September 26, 201?

$\boxed{\text{LAST CLASS}}$ CONTEXT-FREE GRAMMAR

start variable

$\boxed{A} \to A+B \mid B$

$B \to B \times C \mid C$

$C \to (A) \mid a$

unambiguous

$a+a\times a: \; A \Rightarrow A+B$
$\Rightarrow A+B\times C$
$\Rightarrow A+B\times a$
$\Rightarrow A+C\times a$
$\Rightarrow A+a\times a$
$\Rightarrow B+a\times a$
$\Rightarrow C+a\times a$
$\Rightarrow a+a\times a$
$\Downarrow$

HW #3 (4b)

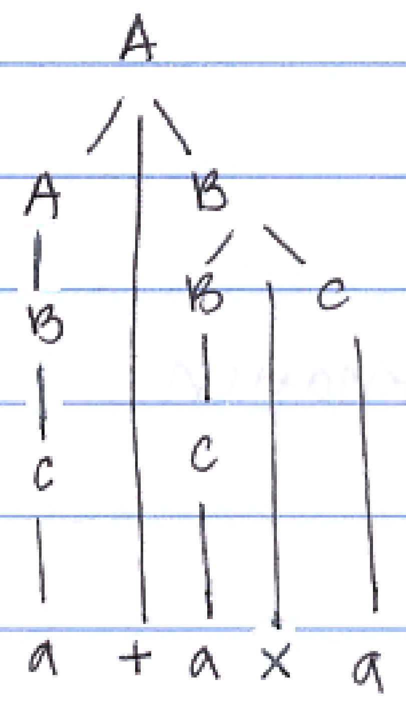
$L = \{x \mid x \text{ is not a palindrome}\}$

$\overline{L} = \{x \mid x \text{ is } \cancel{\text{not}} \text{ a palindrome}\}$

$s = 0^p 1^p$

$0^p 1 0^p 1$

Parse tree (con't example)



terminals

Ambiguity for grammars

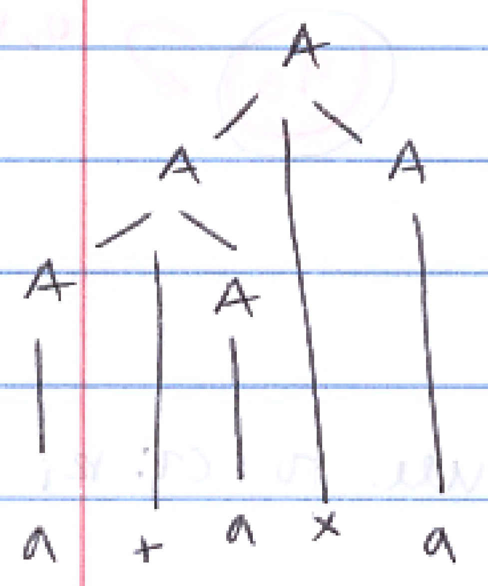
Def -2.7 - A string w is ambiguously derived in CFG G is w has atleast 2 different "left-most derivations". (always pick left most variable to replace.

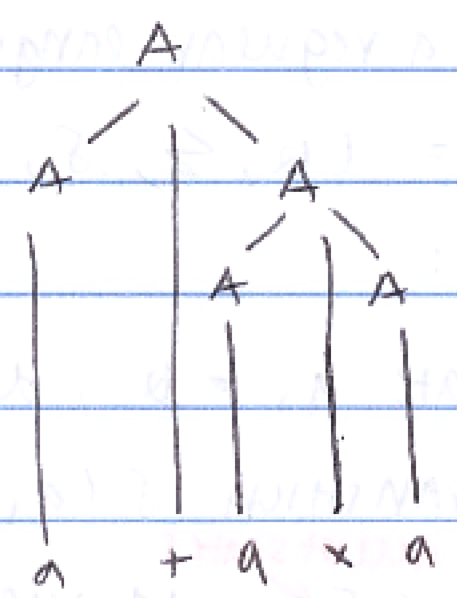
G is ambigous if it generates some string ambiguously.

NOTE: some languages cannot be generated by non-ambiguous grammar! (Prob 2.29)

Example: G_5 $A \to A+A \mid A\times A \mid (A) \mid a$

$a+a\times a$ derived ambiguously in G_5 $\Rightarrow$ 2 diff parse trees.





$A \to A\times A \to A+A\times A$.

closure properties

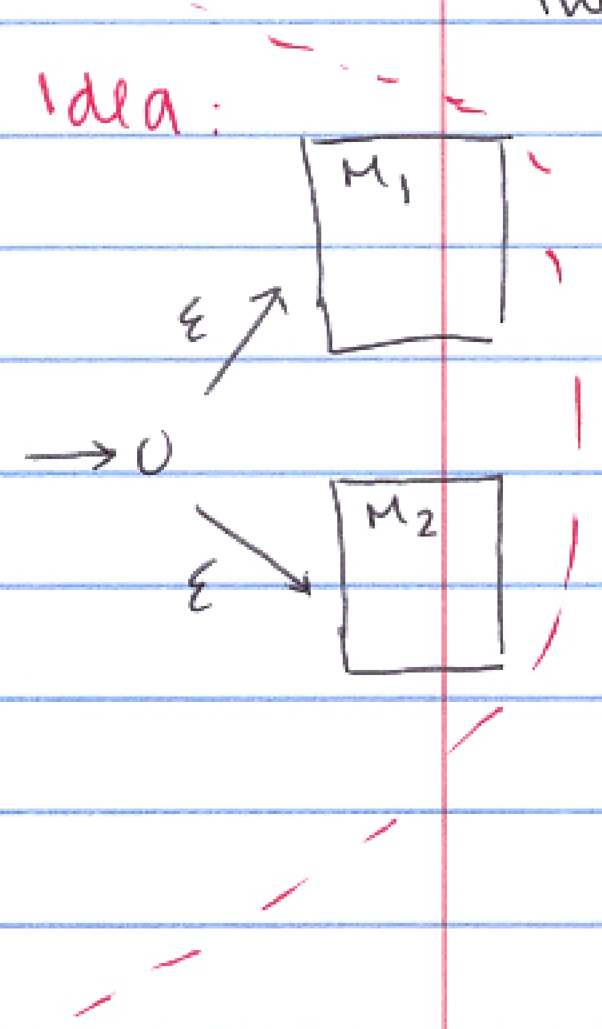
<u>lemma</u>: CFLs are closed under union

Pf/ Let G_1 \$ G_2 be CFG's generating CFL's $L(G_1)$ \$ $L(G_2)$ respectively

let S_1 \$ S_2 be start variable for G_1 \$ G_2, respectively

Then CFG for $L(G_1) \cup L(G_2)$ is:

Idea:



$S \to S_1 \mid S_2$

$\boxed{G_1}$

$\boxed{G_2}$

<u>lemma</u>: CFLs are closed under concatenation.

Pf/ same setup as union.

$S \to S_1 S_2$

$\boxed{S_1 \to \ldots}$ G_1

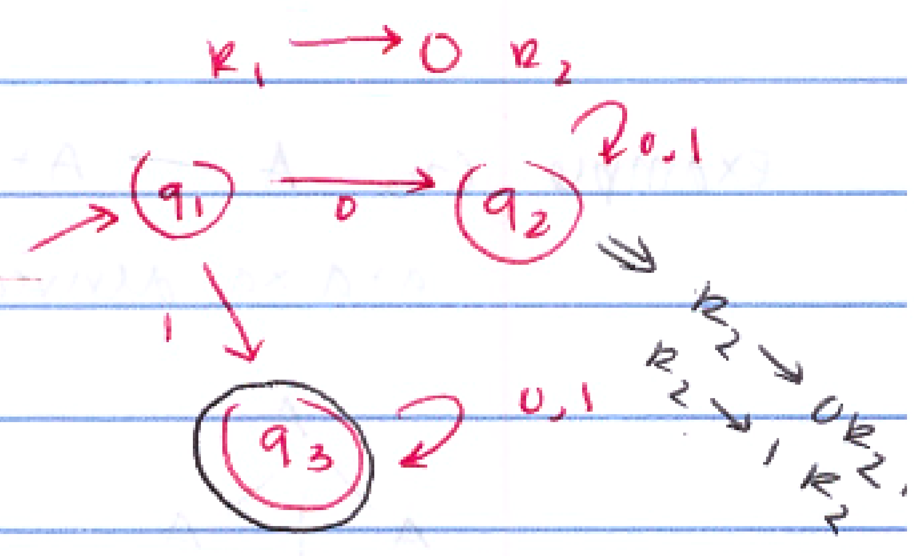
$\boxed{S_2 \to \ldots}$ G_2

<u>Lemma</u>: CFLs closed under star.

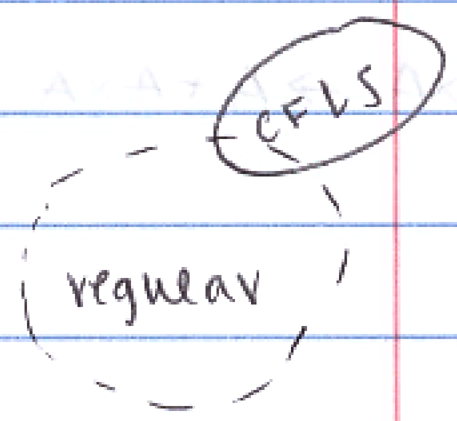
Pf/ Given CFG G_1 with start var S_1

$S \to S_1 S \mid \varepsilon$

$\boxed{S_1 \ldots}$ G_1



<u>lemma</u> All languages are context-free!



Pf/ Let L be a regular language with

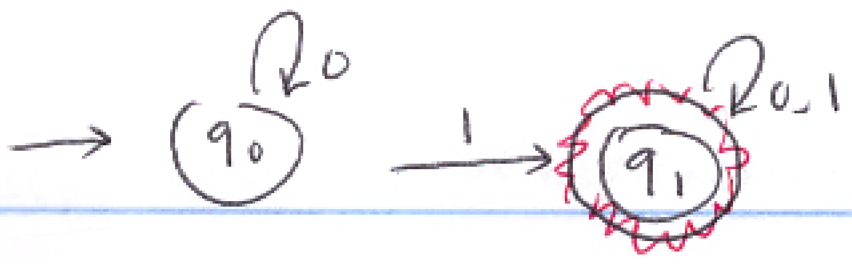
DFA $M = (Q, \Sigma, \delta, q_0, F)$

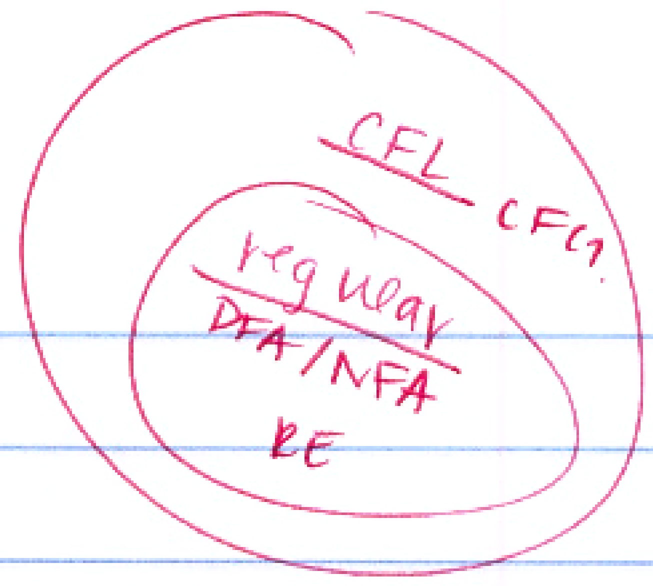
Define CFG as:

① For each state $q_i \in Q$, define variable R_i in G.

② For each transition $\delta(q_i, a) = q_j$, add sub. rule to G: $R_i \to a R_j$

③ For all $q_i \in F$ (accept states), add rule $R_i \to \varepsilon$

e.g. 



G: $R_0 \to 0R_0 \mid 1R_1$

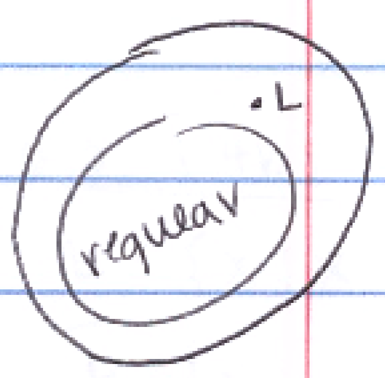
$R_1 \to 0R_1 \mid 1R_1 \mid \cancel{\varepsilon}$

$L(G) = \emptyset$

Q: Regular = CFL ?

Recall: $L = \{0^n 1^n \mid n \geq 0\}$ not regular}

But L is a CFL $\quad S \to 0S1 \mid \varepsilon$



2.2

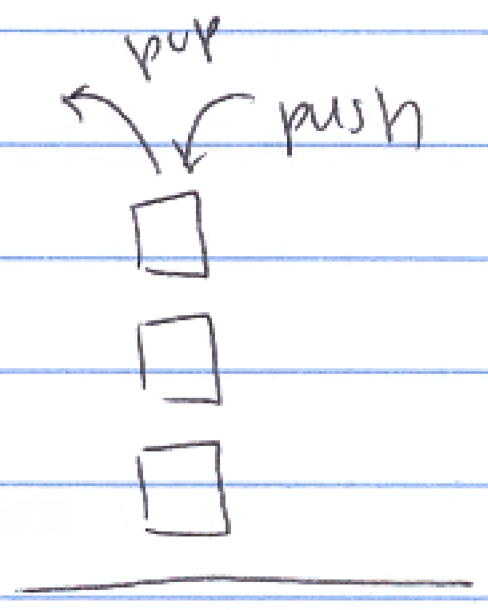
Pushdown Automata

DFAs lack memory!

LIFO data structure: Stack!

A pushdown automata (PDA)

$\mapsto$ a NFA with a stack!



How to recognize with PDA?

Idea: (1) If see 1 first, reject!

(2) Each time read 0, push stack $\quad \begin{array}{c}0\\0\end{array}$

(3) Each time read 1, pop a 0 off stack. $\quad \begin{array}{c}\cancel{0}\\0\end{array}$

(4) Accept if stack is empty & all symbols are read.