CS 3120 Quiz Day 3

This packet contains the quizzes for this quiz day. This **cover sheet** is here to provide instructions, and to cover the questions until the quiz begins. **do not open this quiz packet** until your proctor instructs you to do so.

You will have the 1 hour to complete these quizzes. Each quiz is two pages (front and back of one sheet of paper) worth of questions. Make sure to **write your name and computing id at the top of each individual quiz**.

When you are finished, simply submit this packet at the front of the classroom.

This quiz is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

In theory, there is no difference between theory and practice.
But, in practice, there is.

THIS COVER SHEET WILL NOT BE SUBMITTED. DO NOT PUT WORK YOU WANT GRADED ON THIS PAGE

Quiz - Module 3: Context-Free Languages

Name ____________________

1. [8 points] Answer the following True/False questions.

When generating a string from a *context-free grammar*, no *variables* can remain in the final string	**True**	**False**
$S \rightarrow 0S10S1 \mid \epsilon$ is a working grammar for the language $0^n1^n0^n1^n$	**True**	**False**
ww^R is a *context-free language* but ww is not	**True**	**False**
The behavior (output) of a *pushdown automata* might be changed if it begins to execute with symbols already present on the stack	**True**	**False**
We can use any symbol to mark the bottom of the stack (in class we used \$), but the *PDA* may not work correctly if it already uses that symbol	**True**	**False**
When converting a *grammar* to a *PDA*, we checked that the generated string matched the input by including transitions of the form $x, x \rightarrow A$ for each $x \in Sigma$ and $A \in V$	**True**	**False**
When converting a *PDA* into a *grammar*, we included rules of the form $A_{pq} \rightarrow aA_{rs}b$, but these rules can only be included when the transition from p to r pushes the symbol a and the transition from s to q pops the symbol b	**True**	**False**
Non-deterministic *PDA*s are more powerful than deterministic *PDA*s	**True**	**False**

2. [2 points] Give an *context-free grammar* for the following language: The set of strings w such that the number of a's in w equals the number of b's in w. For this language, $\Sigma = \{a, b\}$. You should try to use as few rules as you can.

3. [2 points] Draw a *pushdown automata (PDA)* for the same language from the previous question (i.e., strings that contain an equal number of a's and b's, $\Sigma = \{a, b\}$). You should try to use as few states as possible.

4. [2 points] The language $A = \{0^i 1^i 0^j 1^j\}$ is *context-free*. Choose a string from this language, and select a portion of it to *pump*. Then, pump the string *twice* and show that the result is still in the language. *Recall that the pumping lemma states that strings can be divided* $w = uvxyz$ *such that* $\forall_{i \geq 0} uv^i xy^i z \in A$ *and* $|vxy| \leq p$.

5. [2 points] Suppose you have a *Context-Free Grammar* C and a *Regular Language* R. Consider the language $L = C \cap R$. Is it possible that $C \cap R$ is a *regular language*? If so, provide a concrete example. If not, briefly discuss why.

6. [4 points] Prove or disprove the following claim: If C is *context-free* and R is *regular*, then $L = C \cap R$ is *context-free*. *Hint: Consider the PDA for* C *and the DFA for* R *and construct a PDA for* $L = C \cap R$

First, describe how you will construct the set of states Q for the *PDA* of L? how is the *start state* chosen and how are the *final states* chosen?

Next, describe how you will decide what transitions to include in this *PDA* for L. Make sure to address when/how stack operations are included.