CS 3120 Quiz Day 1

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 remove this cover sheet** until your proctor instructs you to do so.

You will have the entire class period 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 done, you will come to the front of the room and cut off the staple to this quiz booklet. Afterward, you will discard this cover sheet and submit each quiz separately in a different pile. The proctors will be available at the front of the room to clarify this if you have any questions.

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 1: Proofs, Computers, and Cardinality

Name ______________________________

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

$\{a,b\}^2 = \{\epsilon, a, b, aa, ab, ba, bb\}$	**True**	**False**
Computers are simply any device that transforms strings into other strings.	**True**	**False**
$x \neq y \implies f(x) \neq f(y)$ is one way to define an injective function.	**True**	**False**
We can define a surjective function as $\forall d \in D, \exists c \in C : f(d) = c$	**True**	**False**
Suppose $\lvert\Sigma\rvert = y$. The length of $\lvert\Sigma^x\rvert = x^y$	**True**	**False**
The number of functions $f : \Sigma^* \implies \Sigma^*$ is countably infinite	**True**	**False**
$\mathbb{N} \times \mathbb{N}$ is countably infinite	**True**	**False**
Finite functions always have an injection to $\mathbb{N}$	**True**	**False**

2. [3 points] Explain, in your own words, why giving a *bijection* between sets A and B proves that $|A| = |B|$.

3. [3 points] In your HW, you proved that any connected graph G with $|V| \geq 2$ must have two nodes with equal degree. Read the proof below and either explain why it is correct OR explain/fix the error if one exists:

   Suppose for the sake of contradiction that G does not have two nodes of equal degree. G has $|V|$ nodes, and each can have a degree from 0 to $|V| - 1$. By the pigeonhole principle, there are more nodes than possible degrees and thus one pair of nodes must have the same degree.

Suppose I have an arbitrary, countably infinite set $S = \{x_1, x_2, x_3, ...\}$. The next few questions will ask you to prove that the power set $\mathcal{P}(S)$ is uncountably infinite.

4. [3 points] First, explain how/why we can express $\mathcal{P}(S)$ as the set of infinite bit-strings. Specifically, give a bijection between $\mathcal{P}(S)$ and $\{0,1\}^*$.

5. [3 points] Next, use a proof by diagonalization to show the $\{0,1\}^*$ is uncountably infinite (Yes, we did this in class. I am asking you to reproduce that proof).

NOTHING BELOW THIS POINT WILL BE GRADED

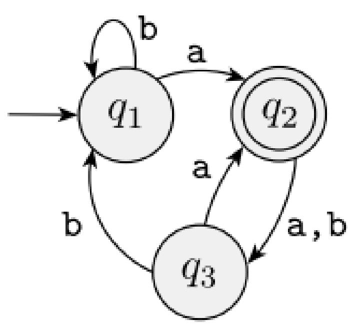
Quiz - Module 2: Regular Languages

Name ______________________________

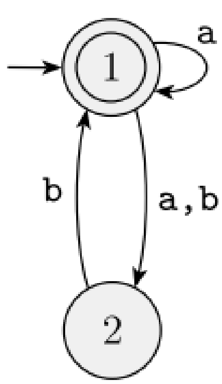
1. [8 points] Answer the following True/False questions.

A finite language is always recognizable by a *DFA*	**True**	**False**
*DFA*s and *NFA*s recognize the exact same set of languages	**True**	**False**
A *DFA* accepts if it ever enters an accept state	**True**	**False**
An *NFA* only accepts if all branches reach an accept state	**True**	**False**
An *NFA* can be in an infinite number of states at once	**True**	**False**
Regular languages are closed under complement (i.e., the set of all strings NOT in the original language)	**True**	**False**
All *NFA*s can be converted into a *regular expression* that describes the same language	**True**	**False**
0^n1^n is not a valid *regular expression*	**True**	**False**

2. [3 points] Give a regular expression for (or describe in words) the language that this *DFA* recognizes.



3. [3 points] Convert the following *NFA* into an equivalent *DFA* using the construction we saw in class. Specifically, the *DFA* should have states representing subsets of states in the *NFA* with appropriate transitions.



These questions are about using the pumping lemma to prove that the language $A = \{0^i 1^j 2^k | i + j = k\}$ is not a *regular language*.

4. [3 points] First, list a string that cannot be pumped (just list it, you will get credit even if you do not know why it cannot be pumped).

5. [3 points] Now, describe precisely why the string cannot be pumped. Make sure to include all possible cases if applicable.

NOTHING BELOW THIS POINT WILL BE GRADED