Making new CF langs out of known CF langs

10/22

Suppose L_1 & L_2 are context-free languages, w/ grammars G_1 & G_2 and PDAs M_1 & M_2.

$L_1 \cup L_2$ is context free. We can create a new grammar G with start var S & prod.

$S \to S_1 \mid S_2$ where S_1 & S_2 are the start vars of G_1 & G_2.

(and the vars of G_1 & G_2 are distinct (or renamed))

$L_1 L_2$ is context-free. Have

$S \to S_1 S_2$.

L_1^* (strings that are concatenations of some # of strings in L_1)

$S \rightarrow S_1 | SS | \lambda$

or

$S \rightarrow S_1 S | \lambda$

$L_1 \cap L_2$ - this is <u>not</u> necessarily context free; we'll see an example next week.

- The proof for reg. langs. doesn't carry over b/c to run 2 PDAs in parallel, you need 2 stacks, while a PDA is only allowed <u>one</u> <u>stack</u>.

It is true that, if L_1 is context-free and L_2 is regular,

$L_1 \cap L_2$ is context-free.

(The proof for reg langs works, b/c L_2 doesn't require a second stack)

$\overline{L_1}$ - the proof for reg langs doesn't work here b/c it requires a deterministic finite automaton; there are context-free langs with no det. PDAs, so that proof won't work

(It does prove that, if L_1 is recognized by a det. PDA, so will $\overline{L_1}$)

We'll also see next week an example where L_1 is CF but $\overline{L_1}$ is not.

$L_1 \setminus L_2 = L_1 \cap \overline{L_2}$ (doesn't work - if $L_1 = \Sigma^*$, $L_1 \setminus L_2 = \overline{L_2}$)

L_1^R - reverse the result of every production in G_1

L_1 / L_2 - I don't think so (unless L_2 is known to be regular)