S \to ASA \mid aB \mid a \mid CD$
$A \to B \mid S \mid \varepsilon$
$B \to b$
$C \to ab$
$D \to aDb$

$S \to ASA \mid aB \mid CD \mid a \mid \cancel{S} \mid AS \mid SA$
$A \to B \mid S$
$B \to b$
$C \to ab$
$D \to aDb$

2) unit production
$S \to ASA \mid aB \mid CD \mid a \mid AS \mid SA$
$A \to B \mid S$
$\Rightarrow A \to b$
$A \to ASA \mid aB \mid CD \mid a \mid AS \mid SA$
$B \to b$
$C \to ab$
$D \to aDb$

3) useless transitions

$S \to ASA \mid aB \mid CD \mid a \mid AS \mid SA$
$A \to B \mid S$
$\Rightarrow A \to b$
$A \to ASA \mid aB \mid CD \mid a \mid AS \mid SA$
$B \to b$
$C \to ab$
$D \to aDb$

$D \to aDb$ no terminal string
$\Rightarrow S \to CD$
$A \to CD$ removed
$\Rightarrow C$ unreachable
remove

$\Rightarrow$

$S \to EA \mid FB \mid a \mid AS \mid SA$
$A \to b \mid EA \mid FB \mid a \mid AS \mid SA$
$B \to b$
$E \to AS$
$F \to a$

$S \to a \mid aA \mid B$
$A \to aBB \mid \cancel{\varepsilon}$
$B \to Aa \mid b$

1) ε transitions
$S \to a$
$S \to aA$
× $S \to a$ to compensate $aA \to a\varepsilon \to a$
$S \to B$
$A \to aBB$
$B \to Aa$
$B \to a$
$B \to b$

2) unit production
$S \to a$
$S \to aA$
$\cancel{S \to B}$
$\Rightarrow S \to Aa \mid a \mid b$ ×
$A \to aBB$
$B \to Aa$
$B \to a$
$B \to b$

3) useless transitions

skip

CNF
$S \to a$
$S \to CA$
$S \to AC$
$S \to b$
$A \to CBB$
$B \to AC$
$B \to a$
$B \to b$

$C \to a$ for aA

$\to S \to a \mid CA \mid AC \mid b$
$A \to DB$
$B \to AC \mid a \mid b$
$C \to a$
$D \to CB$

$S \to ASB \mid b$
$A \to aAS \mid a \mid \varepsilon$
$B \to SbS \mid A \mid bb$

1) $A \to \varepsilon$ ×
$S \to b \mid ASB \mid SB$
$A \to aAS \mid aS \mid a$
$B \to SbS \mid A \mid \varepsilon \mid bb$

$B \to \varepsilon$ ×
$S \to b \mid ASB \mid SB \mid AS \mid \cancel{S}$
$A \to aAS \mid aS \mid a$
$B \to SbS \mid A \mid bb$

2) $B \to A$ ×
$B \to SbS \mid aAS \mid aS \mid a \mid bb$

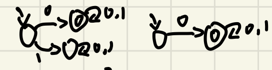
3) skip

CNF
$S \to b \mid CB \mid SB \mid AS$
$A \to DC \mid DS \mid a$
$B \to ES \mid DC \mid DS \mid a \mid FF$

$C \to AS$
$D \to a$
$E \to Sb \mid E \to SF$
$F \to b$

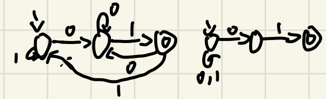
Push Down Automata (actual)

$L_1 = \{w \mid w \text{ starts with } 0\}$



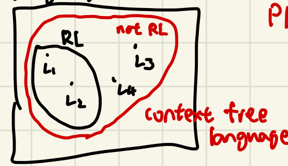
$0(0 \cup 1)^*$

$L_2 = \{w \mid w \text{ ends with } 01\}$



$(0 \cup 1)^* 01$

Language



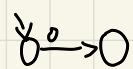
PDA : finite state automata (NFA) with a stack LOL

$L_3 = \{w \mid w = 0^n 1^n, n \geq 0\}$
- can grow infinitely large

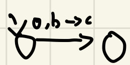
$L_4 = \{w \mid w = uu^r, u \text{ is any binary string}\}$ (r → reversed)

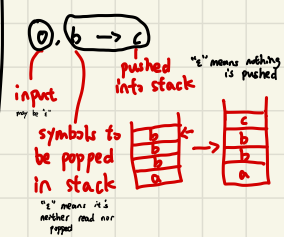
CFG for CF languages ≈ RE for Regular language

NFA



PDA



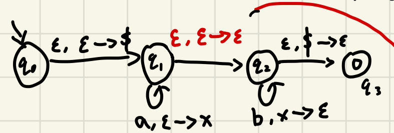


$0, x \to y$

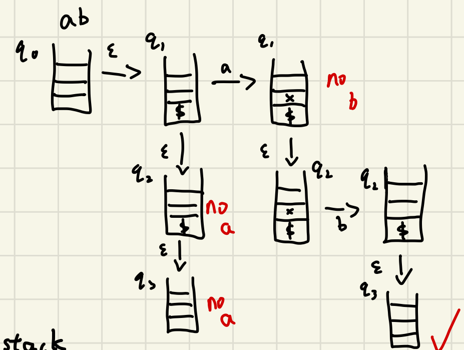
if x is not found on top of stack, the transition will be dead

$L_1 = w \mid w = a^n b^n, n \geq 0$

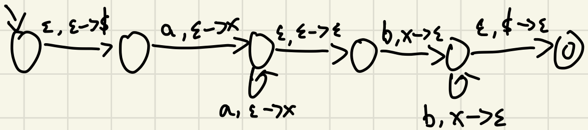
$\Sigma = \{a, b\}$
$\Gamma = \{\$, x\}$



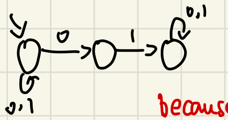
because $n \geq 0$



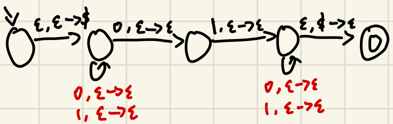
$w \mid w = a^n b^n,\ n \geq 1$



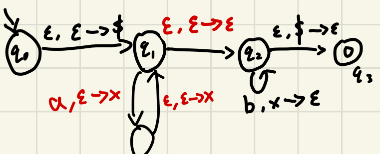
$L_2 = \{w \mid w$ has substring $01\}$



because this can reach substring 01 without using stack



$L_3 = w \mid w = a^n b^{2n},\ n \geq 0$



push 2x in stack for b^{2n} to pop later

CNF lab

b) $S \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid B \mid BB$

$A \to C$

$B \to S \mid A$

$C \to S$

$S \to B$ X

because A, B and C all points to S, so S could be simplified to

$S \to 0S0 \mid 00 \mid 1S1 \mid 11 \mid SS$

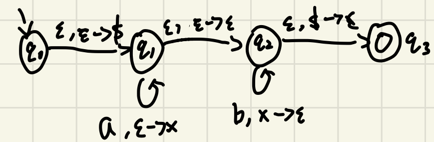
$S \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$

$A \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$

$B \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$

C removed because unreachable

$a^n b^n,\ n \geq 0$



1) $S = \{q_0, q_1, q_2, q_3\}$

2) $\Sigma = \{a, b\}$

3) $L_1 = \{x, \$\}$ alphabets used

4) start state q_0

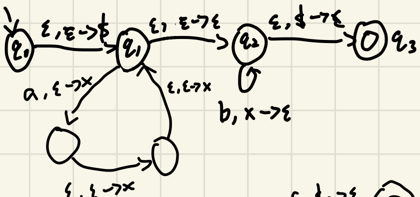
5) set of final state(s) $\{q_3\}$

6)

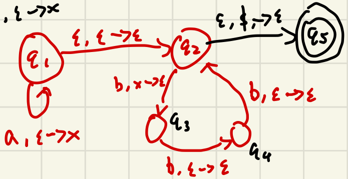
	a: $	a: x	a: ε	b: $	b: x	b: ε	ε: $	ε: x	ε: ε
q_0									$\{(q_1, \$)\}$
q_1			$\{(q_1, x)\}$						$\{(q_2, ε)\}$
q_2					$\{(q_2, x)\}$		$\{(q_3, ε)\}$		
q_3									

$a^n b^{3n}, \ n \geq 0$

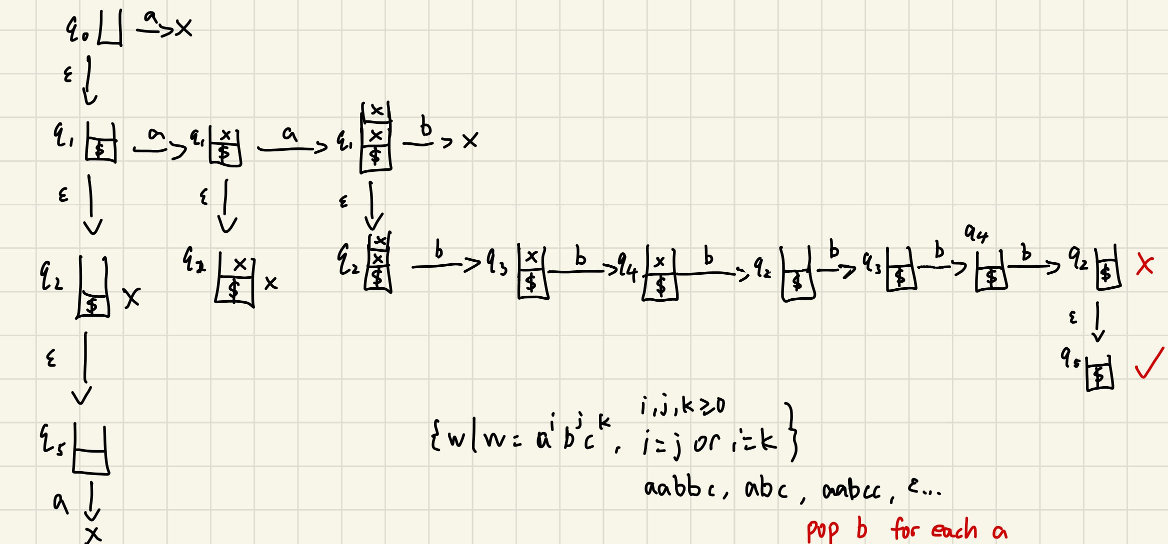
aabbbbbb



or



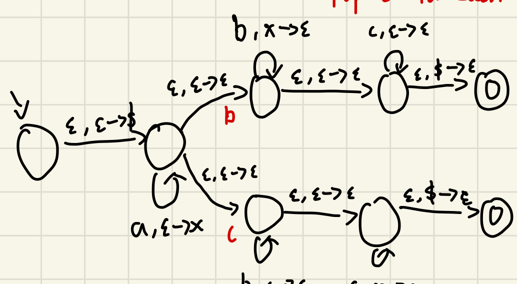
for each a,
read 3 b's



$\{w \mid w = a^i b^j c^k, \ i = j \text{ or } i = k\}$, $i, j, k \geq 0$

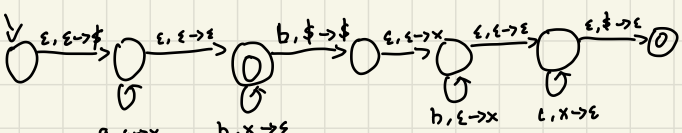
aabbc, abc, aabcc, ε...

pop b for each a



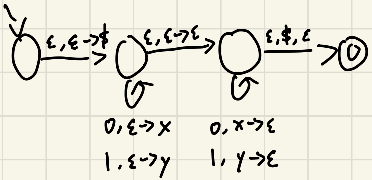
pop c for each a

$L_5 = \{w \mid w = a^i b^j c^k, \ i + k = j\}$, $i, j, k \geq 0$

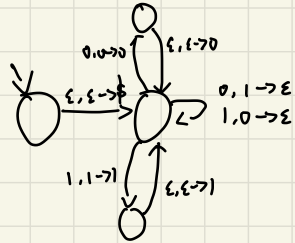


$L = \{w \mid w = vv^r,\ v = (0 \cup 1)^*\}$

00, 0110 ...



$L = \{w \mid w$ has equal # of 0 and 1 $\}$



$L = \{w \mid w$ has exactly one more a than b $\}$

$S \to aSb \mid bSa \mid a$

$S \to aA \mid bS \mid a$
$A \to aB \mid bAA \mid \varepsilon$
$B \to aBB \mid bA \mid b$

S → generates 1 more a than b
A → generate equal a and b
B → generate 1 more b than a

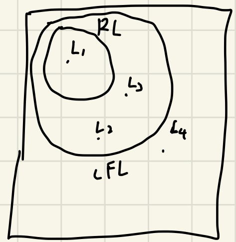
Turing Machine

given any string w, is w a 0 followed by arbitrary # of 1s followed by a 0

$\Rightarrow L_1 = \{010, 0110, 01110 \dots\}$

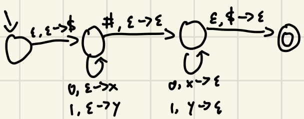
$= 01^*0$

have DFA and PDA



$L_2 = \{w = u\#u^r,\ u = (0 \cup 1)^*\}$

$= \{0\#0,\ 01\#10 \dots\}$



have PDA, doesn't have DFA

$L_4 = \{w = u\#u,\ u = (0 \cup 1)^*\}$

$= \{001\#001,\ \#,\ 11\#11 \dots\}$

no PDA and DFA

$L_3 = \{w = 0^n1^n,\ n \geq 0\}$

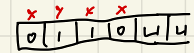
have PDA, no DFA

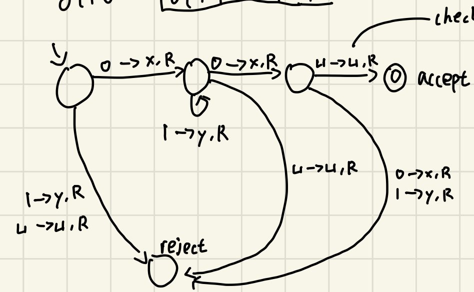
$L_1 = 01^*0$

0 -> x, R

if 0, rewrite to x and move head to the right

0110

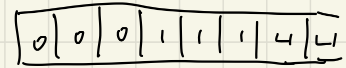


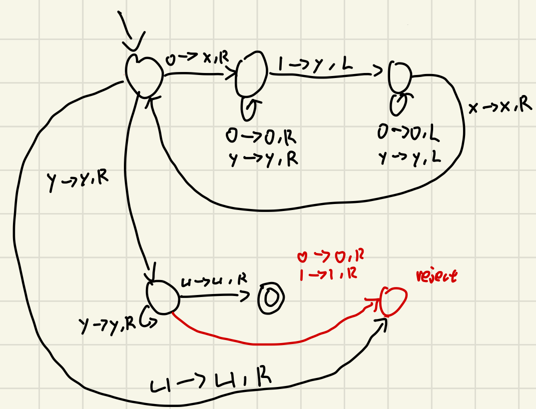


$\Sigma = \{0, 1\}$

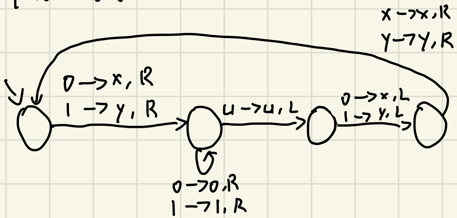
$\Gamma = \{0, 1, ⊔, x, y\}$

$L_3 = 0^n1^n,\ n \geq 0$

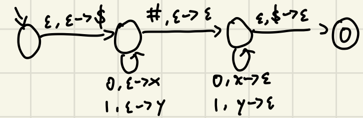


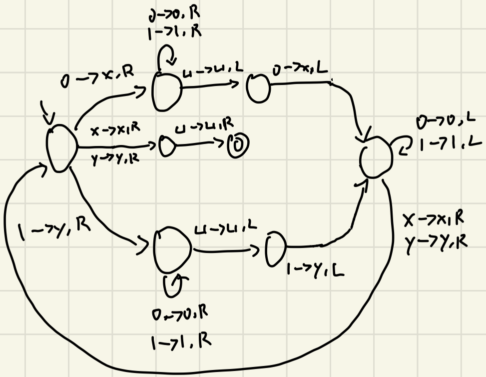


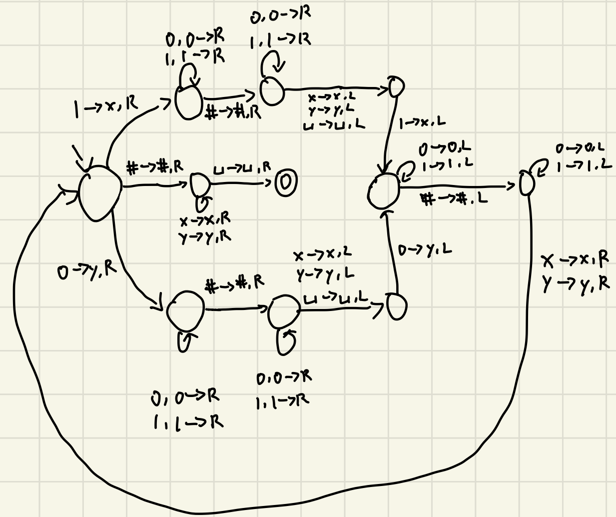
$L_3 = w \mid w = uu^r,\ u = (0 \cup 1)^*$



$L_4 = w \mid w = u\#u^r,\ u = (0 \cup 1)^*$

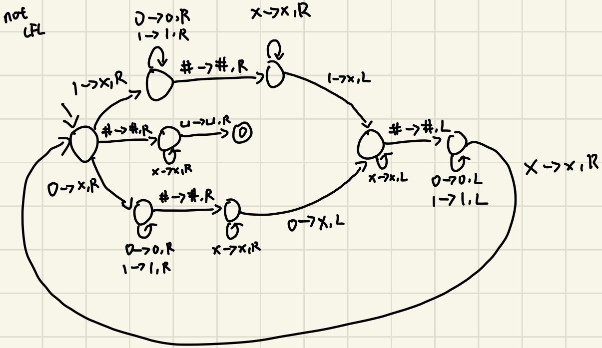




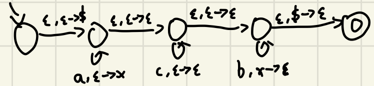


$L = w \mid w = u\#u,\ u = (0 \cup 1)^*$

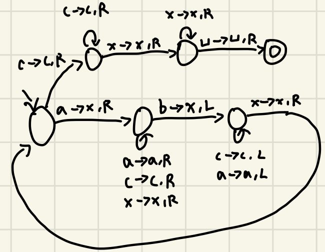
not CFL

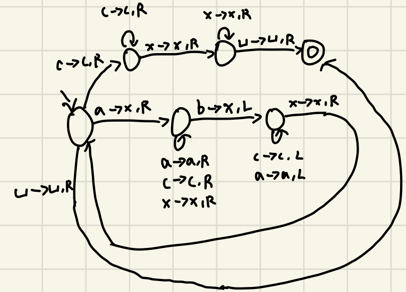


CFL $L = w \mid w: a^n c^m b^n,\ m,n \geq 1$



$m \geq 0,\ n > 0$





$B = ⊔$ blank

States = {q (start), f (accept)}

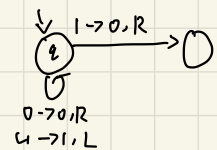
input = {0, 1}

Tape = {0, 1, B}

$\delta(q, 0) = (q, 0, R)$

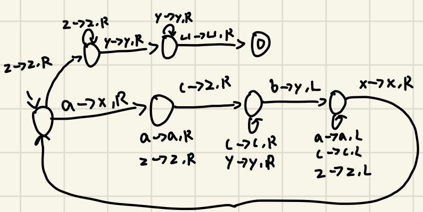
$\delta(q, 1) = (f, 0, R)$

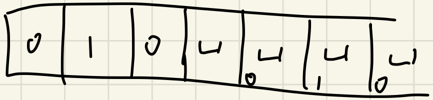
$\delta(q, B) = (q, 1, L)$

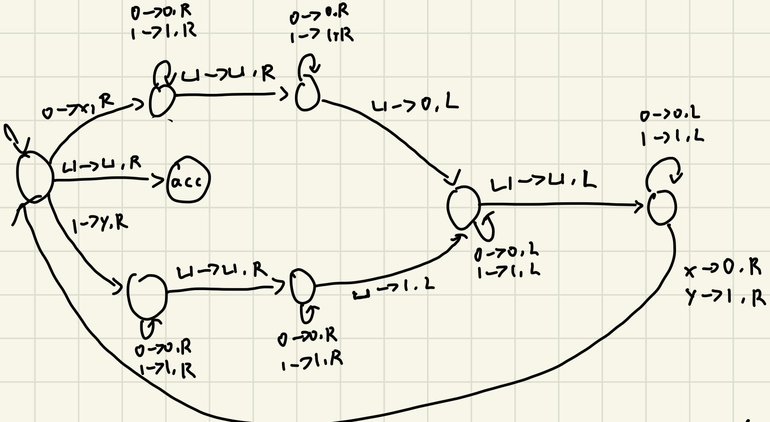


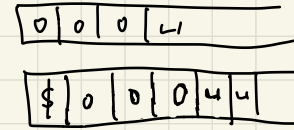
q0	0	1	⊔
0	q0	1	⊔
0	0	q1	⊔
0	0	0	⊔ | acc

$L = w \mid w = a^n c^n b^n,\ n \geq 1$

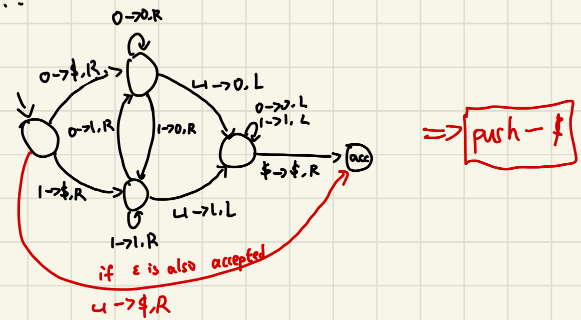




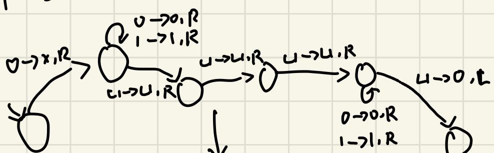




. . .

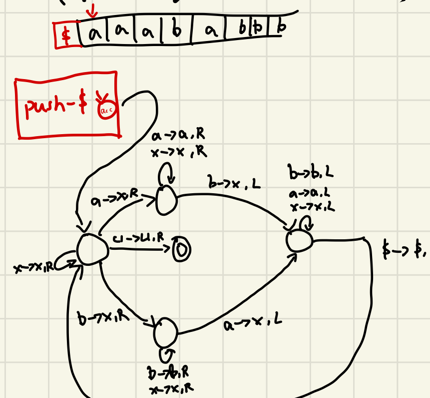


if 3 blanks between og and copied

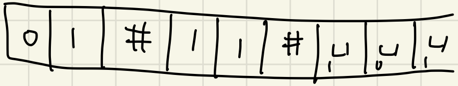


(same thing for 1 → y, R)

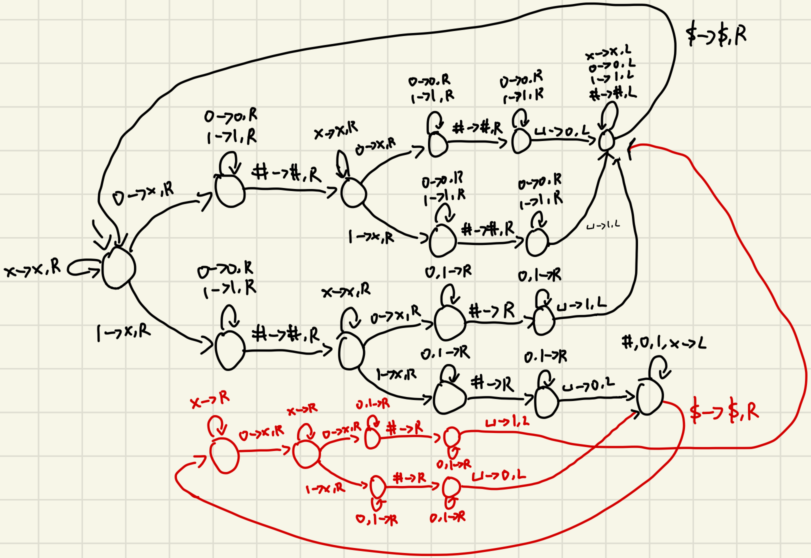
L = {w | w has equal # of a and b}



extra. TM for binary computations



not even finished lmao



because of "carried" value

TM recognizable / decidable

recog - if $w \in L$, eventually enters ⓞ
$w \notin L$, enters rej, or loops

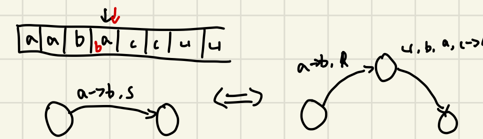
decidable - if $w \in L$, ... enters ⓞ
$w \notin L$, ... enters rej

1) TM that head of tape can stay

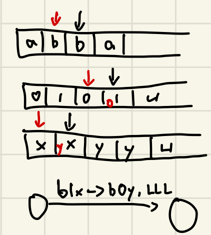
2) multitape TM

3) Non-deterministic TM

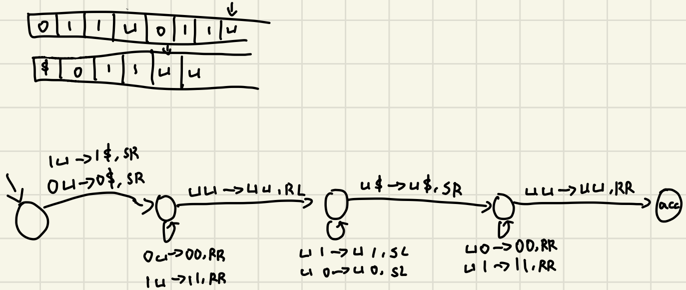
1)



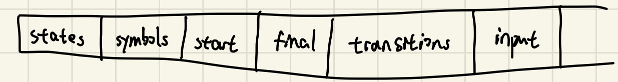
2)



all 3 needs to be the same in transition



formal definition of a DFA
can be used as a tape for TM



Given any Graph G, is G connected ?

M: on input w
if w is not a graph, reject
otherwise
mark a vertex of w
repeat until no new vertex gets marked
- mark a vertex that is not marked yet
if there is a direct edge between that vertex
and some vertex that is already marked

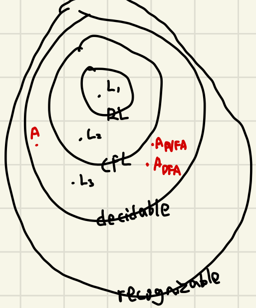
if all vertices are marked, accept
otherwise, reject

Given any DFA like D and string like w,
is w in the language of D

$\{ \langle D, w \rangle \mid w \in L(D) \}$

<encoding of DFA, input string>

M: on input w
if w is not encoding of DFA
followed by some string, reject
otherwise
w is in the form $\langle D, w' \rangle$
simulate D on w'
if D accepts w'
M accepts $\langle D, w' \rangle$
if D rejects w'
M rejects $\langle D, w' \rangle$



$L_1 = w \mid w$ has substring 001

$L_2 = w \mid w = uu^r$, u is a binary string

$L_3 = w \mid w = u\#u$, u .. binary string

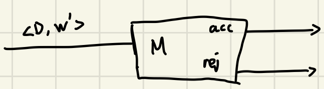
Given any DFA like D and a string like w'
does D accept w

$$\boxed{A_{DFA} = \{ w \mid w = \langle D, w' \rangle \text{ where } w' \in L(D) \}}$$

$$= \{ \langle D_1, w_1' \rangle, \cancel{\langle D_2, w_2' \rangle}, \langle D3, w_3' \rangle \ldots \}$$

M on input w
- if w is not a valid DFA with a string
  - reject
- otherwise
  - w is in the format $\langle D, w' \rangle$
  - Simulate $\langle D \rangle$ on w'
  - if $\langle D \rangle$ accepts w'
    - M should accept $\langle D, w' \rangle$
  - otherwise
    - M should reject $\langle D, w' \rangle$

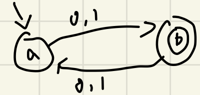
A_{NFA}



D3

$(a), (0,1), (a), (a), ((a,0,a), (a,1,a)), 1010$

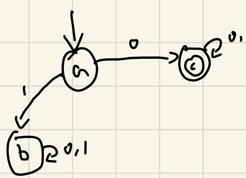
D1



$$\underbrace{(a, b), (0,1), (a), (b), ((a,0,b), (a,1,b), (b,0,a), (b,1,a))}_{\langle D1 \rangle}, \underbrace{110}_{w_1'}$$

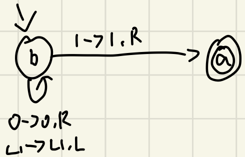
$\langle D1, w_1' \rangle \in A_{DFA}$

D2



$\langle D2, w_2' \rangle \notin A_{DFA}$

$(a, b, c), (0, 1), (a), (c), ((a,0,c), (a,1,b), (b,0,b), (b,1,b), (c,0,c), (c,1,c)), 1111$



(b, a, r), (0, 1, ⊔), (b), (a), ((b, 0, b, 0, R), (b, ⊔, b, ⊔, L), (b, 1, a, 1, R)), 0 0 1

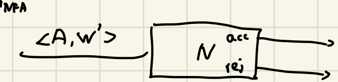
b0 0 1
0 b0 1
0 0 b1
0 0 1 a⊔

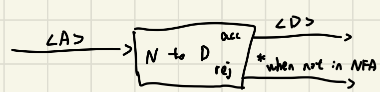
Given any NFA like A and a string like w'
does A accept w'

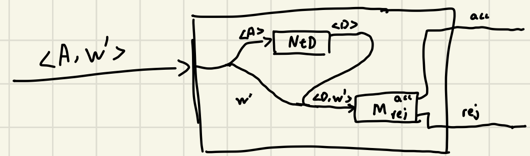
{w | w = <A, w'> where w' ∈ L(A)}

N on input w
if w is not a valid NFA with a string
reject
otherwise w is <A, w'>
simulate <A> on w'
if <A> accept w'
N should accept <A, w'>
otherwise
N reject <A, w'>

A_{NFA}







N on input w
if w is not a valid NFA with a string
reject
otherwise w is <A, w'>
simulate NtD on <A> and get
equivalent DFA <D>
simulate M on <D, w'>
if M accepts <D, w'>,
N also accepts w = <A, w'>
otherwise, N rejects

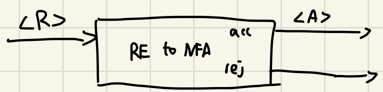
Given any RE like R and some string w, does R generate w

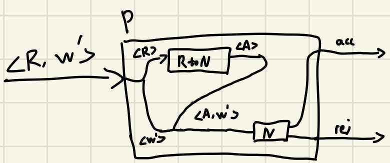
$\boxed{A_{reg} = \{ w \mid w = \langle R, w' \rangle, w' \in L(R) \}}$

$= \{ \langle R_1, 1001 \rangle, \langle R_1, 00 \rangle, \overset{\times}{\langle R_1, \varepsilon \rangle}, \overset{\times}{\langle R_2, 010 \rangle}, \ldots \}$

$R_1 = (0 \cup 1)^* 00 (0 \cup 1)^*$

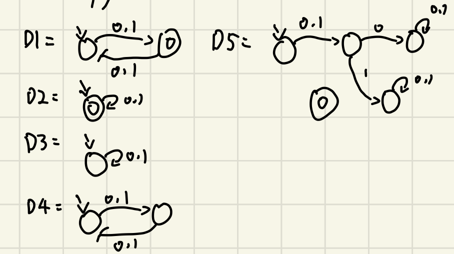
$R_2 = ((0 \cup 1)(0 \cup 1))^*$





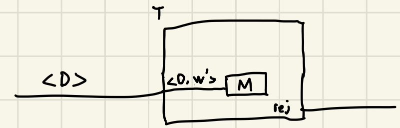
$\boxed{\begin{array}{l}
P \text{ on input } w \\
\quad \text{if } w \text{ not in } \langle R, w' \rangle, \\
\quad\quad P \text{ reject} \\
\quad \text{Otherwise,} \\
\quad\quad w \text{ is in the form } \langle R, w' \rangle \\
\quad\quad \text{Simulate RtN on } \langle R \rangle \text{ to get } \langle A \rangle \\
\quad\quad \text{simulate } N \text{ on } \langle A, w' \rangle \\
\quad\quad \text{if } N \text{ accepts } \langle A, w' \rangle \\
\quad\quad\quad P \text{ also accepts } w = \langle R, w' \rangle \\
\quad\quad \text{otherwise} \\
\quad\quad\quad P \text{ rejects } w
\end{array}}$

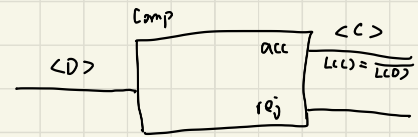
Given a DFA like D, is language of D empty



$\boxed{E_{DFA} = \{ w \mid w = \langle D \rangle, L(D) = \emptyset \}}$

$= \{ \overset{\times}{\langle D1 \rangle}, \overset{\times}{\langle D2 \rangle}, \langle D3 \rangle, \langle D4 \rangle, \langle D5 \rangle \ldots \}$





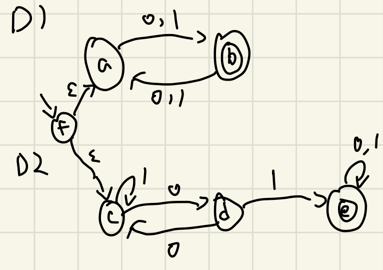
$\boxed{\begin{array}{l}
T \text{ on input } w \\
\quad \text{if } w \text{ not in form } \langle D \rangle \\
\quad\quad T \text{ reject } w \\
\quad \text{otherwise } w \text{ is in form } \langle D \rangle \\
\quad\quad \text{mark start state} \\
\quad\quad \text{until no new state is marked} \\
\quad\quad\quad \text{mark any state that has an} \\
\quad\quad\quad \text{incoming transition from a state} \\
\quad\quad\quad \text{that is already marked} \\
\quad\quad \text{if any final state is marked,} \\
\quad\quad\quad T \text{ reject } \langle D \rangle \\
\quad\quad \text{otherwise} \\
\quad\quad\quad T \text{ accept } \langle D \rangle
\end{array}}$

loops on purpose

not decidable

obsolete

for all binary string like w'
simulate M on $\langle D, w' \rangle$
if M accepts $\langle D, w' \rangle$
T should reject $w = \langle D \rangle$ (language not empty)
otherwise M rejects
jump to loop
if no binary string accepted by $\langle D \rangle$
T accept $\langle D \rangle$

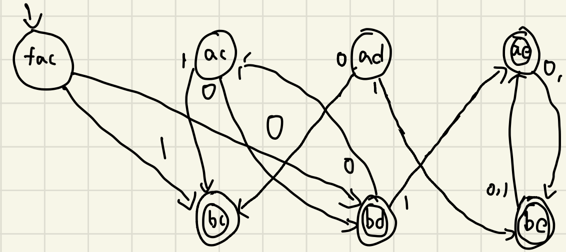


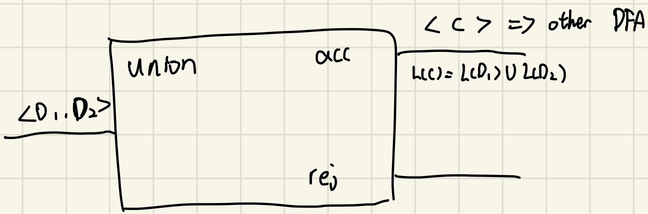
D3

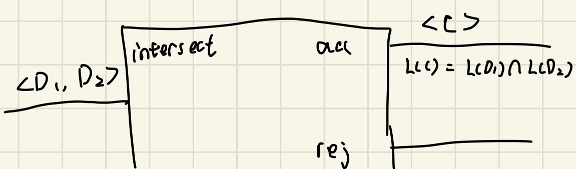
$L(D3) = L(D1) \cup L(D2)$

	0	1
{f,a,c}	{b,d}	{b,c}
{b,d}	{a,c}	{a,e}
{b,c}	{a,d}	{a,c}
{a,c}	{b,d}	{b,c}
{a,e}	{b,e}	{b,e}
{a,d}	{b,c}	{b,e}
{b,e}	{a,e}	{a,e}

<C>

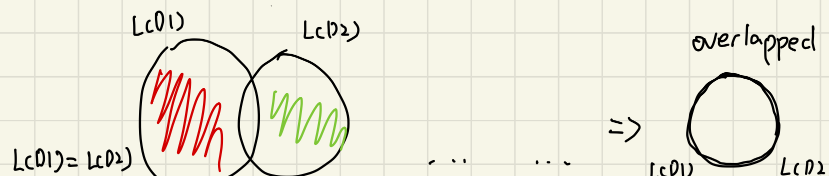




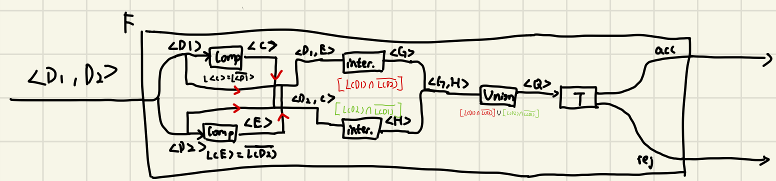


exact same, but different final state

$EQ_{DFA} = \{ w \mid w = \langle D_1, D_2 \rangle,\ L(D_1) = L(D_2) \}$



$[L(D1) \cap \overline{L(D2)}] \cup [L(D2) \cap \overline{L(D1)}]$ ... $= \phi$



F on input w

if w not 2 DFAs

reject w

otherwise, w is in form 2 DFAs

simulate Comp. on input $\langle A \rangle$ to get $\langle C \rangle$ $L(C) = \overline{L(A)}$

. . . . input $\langle B \rangle$ . . . $\langle D \rangle$ $L(D) = \overline{L(B)}$

. . . . intersect on $\langle A, D \rangle$ . . . . $\langle E \rangle$ $L(E) = L(A) \cap L(D) = L(A) \cap \overline{L(B)}$

. . . . $\langle B, C \rangle$ . . . . $\langle G \rangle$ $L(G) = L(B) \cap L(C) = L(B) \cap \overline{L(A)}$

. . . . union . . $\langle E, G \rangle$ . . . $\langle H \rangle$ $L(H) = L(E) \cup L(G)$

. . . . T . . . $\langle H \rangle$

if T accepts $\langle H \rangle$

F accept w

otherwise T rejects $\langle H \rangle$

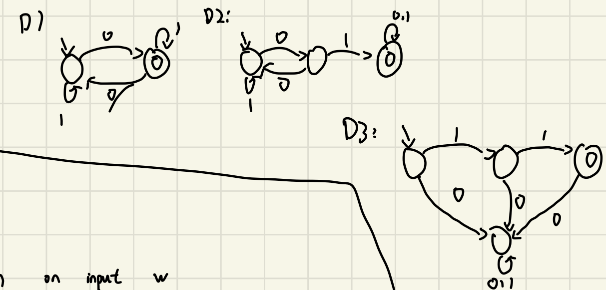
F reject w

Given any DFA like D, does D accept string with substring 01

$A = \{ w \mid w = \langle D \rangle,$ D accepts string with substring 01 $\}$

$= \{ \langle D1 \rangle, \langle D2 \rangle, \cancel{\langle D3 \rangle} \ldots \}$

any language that accepts at least 1 string with 01

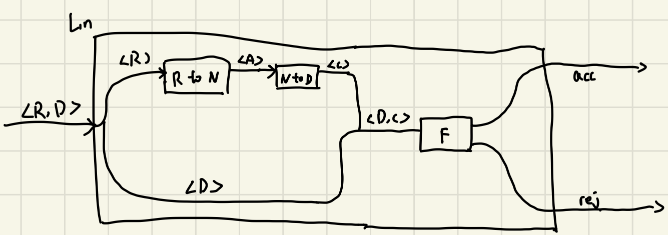


doesn't exist

Given any RE like R, and any DFA like D,

is $L(R) = L(D)$?

$EQ_{R,D} = \{ w \mid w = \langle R, D \rangle, L(R) = L(D) \}$



Lin on input w

if w not in form, reject

otherwise it is in form $\langle R, D \rangle$

simulate RtN on $\langle R \rangle$, get $\langle A \rangle$

. . . NtD on $\langle A \rangle$, get $\langle C \rangle$

. . . . F on $\langle D, C \rangle$

if F accepts $\langle D, C \rangle$

Lin accepts w

otherwise

Lin reject w

prove A is Turing-recognizable

K on input w

if w not $\langle D \rangle$, reject

otherwise w is $\langle D \rangle$

for all bin. string with 01 like w'

simulate M on $\langle D, w' \rangle$

if M accepts $\langle D, w' \rangle$

H accept $\langle D \rangle$

otherwise infinite loop

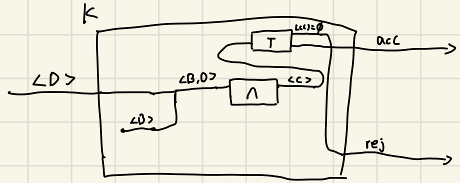
go back to loop

if D doesn't accept any w'

H reject $\langle D \rangle$

$\Leftarrow$ prove A is decidable / computable

K on input w
- if w not in <D>,
  - reject
- otherwise
  - let <B> = (a, b, c), (a), (c), ((a, 1, a), (a, 0, b), (b, 1, c), (b, 0, a), (c, 0, c), (c, 1, c))
  - simulate $\cap$ on <D, B> to get <C>
  - simulate T on <C>
  - if T accepts <C>
    - K reject w
  - otherwise
    - K accepts w



Given any CFG like G
is $L(G) = \phi$

$E_{CFG} = \{w \mid w = \langle G \rangle, L(G) = \phi\}$

L on input w:
- if w not in <G>,
  - reject
- otherwise
  - mark any terminal variables
  - repeat until no new variable marked
    - mark any variables with rule like $V \to u_1, u_2, u_3 \ldots u_k$ where u_i is marked
  - if start variable is marked
    - L should reject w
  - otherwise
    - L accept w

not empty

$S \to T \mid F$
$T \to BB\sigma BB$
$B \to \sigma \mid x$
$F \to \sigma A \sigma \mid xAx \mid \sigma \mid x$
$A \to \sigma A \mid xA \mid x$

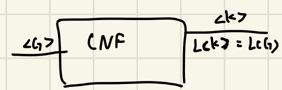
empty

$S \to AB \mid \phi \mid E$
$A \to \phi\phi \mid F$
$B \to CD$
$C \to xC \mid x$
$D \to \phi D \mid D$
$E \to \phi S \mid xS$

Given any CFG like <G> and a string like w',
is $w' \in L(G)$

$B = \{w \mid w = \langle G, w' \rangle, w' \in L(G)\}$

S on input w:
- if w not in <G, w'>
  - reject
- otherwise ...
  - simulate CNF on <G> to get <k>
  - for all $2n-1$ step derivations d
  - (n = length of w')
  - if d generates w'
    - S accepts w
  - otherwise
    - go back to loop
  - if no derivation generate w'
    - S reject w



for all derivation like d
- if d results in w',
  - S accept w
- otherwise
  - go back to loop
- if no derivation generate w'
  - S reject w

infinite loop

$\{w \mid w = \langle G \rangle, L(G) \neq \Sigma^*\}$

language of $\langle G \rangle$ is not everything

only $S \to 0S \mid 1S \mid \varepsilon$ is not in here

prove it's recognizable

L on input w

if w ...

reject

else

for binary string like w'

simulate S on $\langle G, w' \rangle$

if S rejects $\langle G, w' \rangle$

L accept $\langle G \rangle$

else

loop

not ok for decidable

it's ok to loop for recog.

$A_{TM} = \{w \mid w = \langle M, w' \rangle, w' \in L(M)\}$ recognizable

A on input w:

if w ...

reject

else

simulate M on w'

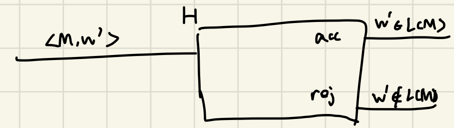
runs into a problem if TM loops

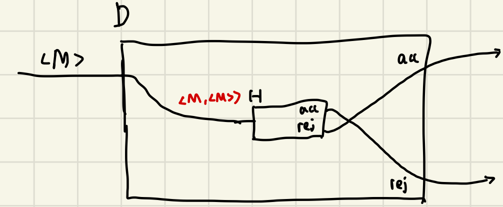
if M accept w'

A accept $\langle M, w' \rangle$

else

A reject

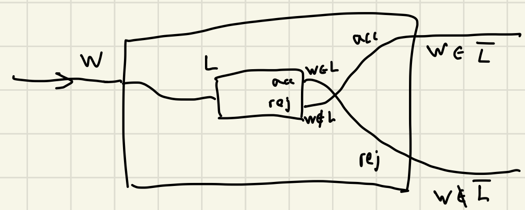




WEIRD AF

if language L is decidable,

L is recognizable, also the same with $\bar{L}$



HALT = $\{w \mid w = \langle M, w' \rangle$, M halts on w'$\}$

accept or reject

prove undecidable : assume decidable

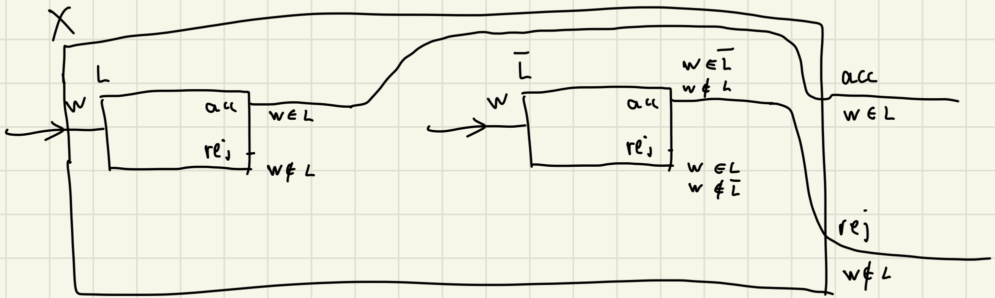
there exist a machine

that reject when M loops on w'

and accept when M halts on w'

if language L is recognizable, and $\bar{L}$ is recognizable

L is decidable,



X on input w

- simulate L and $\bar{L}$ on w
- simulate L and $\bar{L}$ nondeterministically (same time)

  if L accept w, X accept w

  if $\bar{L}$ accept w, X reject w

or X on input w

simulate L on w

if L accept w, X accept w

if L reject

  simulate $\bar{L}$ on w

    if $\bar{L}$ accept w, X reject w

    elif $\bar{L}$ reject w, X accept w

Time Complexity

L

accept $w \in L$

reject / loop $w \notin L$

recognizable

L

accept $w \in L$

reject $w \notin L$

decidable

Big O h no

$n^2 + 5n$ $O(n^2)$

$1000n + n^3$ $O(n^3)$

$(n+4)(n-7)$ $O(n^2)$

$f(n) = $ # of digits in n

$n = 5432$ $f(5432) = 4$

230 $= 3$

5 $= 1$

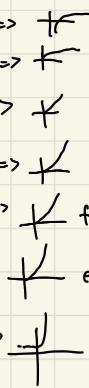
$f(n)$ growing similarly to $\log(n)$

$\Rightarrow O(\log(n))$

$f(n) = 10n^3 \cdot \log(10n^5) + 100n^2 + 1000$

$O(n^3 \log(n^5)) \Rightarrow O(n^3 \log(n))$

$\log n^5 = \cancel{5} \log n$ constant not shown in O



$O(1) \Rightarrow y = 1$

$O(\log n) \Rightarrow$

$O(\sqrt{n}) \Rightarrow$

$O(n) \Rightarrow$

$O(n \log n) \Rightarrow$

$O(n^2) \Rightarrow$ faster

$O(n^3) \Rightarrow$ even faster

$O(2^n) \Rightarrow$ formula 1 fast

hard to write $\log(n)$ algorithms

because input has size n, reading input needs $O(n)$

Class P

algorithms that run on polynomial time on a deterministic TM

if single $\Rightarrow O(n^3)$ $O(n^6)$ $O(n^{10})$ $O(3^n)$

slightly faster multi $\Rightarrow O(n^{1.5})$ $O(n^3)$ $O(n^5)$ $O(2^n)$

assume $n = 100\,000$

and processor of 10^6 operation ps

$\log N$ is fastest

while 2^n is slowest

assume

algo to decide $0^k 1^k$, $k \geq 0$

... steps ...

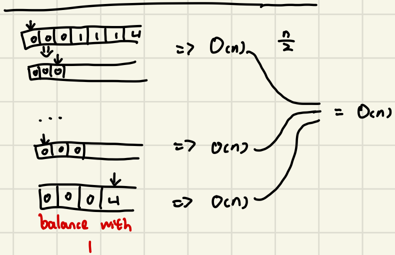
1) scan tape $\Rightarrow O(n)$

2) repeat ... $\Rightarrow O(n)$ } $\Rightarrow O(n^2)$

3) mark 0/1 $\Rightarrow O(n)$

4) check leftover $\Rightarrow O(n)$

$O(n) + \underline{O(n^2)} + O(n) \Rightarrow O(n^2)$



Class P

algorithms that run on polynomial time on a deterministic TM

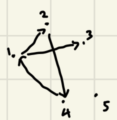
$P = \bigcup_k TIME(n^k)$

give any directed graph like G and 2 nodes in G like s and t, is there a path from s → t

$A = \{ <G1, ~~1, 5~~ >, <G1, 1, 4> \ldots \}$

$= \{ w \mid w = <G, s, t>,$ there exists a path between s and t in directed graph G $\}$

G1



path : route on graph that doesn't revisit a vertex

consider worst case of time complexity

time complex. $= O(n^3)$

A on input w:

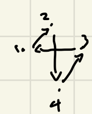
- if w ..... reject — $O(n)$
- else
  - mark s — $O(1)$
  - repeat until no new vertex — $O(n)$
    - check all edges like (a, b), if a is marked but not b, mark b — $O(\frac{n(n-1)}{2}) \Rightarrow O(\frac{1}{2}n^2 - \frac{1}{2}n) \Rightarrow O(n^2)$ $\}$ $O(n^3)$
  - if t is marked
    - A accept w — $O(1) / O(n)$
  - else
    - A reject w

hamiltonian path
- visits all vertex
- visits only once

given any directed graph, and 2 nodes in graph like s and t, is there a hamiltonian path from s to t

$B = \{ <G1, 1, 3>, <G1, 2, 4>, \ldots \}$
✓ ✗

G1



B on input w — $O(n!)$

- $O(n)$ { if w ..... reject / else
- $O(n!)$ ⟵ generate all permutation of vertexs (node)

a route that can be found from s → t

sequences : $S_1, S_2 \ldots S_{n!}$

for i=1 to n! — $O(n!)$

- $O(n^3)$ ⟵ if S_i is a valid hamil. path → accept

if all sequence fails reject — $O(1)$

let $C = c_1, c_2, c_3, \ldots c_n$

verify $c_1 = s$ $O(1)$

$c_n = t$ $O(1)$

for $i = 1$ to $n-1$ $O(n)$

check there is an edge in graph $O(n^2)$ from c_i to c_{i+1}

$= O(n^3)$

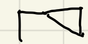
test if c_i is a clique

c_i is a subgraph

make sure c_i has all edges connecting k nodes to each other

total # of edges in c_i $\Rightarrow \frac{k(k-1)}{2} \Rightarrow O(k^2)$

$k = 5$



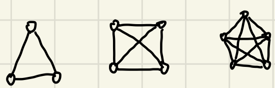
$\Rightarrow \frac{5 \times 4}{2} = 10$ edges

Class NP (nondeterministic polynomial)

algorithms that run on polynomial time on a nondeterministic TM

clique problem

- every node on the clique is connected to every other node



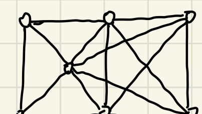
Given any undirected graph G, and input k

is there a k-clique subgraph in G

$C = \{ \langle G1, 3 \rangle, \langle G1, 4 \rangle, \langle G1, 5 \rangle, \langle G1, 6 \rangle \ldots \}$

X

G1



C on input w

$O(n)$ if w not ... reject

else

$[1 \leq i \leq \frac{n!}{(n-k)!k!}]$

$C(n,k) = \frac{n!}{(n-k)!k!} = O(n!)$

repeat for all subgraphs of size k, c_i

$O(k^2)$ test if c_i is a clique

if c_i is a k-clique, accept

$O(1)$ if all subgraph failed, C reject

one-by-one $O(n!) \cdot O(k^2) = O(n!) \cdot O(n^2)$

nondeterministically $O(k^2) = O(n^2)$

if $k = 5$,

$\underline{7}, \underline{6}, \underline{5}, \underline{4}, \underline{3} \Rightarrow \frac{7!}{(7-5)! \cdot 5!}$

$_7C_5$

SAT $\in$ NP

Satisfiability

given any logical proposition like T
, is T satisfiable

$G = \{$ 3, 3, 1, -2, 0, 2, -3, 0 3, -1, 0 $\}$ seperator

$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ —> satisfiable when p, q, r = T

3 comp.
3 variable
p, q, r = 1, 2, 3

p	q	$(p \vee \neg q) \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

$\Rightarrow (\neg p \vee \neg q) \wedge (\neg p \vee q) \wedge (p \vee \neg q)$

$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ -> satisfiable when p=T and q=F
2, 3, 1, 2, 3, 0, -1, -2, -3, 0

G on input w,
if w is not a prop.
reject
else ...

$(1 \le i \le i^2)$

$O(2^n)$
repeat for all possible T/F assignments on variables, C_i
test if C_i satisfies w
if accept, G accepts
if all T/F values assign. fails on w
G reject

$(p \vee q \vee \neg r) \wedge (\neg p \vee r)$ NP complete
satisfiable ?

p -> T
q -> ?
r -> T

✓

default ← p cnf {# of set} {# of symbols}

first set $p \vee \neg q$ => 1 -2 0
second set $p \vee \neg r$ => 1 -3 0
... $r \vee \neg p$ => 3 -1 0

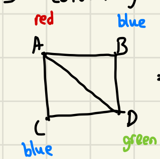
returns p, q, r values so is true

example

$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$

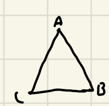
p cnf 5 3
1 -2 0
2 -3 0
3 -1 0
1 2 3 0
-1 -2 -3 0

3 - coloring



=> ✓

color the graph such that
no nodes that are connected
have same color



Ared, Ablue, Agreen 1, 2, 3
Bred, Bblue, Bgreen => 4, 5, 6
Cred, Cblue, Cgreen 7, 8, 9
3 nodes => 12 boolean

applies to other
$(\sim \ldots \vee \sim \ldots)$

$\sim(1 \wedge 2)$ $\sim(1 \wedge 3)$ $\sim(2 \wedge 3)$ ...

$(1 \vee 2 \vee 3) \wedge (\sim 1 \vee \sim 2) \wedge (\sim 1 \vee \sim 3) \wedge (\sim 2 \vee \sim 3) \wedge$
$(4 \vee 5 \vee 6) \wedge (\sim 4 \vee \sim 5) \wedge (\sim 4 \vee \sim 6) \wedge (\sim 5 \vee \sim 6) \wedge$
$(7 \vee 8 \vee 9) \wedge (\sim 7 \vee \sim 8) \wedge (\sim 7 \vee \sim 9) \wedge (\sim 8 \vee \sim 9) \wedge$
$(\sim 1 \vee \sim 4) \wedge (\sim 2 \vee \sim 5) \wedge (\sim 3 \vee \sim 6) \wedge$
$(\sim 4 \vee \sim 7) \wedge (\sim 5 \vee \sim 8) \wedge (\sim 6 \wedge \sim 9) \wedge$
$(\sim 1 \vee \sim 7) \wedge (\sim 2 \vee \sim 8) \wedge (\sim 3 \wedge \sim 9)$

p cnf 9 21

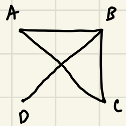
1 2 3 0
-1 -2 0
-1 -3 0
-2 -3 0
4 5 6 0
-4 -5 0
-4 -6 0
-5 -6 0
7 8 9 0
-7 -8 0
-7 -9 0
-8 -9 0

edge constraint

-1 -4 0
-2 -5 0
-3 -6 0
-4 -7 0
-5 -8 0
-6 -9 0
-1 -7 0
-2 -8 0
-3 -9 0

take home

a: b c
b: a c d
c: a b
d: b



Ar	Ag	Ab
Br	Bg	Bb
Cr	Cg	Cb
Dr	Dg	Db

1	2	3
4	5	6
7	8	9
10	11	12

p cnf 12 28

a
1 2 3 0
-1 -2 0
-1 -3 0
-2 -3 0

b
4 5 6 0
-4 -5 0
-4 -6 0
-5 -6 0

c
7 8 9 0
-7 -8 0
-7 -9 0
-8 -9 0

d
10 11 12 0
-10 -11 0
-10 -12 0
-11 -12 0

vertex

16

edge

a->b
-1 -4 0
-2 -5 0
-3 -6 0

b->c
-4 -7 0
-5 -8 0
-6 -9 0

a->c
-1 -7 0
-2 -8 0
-3 -9 0

b->d
-4 -10 0
-5 -11 0
-6 -12 0

12