CS46 Homework 5

This homework is due at 11:59PM on Sunday, February 20. This is a **10 point** homework. This homework has 3 parts:

- part 1 is individual
- part 2 and part 3 will be completed with a partner

General instructions: It's ok to discuss approaches at a high level with other students, but for part 1 your work should be your own, and for parts 2 and 3 your discussions should be just with your partner. Your partnership's write-up and code 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 Lila), then you **must** cite these in your post-homework survey. Please refer to the course webpage or directly ask any questions you have about this policy.

The main **learning goal** of this homework is to work with and think about context-free grammars, and to practice using (possibly *in combination*) the tools we have accumulated over the past few weeks.

1 Automata Tutor problem

This problem (worth 3 points) should be completed on Automata Tutor. You are allowed five attempts. I recommend that you *first* try to solve the problem on paper, *then* use the site to debug your solutions.

Give a context-gree grammar that generates the language

$$\{a^i b^j c^k \mid i = j \text{ or } j = k, \text{ where } i, j, k \geq 0\}$$

You do not have to give a proof of correctness, but you should think about what would be required to write a proof. This will help you plan and debug your grammar. (Note: we will see a PDA for this language in lecture this week.)

2 Twitter bot

This part is worth 2 points, with another possible 5 points for creativity and flair in execution. (Yes, those are 5 additional *real-valued* points. The maximum score on this homework is 15/10.)

Devise an interesting context-free language, and write a grammar for that language. You are free to choose any alphabet you like, and your grammar can be as complicated or elegant as you want. You should read through these instructions, then check out the examples. In this .tex file, give a high-level description of the language generated by your grammar, as well as the rules to fully describe your grammar. Include the handle of your Twitter bot here, too.

Next, you will set up a Twitter bot that periodically generates a word using your grammar and tweets it to the world.

(a) Create a new account on Twitter using `[your username]@cs.swarthmore.edu`. (This email account forwards to your `@swarthmore.edu` account).

(b) Follow the SwarthmoreCS46 bot.

(c) Log in to Cheap Bots, Done Quick!. Set your source code to be visible to the public, and set your bot to tweet at least once a day.

(d) Write your grammar rules using Tracery syntax. Note that the page will let you test your grammar by generating a random string.

Tracery syntax is as follows:

- The start variable is always `origin`.
- Variables of the grammar appear inside double-quotes on the left-hand side of rules, and inside hashtags on the right-hand side of rules.
- The $\rightarrow$ symbol is ":".
- The right-hand side of a rule is a list, in square brackets, of all possible rules for that variable. Items are separated by commas. Each rule must be a string in double quotes. After the list is over, include a comma. For example, the rule $A \rightarrow aA|b$ becomes:

```
"A":["a#A#","b"],
```

If you have questions about the syntax for Tracery, please ask on the forum. You don't need to spend hours debugging Tracery syntax, the purpose of this exercise is to have fun making a Twitter bot.

3 Written problems

These problems (worth 5 points) should be typeset in LaTeX and submitted using **github**.

1. Consider the class of context-free languages. Let G_1 and G_2 be two context-free grammars.

   (a) Show that $L(G_1) \cup L(G_2)$ is context-free, so the class of context-free languages is closed under union. (Use a constructive proof — build a new grammar for the union language.)

   (b) Show that $L(G_1) \circ L(G_2)$ is context-free, so the class of context-free languages is closed under concatenation. (Use a constructive proof — build a new grammar for the concatenation language.)

   (c) Show that $L(G_1)^*$ is context-free, so the class of context-free languages is closed under Kleene star. (Use a constructive proof — build a new grammar for the Kleene star language.)

   (d) Theorem 1.25 proves that the class of regular languages is closed under intersection. Technically, it proves closure under union, but as the footnote in step 5 notes, a slight tweak makes this proof work for intersection, too. Can a similar technique be applied to pushdown automata to show the class of context-free languages is closed under intersection? Explain your answer briefly, but you do not need to give a full proof/counterargument.

2. Consider the grammar G with rules: $\begin{cases} S & \to AS \mid \varepsilon \\ A & \to 0A \mid A1 \mid \varepsilon \end{cases}$

   Show that G is ambiguous.

3. **(extra credit) Disambiguating G.**

   Revisit the ambiguous grammar G (from above) with rules $\begin{cases} S & \to AS \mid \varepsilon \\ A & \to 0A \mid A1 \mid \varepsilon \end{cases}$

   Give an equivalent grammar to G which is not ambiguous. (No proof is required, but you should explain why it's not ambiguous and why it generates exactly the same strings as G.)

4. **(callback extra credit) "Regular expressions" is a context-free language.**

   Regular expressions over alphabet $\Sigma = \{a, b\}$ are just strings over the larger alphabet $\Sigma' = \{a, b, \emptyset, \varepsilon, \cup, \circ, ^*, (,)\}$. (Note that here, '$\varepsilon$' is *a single letter*, being used as a symbol, so $|\varepsilon| = 1$. This is different from our usual usage of ε.) The language of regular expressions is L, where:

   $$L = \{w \mid w \text{ is a regular expression over } \Sigma\}$$

   Prove that L is context-free. (Last week's extra credit was to show it was *not* regular.) Note that you will probably want to use some special symbol to stand in for ε so that it is clear when your grammar is producing *the character* 'ε' and when your grammar is producing *the string* "".

5. **(dangerously extra credit)** Consider the languages from lab this week (practice problems 4) and from last week's homework (lab 4). For any of those languages that **are** context-free and **are not** regular, give a CFG or PDA for that language. (Warning: some of the languages might not even be context-free, so... use your time wisely. We are still learning the tools to deal with all of these.)