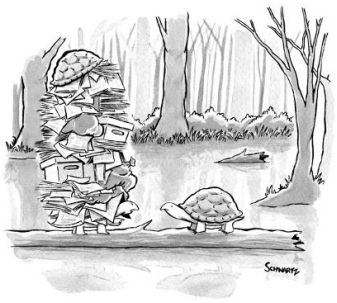
COMP 330 - Fall 2020 - Assignment 3

Due: 11:59pm Oct 23rd.

General rules: In solving these questions you may consult books or other available notes, but you need to provide citations in that case. You can discuss high level ideas with each other, but each student must find and write her/his own solution. Copying solutions from any source, completely or partially, allowing others to copy your work, will not be tolerated, and will be reported to the disciplinary office.
You should upload the pdf file (either typed, or a clear and readable scan) of your solution to mycourses.



"I'm working from home."

Answer only five questions of your choice

---

1. (20 points) Prove that the intersection of a context-free language C and a *regular language* R over the same alphabet Σ is context-free.

2. (20 points) Either prove that the following language is context-free, or prove that it is not context-free. Here $\Sigma = \{0, 1\}$.

$$L = \{ww^R \mid w \in \{0,1\}^*\} \cap \{w \mid w \text{ has the same number of 0's and 1's}\}.$$

3. (20 points) Either prove that the following language is context-free, or prove that it is not context-free. Here $\Sigma = \{a, b, c, d\}$.

   $L = \{w \mid w$ has the same number of a's and b's, and additionally the same number of c's and d's$\}$.

   For example $accddb \in L$.

4. (20 points) Give a description of a Turing Machine that decides the strings of the form "$u < v$" where u and v are two *positive* integers in binary and the string is valid as an inequality. Here the alphabet is $\{0, 1, <\}$. (For example $10 < 100$ is in the language but $11 < 10$ is not; Also note that the leftmost digit of the binary representation of every positive integer is 1.) Explain why your Turing Machine works. Your description can have high level lines such as "Scan the tape to the right, until you see the first 0".

5. (20 points) Prove that a single-tape Turing Machine that is not allowed to write on the input portion of the tape can only recognize regular languages.

6. (20 points) Prove that a PDA that has access to two stacks is as powerful as a Turing Machine. Such a PDA has labels on the transition arrows of the form $(a, \alpha, \beta \to \alpha', \beta')$, meaning that read an a from the input, pop an α from the first stack, β from the second stack, and push α' on the first stack, and β' on the second stack.