CS46 Homework 5

This homework is due at **11:59pm on Sunday March 21**. Write your solution using LaTeX. Submit this homework using **github** as a file called **hw5.tex**. This is a **13 point** homework.

For this homework, you will work with a partner. It's ok to discuss approaches at a high level with other students, but most of your discussions should just be with your partner. Your partnership's write-up is your own: do not share it, and do not read other teams' write-ups. If you use any out-of-class references (anything except class notes, the textbook, or asking Joshua), then you **must** cite these in your post-homework survey. Please refer to the course webpage or ask me any questions you have about this policy. The main **learning goal** of this homework is to work with and think about context-free languages (and their limits!), and to practice using (possibly *in combination*) the tools we have accumulated over the past few weeks.

Note: You must submit your solutions in a file named **hw5.tex**, and your submission must compile without errors using `pdflatex`. Any .pdf submissions will be ignored. Any .tex files not named hw5.tex, .tex files that don't compile, or not submitting a post-homework survey will earn up to a **-0.5** point deduction.

Note also the unusual deadline. This homework is a bit larger than normal, but you have extra time until the Sunday before spring break. No late day submissions will be accepted after 5pm Tuesday March 23. (do no CS46 work over break!)

1. The textbook (example 2.38) shows that the language

   $$L = \{ww \mid w \in \{0,1\}^*\}$$

   is not context-free. Prove that $\overline{L}$ is context-free. (Notice: This shows that the context-free languages are *not* closed under complementation!)

   (Note: this language's alphabet does not include #, so it is different from the problem you saw on lab. Be careful! We've seen examples where one character makes a big difference.)

2. (Sipser 2.47) Let $\Sigma = \{0,1\}$ and let B be the collection of strings that contain at least one 1 in their second half:

   $$B = \{uv \mid u \in \Sigma^*, v \in \Sigma^* 1 \Sigma^*, |u| \geq |v|\}$$

   (a) Give a PDA that recognizes B.

   (b) Give a CFG that recognizes B.

   You do not have to give proofs that your constructions are correct, but you should explain your reasoning at a high level. You should definitely *convince yourself* that they are correct by doing the same checks that you would perform if you were writing complete proofs.

3. Using the stack in a PDA can be subtle. One way to see this is examine two languages that are *very* similar, and show that they require different computational power to recognize. (Hint: the difference is going to be in the usage of the stack!)

   Note that:

   - i and j are not necessarily distinct in part (a)
   - any palindrome x satisfies $x = x^R$

- $|x_i|$ can be zero for any i

(a) Give a context-free grammar that generates (or pushdown automata that recognizes) the language:

$$\{t_1\#t_2\#\cdots\#t_k \mid k \geq 1, \text{ each } t_i \in \{a,b\}^*, \text{ and } t_i = t_j^R \text{ for some } i,j\}$$

You do not have to prove the correctness of your grammar/PDA, but you should give a high-level explanation of why/how you designed it, and why both directions of proof should work.

(b) Use the pumping lemma for context-free languages to show that the following language is not context-free:

$$\{t_1\#t_2\#\cdots\#t_k \mid k \geq 2, \text{ each } t_i \in \{a,b\}^*, \text{ and } t_i = t_j \text{ for some } i \neq j\}$$

(c) **(extra credit)** Use closure properties to show that the language from part (b) is not context-free.

4. **(extra credit)** Deterministic PDAs?

In lecture, you've seen that for any NFA there is a DFA that accepts the same language. In this way, deterministic finite automata are just as powerful as nondeterministic finite automata. You've also seen that an NFA with a stack (i.e., a PDA) can compute a context-free language. What happens if we augment a DFA with a stack? Call this a *deterministic pushdown automata*, or dPDA.

- Attempt to modify the construction that converted an NFA into a DFA. If you wanted to convert a PDA into a dPDA, how would this construction need to change? What would you need to do to handle the stack?
- Give a context-free language L that cannot be recognized by any dPDA. Construct a PDA that recognizes L and prove that L cannot be computed by a dPDA.

5. **(extra credit)** Inspired by genetics, define the CROSSOVER operator as follows:

$$\textsc{Crossover}(A,B) = \{x_1y_2, x_2y_1 \mid x_1x_2 \in A, y_1y_2 \in B, \text{ and } |x_1| = |x_2| = |y_1| = |y_2|\}$$

So for every pair of equal-length strings $x_1x_2 \in A$ and $y_1y_2 \in B$, we add two strings to the crossover language by cutting them in half and recombining them.

For example, if $A = \{a, aa, aabb\}$ and $B = \{\varepsilon, cc, caca, aacaa\}$ then $\textsc{Crossover}(A,B) = \{ac, ca, aaca, cabb\}$.

Show that if A and B are regular, then $\textsc{Crossover}(A,B)$ is not necessarily regular.

Show that if A and B are regular, then $\textsc{Crossover}(A,B)$ is context-free.