Extra Credit: Practice CS103 Final Exam

This practice exam is worth **5 extra credit points**. We will not give points based on whether or not your answers are correct, but rather on whether or not you have made a good-faith effort to answer all the questions. On the honor code, we assume that any answers you submit for these problems represent a good, honest effort on your part.

We will **not** release solutions to this practice exam. If you have any questions about it, feel free to stop by office hours or the review sessions on Saturday or Sunday. It is perfectly fine to work on these problems in a group or even to ask questions about them at the review session, but I strongly suggest taking this practice exam under exam conditions.

The 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.

Normally, I would leave extra space between problems so that you would have room to write out your answers, but to save paper I have tried to minimize the amount of blank space in this handout. You do not need to bring extra scratch paper to the final exam, but I would suggest doing so in case you want to try out various solutions to the problems. You will have three hours to complete this final exam. There will be 180 total points, which corresponds to roughly one point per question. The exam will be worth 25% of your total grade in this course.

Question		**Points**	**Grader**
(1) Natural Languages	(15)	/15	
(2) Regular Languages	(30)	/30	
(3) Context-Free Languages	(30)	/30	
(4) **R** and **RE** Languages	(65)	/65	
(5) **P** and **NP** Languages	(40)	/40	
	(180)	**/180**	

(Note that these points are to give a relative sense of the weights on the final exam and have no bearing on extra credit points)

Optional, but due just before you take the final exam.

Problem 1: Natural Languages (15 points total)

Suppose you are watching a movie with nine friends (meaning that there are ten total people present). During the course of the movie, you and your friends speak a total of 40 words to one another.

Prove that at least two people in your group must have spoken exactly the same number of words during the movie.

Problem 2: Regular Languages (30 points total)

Consider the following language over $\Sigma = \{ \texttt{O}, \texttt{E} \}$:

$$PARITY = \{ w \mid w \text{ has even length and has the form } \texttt{E}^n \text{ or } w \text{ has odd length and has the form } \texttt{O}^n \}$$

For example, **EE** $\in$ *PARITY,* **OOOOO** $\in$ *PARITY*, **EEEE** $\in$ *PARITY*, and $\varepsilon \in$ *PARITY*, but **EEE** $\notin$ *PARITY*, **EO** $\notin$ *PARITY,* and **OOOO** $\notin$ *PARITY*.

(i) Regular Expressions (10 Points)

Write a regular expression for *PARITY*.

(ii) Finite Automata (10 Points)

Design a DFA that accepts *PARITY*.

(iii) The Pumping Lemma (10 Points)

Consider the following language over the alphabet $\Sigma = \{\texttt{0}, \texttt{1}\}$:

$$TWICE = \{ ww \mid w \in \Sigma^* \}$$

For example, **0101** $\in$ *TWICE*, **001001** $\in$ *TWICE*, **1111** $\in$ *TWICE*, and $\varepsilon \in$ *TWICE*, but **01** $\notin$ *TWICE*.

Using the pumping lemma for regular languages, prove that *TWICE* is not regular.

Problem 3: Context-Free Languages (30 points total)

(i) Designing CFGs (10 Points)

On Problem Set 5 and 6, you explored the language *ADD* over the alphabet { **1**, **+**, **=** }, which was defined as follows:

$$ADD = \{ \mathtt{1}^m\mathtt{+1}^n\mathtt{=1}^{m+n} \mid m, n \in \mathbb{N} \}$$

Consider the following generalization of *ADD*, which we will call *MULTIADD*, which consists of all strings describing unary encodings of two sums that equal one another. For example:

1 + 3 = 4	would be encoded as	`1+111=1111`
4 = 1 + 3	would be encoded as	`1111=1+111`
2 + 2 = 1 + 3	would be encoded as	`11+11=1+111`
2+0+2+0=0+4+0	would be encoded as	`11++11+=+1111+`
0=0	would be encoded as	`=`

Notice that there can be any number of summands on each side of the **=**, but there should be exactly one **=** in the string; thus **1=1=1** ∉ *MULTIADD*.

Write a CFG that generates *MULTIADD*.

(ii) Designing DPDAs (20 Points)

On Problem Set 5, you showed that the language of all strings over Σ = {**0**, **1**} containing the same number of copies of the substring **01** and **10** was regular by constructing a DFA for it. However, consider the following language:

$$L = \{ w \in \{\mathtt{0}, \mathtt{1}, \mathtt{2}\}^* \mid w \text{ contains the same number of instances of the substrings } \mathtt{01} \text{ and } \mathtt{10} \}$$

The substrings **01** and **10** are allowed to overlap, so **010** and **101** are both in language *L*. Other examples of strings in *L* include **012012102102**, **000121000**, **22**, **0212**, and ε.

Design a (possibly nondeterministic) PDA for the language *L*.

Problem 4: R and RE Languages (65 points total)

(i) Same Difference? (25 Points)

Prove or disprove: If $L_1 \in \mathbf{R}$ and $L_2 \in \mathbf{R}$, then $L_1 - L_2 \in \mathbf{R}$.
Prove or disprove: If $L_1 \in \mathbf{RE}$ and $L_2 \in \mathbf{RE}$, then $L_1 - L_2 \in \mathbf{RE}$.

(ii) Accept Most of the Strings! (20 Points)

Consider the language

$$A_{MOST} = \{ \langle M, n \rangle \mid M \text{ accepts all strings of length at least } n \}$$

Prove that A_{MOST} is undecidable by reducing *HALT* to it.

(iii) Accept Most of the Strings! (Take Two) (20 Points)

Prove that A_{MOST} is unrecognizable by reducing A_{ALL} from Problem Set 8 to it. As a reminder:

$$A_{ALL} = \{ \langle M \rangle \mid \mathscr{L}(M) = \Sigma^* \}$$

Problem 5: P and NP (40 points total)

(i) Closure under Complement (15 Points)

Prove that **P** is closed under complementation. *(Hint: Show how to turn a polynomial-time decider for a language L into a polynomial-time decider for the language* $\overline{L}$*)*

While we know that **P** is closed under complementation, it is unknown whether **NP** is closed under complementation. The class of problems that are the complements of problems in **NP** is an interesting one, and it is so important that we give it the name co-**NP**. Formally, co-**NP** is the set of languages L such that $\overline{L} \in$ **NP**. For example, the language

$$SAT = \{ \langle \varphi \rangle \mid \varphi \text{ is a satisfiable propositional logic formula} \}$$

is known to be in **NP**, while its complement

$$\overline{SAT} = \{ \langle \varphi \rangle \mid \varphi \text{ is an unsatisfiable propositional logic formula} \}$$

is contained in co-**NP**.

Just as the relation between **P** and **NP** is unknown, the relation between **NP** and co-**NP** is also unknown and is a major open problem in complexity theory. However, we do know of one interesting result about how **P**, **NP**, and co-**NP** are connected.

(ii) NP and co-NP (10 Points)

Prove that if **NP** $\neq$ co-**NP**, then **P** $\neq$ **NP**.

(iii) What Do We Know? **(15 Points)**

Below are ten statements, some of which are definitely true, some of which are definitely false, and some of which are not necessarily true or false (either because the truth of the statement depends on the choice of some particular language, or because the statement depends on an open problem such as whether **P** = **NP**). For each of these statements, write a **T** if the statement is always true, an **F** if the statement is always false, and a **?** if with what is provided the statement cannot be definitively shown to be true or false.

If $L \in \mathbf{P}$, then $L \in \mathbf{NP}$. ______

If $L \in \mathbf{NP}$, then $L \in \mathbf{P}$. ______

If L is **NP**-complete and $L' \leq_P L$, then $L' \in \mathbf{P}$. ______

If L is **NP**-complete and $L' \leq_P L$, then $L' \in \mathbf{NP}$. ______

If L is **NP**-complete and $L' \leq_P L$, then $L' \in \mathbf{NPC}$. ______

If 3SAT is decidable in time $O(n^{10})$, then $\mathbf{P} = \mathbf{NP}$. ______

If 3SAT is not decidable in time $O(n^{10})$, then $\mathbf{P} \neq \mathbf{NP}$. ______

There exists an **NP** language that is not in **R**. ______

There exists an **NP-*complete*** language that is not in **R**. ______

There exists an **NP-*hard*** language that is not in **R**. ______