Prelim II will be on Tuesday, April 20, remotely given in class period. The rules and logistics will be the same as for Prelim I. It will focus on Chapter 2 of the text (Chapter 3 also included, but section 2.4 not included) and the domain of assignments 5–8. But the exam will be “cumulative” insofar as aspects of regular and non-regular languages are involved with CFLs and even Turing machines.

Reading:

Tuesday’s lecture will cover section 4.1 in depth, not just for the idea of describing Turing machines by prose/pseudocode. The following algorithm in C++/Java-like pseudocode is also sometimes used while converting a CFG \( G = (V, \Sigma, R, S) \) into Chomsky normal form.

```plaintext
bool changed = true;
set<V> NULLABLE; //constructed to be the empty set
while (changed) {
    changed = false;
    for (each rule A->X in R) {
        if (X is in NULLABLE*) {
            NULLABLE = NULLABLE U {A};
            changed = true;
        }
    }
}
output NULLABLE;
```

The while loop always terminates because the set NULLABLE can grow by at most the number of variables. This algorithm thus decides the decision problem “given a CFG \( G \), is \( \epsilon \in L(G) \)” — the answer is yes if and only if the final output NULLABLE includes the start symbol \( S \). So the problem is decidable, but Thursday’s lecture will cover the rest of chapter 4 toward showing that some problems are undecidable. Skim the material on uncountability—I cover the diagonal language

\[ D_{TM} = \{\langle M \rangle : \langle M \rangle \notin L(M) \} \]

directly as not even being accepted by any Turing machine, let alone decided by one. Then I show how this implies the undecidability of the Halting Problem and others.

Homework—part online (TopHat), part written, and all individual work:

1. Using TopHat, the “Worksheet” titled S21 HW7 Online Part (10 Qs, 20 pts.)

The other two problems are to be submitted as PDFs using the CSE Autograder system.

2. Draw on paper a diagram of a Turing machine that recognizes the language \( A = \{a^i b a^j : i < j \} \). You may draw either a 1-tape TM \( M_1 \) such that \( L(M_1) = A \) or a 2-tape TM \( M_2 \)
such that \( L(M_2) = A \), but if you take the latter option, you must make \( M_2 \) be a deterministic pushdown automaton (coded as a 2-tape TM). (18 pts.)

(3) Define \( L = \{ x \in \{a, b\}^* : \#a(x) = \#b(x) \) and the two middle chars are “bb” \}. 

(a) Prove using the CFL Pumping Lemma that \( L \) is not a context-free language. (Hint: Consider strings in \( L \) that also belong to \( b^*a*bba*b^* \). 18 pts.)

(b) Sketch in prose (not an arc-node diagram) a 2-tape deterministic Turing machine \( M \) such that \( L(M) = L \) and \( M \) runs on \( O(n) \) time. Describe the operation of \( M \) as a finite sequence of “passes” and say what happens in each pass. Note that by (a), your \( M \) cannot be a pushdown automaton. Indicate exactly where in one (or more) of your passes the machine does something a PDA is not allowed to do. (18 + 3 = 21 pts., for 39 on the problem and 77 on the set)