American University of Beirut

الجامعة الأميركية في بيروت

Department of Computer Science
CMPS 257 – Theory of Computation
Spring 2003-2004

Final Exam

Date: June 2nd, 3:00 – 5:00pm.
Instructors: Jihad Boulos

ID #: -----------------------------------

This is **NEITHER** an open-book, **NOR** an open-notes exam.

Your exam should have 14 pages, and there are 8 questions totaling 100 points. Your answers should be concise, and when possible should be a list of important points rather than prose. Solve as many problems as you can. I recommend that you start with problems you think look the easiest. I also advise you to spend time on understanding what is being asked by each problem and **budget your time** wisely to be able to solve **all** problems.
Beware, wordy and/or irrelevant answers might reduce your score for that problem. Your answers should be the summary of work done on scratch paper that you do not hand in. Also, do not expect that I will spend much time trying to decipher your hand writing. I will give a ZERO to **illegible** answers. The space allocated for answers should be sufficient. If not, use the back sides of the pages.

For this final, you can consider that SAT, 3SAT, Hamiltonian Cycle, Vertex Cover, Clique, Subset Sum, Partition, 3-Coloring Graph, and the Traveling Salesman Problem are all NP-Complete.

I wish you good work.

	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8
Max Grade	20	6	6	10	10	20	12	16
Your Grade								

Exercise 1 (20 points):

Please circle whether each of the following statements is **True** or **False** and *justify* your answer.

True	**False**	If R is regular and $R \cap X$ is regular, then it is possible that X is not a *CFL*.
True	**False**	If C is a *CFL*, and R is regular, then both $C - R$ and $R - C$ are *CFL*.
True	**False**	The complement of a *CFL* is always Turing-recognizable.
True	**False**	If L is polynomial-time reducible to a finite language, then L is in P.
True	**False**	The number of co-Turing-recognizable languages is countably infinite.
True	**False**	There is no undecidable language L such that both L and $\overline{L}$ are co-Turing-recognizable.
True	**False**	3-SAT is Turing-recognizable.
True	**False**	If an *NP-hard* problem can be solved in polynomial time, then $P = NP$.
True	**False**	There is no undecidable language L such that neither L nor $\overline{L}$ is Turing-recognizable.
True	**False**	A non-deterministic Turing Machine T accepts a language that is infinite. T requires $\Omega(2^n)$ steps before it accepts a string of length n. If $P \neq NP$, it is possible that $L(T) \in P$.

Exercise 2 (6 points):

Give a CFG for each of the following languages:

a. $L_1 = \{a^n b^m : m = 2n + 1 \text{ and } m, n \geq 1\}$
b. $L_2 = \{a^{n+2} b^{m+n+p} a^m b^p : m \geq 1 \text{ and } p \geq 3\}$

Exercise 3 (6 points):

a. Draw a transition diagram for a DFA that accepts the set of binary strings beginning with 010 or ending with 101.

b. Construct a regular expression for the set of all binary strings in which 00 occurs exactly once.

Exercise 4 (10 points):

Prove that the following language is not context free using the context free pumping lemma. You may additionally use any relevant closure properties.

$$L = \{w: w \in \{a, b, c\}^* \text{ and } \#_a(w) = \#_c(w) \text{ and } \#_c(w) > \#_b(w)\}$$

Exercise 5 (10 points):

Consider the following two languages:

Large_Factor = $\{\langle n, t\rangle : n$ and t are positive integers, $t < n$, and n has a factor f satisfying $t \leq f \leq n\}$

Prime = $\{n$: where n is positive integer that is also a prime, $n \geq 2\}$

What exactly is the flaw in the following argument:

> "A Turing Machine for Prime receives an integer n as input. It rejects the input if $n < 2$. If $n = 2$, the input is accepted. Otherwise, the Turing Machine invokes Large_Factor(n, 2) and accepts iff Large_Factor rejects. Thus Prime is polynomial-time reducible to Large_Factor. Since Large_Factor is known to be in NP, we can conclude that Prime is in NP."

Exercise 6 (20 points):

Classify the following languages as one of the following categories.

<u>Decidable</u>: if the language is decidable.
<u>Recognizable</u>: if the language is recognizable but not decidable.
<u>Co-Recognizable</u>: if the language is co-recognizable but not decidable.
<u>Neither</u>: if the language is neither recognizable nor co-recognizable.

a. $L = \{\langle r_1, r_2\rangle : r_1$ and r_2 are regular expressions such that $L(r_1) = L(r_2)\}$

b. $L = \{\langle M\rangle : M$ is a TM and M never uses more than 1000 tape squares$\}$

c. $L = \{\langle M\rangle : M$ is a TM and M loops on some string$\}$

d. $L = \{\langle M\rangle : M$ is a TM and $L(M)$ is infinite$\}$.

e. $L = \{\langle M\rangle : M$ is a TM and $L(M) \subseteq \Sigma^*\}$

Exercise 7 (12 points):

In the dominating set problem the input is an undirected graph G, the problem is to find the smallest dominating set in G. A dominating set is a collection S of vertices with the property that every vertex v in G is either in S, or there is an edge between a vertex in S and v. Show that the dominating set problem is NP-hard using a reduction from the vertex cover problem.

Exercise 8 (16 points):

The Bin Packing problem is one of the most studied optimization problems in Computer Science and Operation Research. It is defined as follows:

- Given items of size $a_1, \ldots, a_n$, and given unlimited supply of bins of size B, we want to pack the items into the bins so as to use the minimum possible number of bins.

We can think of bins/items as being CDs and MP3 files; breaks and commercials; bandwidth and packets, and so on.

The decision version of the problem is:

- Given items of size $a_1, \ldots, a_n$, given bin size B, and parameter k,
- Determine whether it is possible to pack all the items in k bins of size B.

Prove that this problem is NP_complete.