Additional Practice Final 1

This additional practice final is not worth any extra credit points, but should serve as a good review for the final exam. It was given as an actual final exam in Winter 2012 – 2013.

Question		Points	Grader
(1) Discrete Mathematics	(25)	/ 25	
(2) Regular Languages	(40)	/ 40	
(3) Context-Free Languages	(30)	/ 30	
(4) **R**, **RE**, and co-**RE** Languages	(55)	/ 55	
(5) **P** and **NP** Languages	(30)	/ 30	
	(180)	**/ 180**	

Problem One: Discrete Mathematics (25 Points)

There can be many functions from one set A to a second set B. This question explores how many functions of this sort there are.

For any set S, we will denote by 2^S the following set:

$$2^S = \{ f \mid f : S \to \{0, 1\} \}$$

That is, 2^S is the set of all functions whose domain is S and whose codomain is the set $\{0, 1\}$. Note that 2^S does *not* mean "two raised to the Sth power." It's just the notation we use to denote the set of all functions from S to $\{0, 1\}$.

Prove that for any nonempty set S, we have $|2^S| = |\wp(S)|$. You may find the following definition useful: given two functions $f : A \to B$ and $g : A \to B$, we have $f = g$ iff for all $a \in A$, $f(a) = g(a)$. Your proof should work for all sets S, including infinite sets.

Problem Two: Regular Languages (40 Points Total)

(i) Rock, Paper, Scissors (20 Points)

The number of characters in a regular expression is defined to be the total number of symbols used to write out the regular expression. For example, `(a|b)*` is a six-character regular expression, and **ab** is a two-character regular expression.

Let $\Sigma = \{\mathbf{a}, \mathbf{b}\}$. Find examples of all of the following:

- A regular language over Σ with a one-state NFA but no one-state DFA.
- A regular language over Σ with a one-state DFA but no one-character regular expression.
- A regular language over Σ with a one-character regular expression but no one-state NFA.

Prove that all of your examples have the required properties.

(ii) Nonregular Languages (20 Points)

A natural number $n > 1$ is called *composite* iff it can be written as $n = rs$ for natural numbers r and s, where $r \geq 2$ and $s \geq 2$. A natural number $n > 1$ is called *prime* iff it is not composite.

Let $\Sigma = \{ \mathbf{a} \}$ and consider the language $L = \{ \mathbf{a}^n \mid n \text{ is prime} \}$. For example:

- $\varepsilon \notin L$
- $\mathbf{a} \notin L$
- $\mathbf{a}^2 \in L$
- $\mathbf{a}^3 \in L$
- $\mathbf{a}^4 \notin L$
- $\mathbf{a}^5 \in L$
- $\mathbf{a}^6 \notin L$
- $\mathbf{a}^7 \in L$
- $\mathbf{a}^8 \notin L$

Prove that L is not regular. You may want to use the fact that for every natural number n, there is a prime number p such that $p > n$.

*(**A note:** When we gave this problem out on the final exam in Winter, we had covered the pumping lemma for regular languages as a primary tool for showing nonregularity rather than the Myhill-Nerode theorem. Accordingly, the intended solution was shorter than the one from Problem Set 6. That said, make sure you still understand the solution to that problem!)*

Problem Three: Context-Free Languages (30 Points Total)

(i) Context-Free Grammars (20 Points)

Let $\Sigma = \{\mathbf{a}, \mathbf{b}\}$ and let $L = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}$. The complement of this language is the language $\overline{L}$. For example:

- $\mathbf{abb} \in \overline{L}$
- $\mathbf{aab} \in \overline{L}$
- $\mathbf{baab} \in \overline{L}$
- $\mathbf{abab} \in \overline{L}$
- $\varepsilon \notin \overline{L}$
- $\mathbf{ab} \notin \overline{L}$
- $\mathbf{aabb} \notin \overline{L}$
- $\mathbf{aaabbb} \notin \overline{L}$

Write a context-free grammar that generates $\overline{L}$, then give derivations for the four strings listed in the left-hand column.

(Hint: There are several separate cases you need to consider. You might want to design the grammar to consider each of these cases independently of one another.)

(ii) Disjoint Unions (10 Points)

Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ and let L_1 and L_2 be arbitrary context-free languages over Σ. Prove that $L_1 \uplus L_2$ is context-free as well. As a reminder,

$$L_1 \uplus L_2 = \{ \mathbf{0}w \mid w \in L_1 \} \cup \{ \mathbf{1}w \mid w \in L_2 \}$$

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

(i) The Halting Problem (15 Points)

Prove or disprove: for any TMs H and M and any string w, if H is a recognizer for $HALT$ and M loops on w, then H loops on $\langle M, w\rangle$.

(ii) RE Languages (15 Points)

A *palindrome number* is a number whose base-10 representation is a palindrome. For example, 1 is a palindrome number, as is 14941 and 7897987.

Consider the following language:

$$L = \{ \langle n\rangle \mid n \in \mathbb{N} \text{ and there is a number } k \in \mathbb{N} \text{ where } k > 0 \text{ and } nk \text{ is a palindrome number} \}$$

For example, $\langle 106\rangle \in L$ because $106 \times 2 = 212$, which is a palindrome number. Also, $\langle 29\rangle \in L$, because $29 \times 8 = 232$, which is a palindrome number.

Prove or disprove: $L \in \mathbf{RE}$.

(iii) Unsolvable Problems (25 Points)

Consider the following language $DECIDER$:

$$DECIDER = \{ \langle M\rangle \mid M \text{ is a decider} \}$$

Prove that $DECIDER \notin \mathbf{RE}$ and $DECIDER \notin \text{co-}\mathbf{RE}$. We recommend using a mapping reduction involving the language A_{ALL} from Problem Set 8, which is neither **RE** nor co-**RE**. For reference:

$$A_{ALL} = \{ \langle M\rangle \mid M \text{ is a TM and } \mathscr{L}(M) = \Sigma^* \}$$

Problem Five: P and NP Languages (30 Points Total)

(i) Non-NPC Languages (15 Points)

There are exactly two languages in **NP** that we currently know are not **NP**-complete: Ø and Σ*. Prove that Σ* is not **NP**-complete.

(ii) Resolving P $\stackrel{?}{=}$ NP (15 Points)

Suppose that we can prove the following statement:

For every pair of **NP** languages A and B (where neither A nor B is Ø or Σ*), we have $A \leq_P B$.

Under this assumption, decide which of the following is true, then prove your choice is correct.

- **P** is necessarily equal to **NP**.
- **P** is necessarily not equal to **NP**.
- **P** may or may not be equal to **NP**.