Reading:

For the week after break, we will finish section 2.1 by covering parse trees, ambiguity, and Chomsky normal form. Also read section 2.2 just through the top of page 113—that is, without going into the headings of the formal definition, examples, or the long proof of equivalence with CFGs. I always skip that proof and we will define pushdown automata as a special case of two-tape Turing machines instead when we hit section 3.2. For the following week, read section 2.3 and be aware that when the grammar is in Chomsky normal form we can do the whole proof with the simplification \( b = 2 \).

This is a shorter problem set. The snow day prevented covering parse trees and giving problems on them before the break. I will also tighten up the coverage of proofs involving context-free grammars—more on this soon.

Assignment 5, due in hardcopy and in class Thu. 3/30

(*except R4 may submit Fri. 3/31 before 2pm) Please staple any multiple-sheet submission.

And: please write your name, Student ID#, and recitation attended atop your HW.

(1) Define \( L \) to be the language of strings over \( \Sigma = \{a, b\} \) that do not begin with \( aa \) and do not end in \( bb \). (Note this is different from the language on Prelim 1 by having \textbf{and} in place of \textbf{xor}.) Find a PD set \( S \) of size 6 for \( L \).

For some procedural hints, first draw a DFA \( M \) such that \( L(M) = L \). If \( M \) has fewer than 6 states, then there must be an error either in your machine or in the above paragraph. If it has at least 6 states, and you’re pretty sure 6 of those states need to be distinct, then take a shortest string reaching each state as a member of \( S \). (For the start state you can take \( \epsilon \) in \( S \).) Note, however, that this does not yet prove that \( S \) is PD, because your giving \( M \) does not yet prove that it can’t be condensed down to 5 states. Make a \( 6 \times 6 \) array with rows and columns labeled by your strings and block out the upper triangle of 6-choose-2 = 15 cells. For each row \( x \) and column \( y \), you need to fill in a string \( z \) such that \( L(xz) \neq L(yz) \). For starters, if \( x \in L \) and \( y \notin L \) or vice-versa, you can take \( z = \epsilon \); this will already fill in over half your boxes. Complete the table, and finally show how it prov(id)es the design of an optimal 6-state DFA. (18 pts. total)

(2) For the following languages \( L_1, L_2 \) over \( \{0, 1\} \), design context-free grammars \( G_1, G_2 \) such that \( L(G_1) = L_1 \) and \( L(G_2) = L_2 \). You need not prove your grammars correct, but as usual you should include a few comments explaining how and why the grammars work correctly. (2 \( \times \) 12 = 24 pts., for 42 total on the set)

1. \( L_1 = \{0^m 1^n 0^n 1^m : m \geq 1, n \geq 0\} \),
2. \( L_2 = \{xy : \#0(x) = \#1(y)\} \).