CS103 Final Exam

This final exam is open-book, open-note, open-computer, but closed-network. This means that if you want to have your laptop with you when you take the exam, that's perfectly fine, but you **must not** use a network connection. You should only use your computer to look at notes you've downloaded in advance. Although you may use laptops, you **must** hand-write all of your solutions on this physical copy of the exam. No electronic submissions will be considered without prior consent of the course staff.

SUNetID: ______________________

Last Name: ______________________

First Name: ______________________

I accept both the letter and the spirit of the honor code. I have not received any assistance on this test, nor will I give any.

(signed) ____

You have three hours to complete this exam. There are 180 total points, and this exam is worth 25% of your total grade in this course. You may find it useful to read through all the questions to get a sense of what this exam contains. As a rough sense of the difficulty of each question, there is one point on this exam per minute of testing time.

Question		**Points**	**Grader**
(1) Discrete Mathematics	(15)	/ 15	
(2) Regular Languages	(40)	/ 40	
(3) Context-Free Languages	(25)	/ 25	
(4) **R**, **RE**, and co-**RE** Languages	(55)	/ 55	
(5) **P** and **NP** Languages	(35)	/ 35	
	(180)	**/ 180**	

It has been a pleasure teaching CS103 this quarter. Good luck on the final exam!

Problem One: Discrete Mathematics (15 Points)

Prove, by induction on n, that for any $n \in \mathbb{N}$ with $n \geq 2$, that

$$\log_2 3 \cdot \log_3 4 \cdot \log_4 5 \cdot \ldots \cdot \log_n (n+1) = \log_2 (n+1)$$

You might want to use the change of base formula for logarithms: for any $c \in \mathbb{R}$ with $c > 1$,

$$\log_a b = \frac{\log_c b}{\log_c a}$$

(Extra space for Problem 1, if you need it)

Problem Two: Regular Languages (40 Points Total)

Suppose that you really, *really* don't like the string **abba** and want to build a language of everything except that string. Let $\Sigma = \{\texttt{a}, \texttt{b}\}$ and consider the language $NOT_{\texttt{abba}}$ defined as follows:

$$NOT_{\texttt{abba}} = \{ w \in \Sigma^* \mid w \neq \texttt{abba} \}$$

For example, $\varepsilon \in NOT_{\texttt{abba}}$ and $\texttt{abbabb} \in NOT_{\texttt{abba}}$, but (unsurprisingly) $\texttt{abba} \notin NOT_{\texttt{abba}}$.

(i) Finite Automata (10 Points)

Design a DFA for the language $NOT_{\texttt{abba}}$.

(ii) Regular Expressions (10 Points)

Write a regular expression for $NOT_{\texttt{abba}}$.

(iii) Nonregular Languages **(20 Points)**

Let $L = \{\mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N}\}$. Prove that no infinite subset of L is a regular language. This result shows that not only is it impossible to write a regular expression for L, but it is also impossible to write a regular expression that matches infinitely many strings from L without also matching at least one string not in L.

(Extra space for Problem 1.iii, if you need it)

Problem Three: Context-Free Languages (25 Points Total)

(i) Context-Free Grammars (15 Points)

Consider the following language L over the alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}\}$:

$L = \{ w \in \Sigma^* \mid |w| \equiv_4 0$, and the first quarter of the characters in w contains at least one **b** $\}$

For example, **baaa** ∈ L, **abbbbbba** ∈ L, **bbbaaabbbaaa** ∈ L, **ababbbbbbbbb** ∈ L, but **abbb** ∉ L, **aabbbbaa** ∉ L, ε ∉ L, **bbb** ∉ L, and **aaabbbbbbbbb** ∉ L. Here, the first quarter of the characters in each string has been underlined.

Write a context-free grammar for L.

(ii) Pushdown Automata **(10 Points)**

Design a PDA for the language L from part (i) of this problem. As a reminder, L is defined as

$L = \{ w \in \Sigma^* \mid |w| \equiv_4 0$, and the first quarter of the characters in w contains at least one **`b`** $\}$

Problem Four: R, RE, and co-RE Languages (55 Points Total)

(i) Properties of Reductions (15 Points)

A language L is called *nontrivial* iff $L \neq \emptyset$ and $L \neq \Sigma^*$.

Prove or disprove: All nontrivial **RE** languages are mapping reducible to one another.
Prove or disprove: All nontrivial co-**RE** languages are mapping reducible to one another.

(Extra space for Problem 4.i, if you need it)

(ii) RE Languages **(15 Points)**

Consider the following language, which consists of all Turing machines that reject at least one string:

$$L_{\mathrm{R1}} = \{ \langle M \rangle \mid M \text{ rejects at least one string } \}$$

Prove that $L_{\mathrm{R1}} \in$ **RE**.

(Extra space for Problem 4.ii, if you need it)

(iii) Unsolvable Problems **(25 Points)**

Prove that $L_{R1} \notin$ co-**RE**. As a reminder, L_{R1} is defined as follows:

$$L_{R1} = \{ \langle M \rangle \mid M \text{ rejects at least one string } \}$$

(Extra space for Problem 4.iii, if you need it)

Problem Five: P and NP Languages (35 Points Total)

(i) Mutual Reducibility (10 Points)

Prove that for any language L, L is **NP**-complete iff $L \leq_P$ 3SAT and 3SAT $\leq_P L$.

(ii) One-Way Functions **(5 Points)**

Are there any functions that are known to be one-way functions? If so, give an example of one and briefly explain why it is a one-way function. If not, explain why not.

(iii) Resolving P $\stackrel{?}{=}$ NP **(20 Points)**

Suppose that there is a polynomial-time verifier V for an **NP**-complete language L. That is,

$$w \in L \quad \text{iff} \quad \exists x \in \Sigma^*.\ V \text{ accepts } \langle w, x \rangle$$

and

$$V \text{ runs in time polynomial in } |w|$$

Now, suppose that V is a "superverifier" with the property that for any string $w \in L$, V accepts $\langle w, x \rangle$ for *almost* all choices of x. Specifically, for any $w \in L$, there are at most five strings x for which V rejects $\langle w, x \rangle$.

Under these assumptions, prove that **P** = **NP**.

(Extra space for Problem 5.iii, if you need it)