COMP 3721: Theory of Computation
Spring 2013 Final Exam

1. Print your name and student ID at the top of every page (in case the staple falls out!).
2. This is an open-book exam.
3. Time limit: 180 minutes.
4. You should answer all the questions on the exam. At least you should read all the questions.
5. When describing a Turing machine, you can use either pseudocode or plain language, just like how you would describe an algorithm. You may assume the most convenient variant of the TM, unless you are explicitly told otherwise.
6. You can write on the back of the paper if you run out of space. Please let us know if you need more scratch paper.

Marking Schemes

+

Hints

+

Notes

1. (20 pts) Assuming $\mathsf{P} \neq \mathsf{NP}$ and $\mathsf{NP} \neq \mathsf{coNP}$, put the following languages into the correct places in the diagram below. Let $\Sigma = \{0, 1\}$.

   (a) $K_1 = \{$"M" : Turing machine M halts on the empty string$\}$.

   (b) $\overline{K_1}$.

   (c) $\{$"M" : Turing machine M has at most 5 states and it halts on all $w \in \Sigma^*\}$.

   (d) $\{$"M""w" : Turing machine M halts on w in less than $|w|^3$ steps$\}$.

   (e) $\{G : G$ is a graph that has a Hamiltonian cycle$\}$.

   (f) $\{(G, k) : G$ is a graph whose minimum vertex cover has $\leq k$ vertices$\}$.

   (g) $\{(G, k) : G$ is a graph whose minimum vertex cover has $> k$ vertices$\}$.

   (h) $\{ww : w \in \Sigma^*\}$.

   (i) $\{ww^R : w \in \Sigma^*\}$.

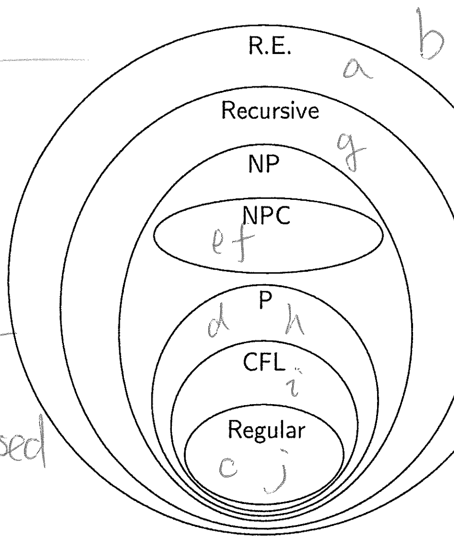
   (j) $\{n : n$ is a prime number less than $10^{10}\}$.

a correct letter : 2 points

a letter adjacent to the correct place : 1 points

others (including missing): 0 pt

if there are duplicates, the one with the ~~lowest~~ fewest points is used



Hint:

c is finite

j is finite, too

2. (10 pts) For each of the following sets, decide whether it is countable. You do not need to justify your answers.

(a) SAT $= \{\phi : \phi$ is a Boolean formula that is satisfiable$\}$.

2 pts for each correct answer

Y

(b) P.

Y

(c) 2^{P} (i.e., the set of all subsets of P).

N

(d) Recursive.

Y

(e) The set of all languages that are outside RE $\cup$ coRE

N

3. (15 pts) RE and coRE (defined as coRE $= \{L : \overline{L} \in \text{RE}\}$) are the largest classes of languages we have defined in class.

Here we construct a concrete language S outside RE and coRE, using a slight different form of diagonalization. Let $\{M_0, M_1, M_2, \dots\}$ be all the TMs, and let $\{x_0, x_1, x_2, \dots\}$ be all the strings. For each $n = 0, 1, 2, \dots$, we include x_{2n} in S iff M_n halts on x_{2n}; and we include x_{2n+1} iff M_n does not halt on x_{2n+1}. Show that $S \notin \text{RE} \cup \text{coRE}$.

Consider the following example, $S = \{x_0, x_2, x_3, x_7, \dots\}$.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
M_0	$\neq\nearrow$	$\neq\nearrow$						
M_1			$\neq\nearrow$	$\nearrow$				
M_2					$\nearrow$	$\neq\nearrow$		
M_3							$\nearrow$	$\nearrow$

an attempt to reduce $\rightarrow$ 5 points

a correct reduction $\rightarrow$ 10 points

(tranform + proof)

Hint:

$S \notin$ RE: suppose M_i semi-decides S.

should M_i halt on X_{2i+1} or not? Neither makes sense

$S \notin$ ~~RE~~ coRE: suppose M_i semi-decides $\overline{S}$

should M_i halt on X_{2i} or not?

4. (15 pts) A "perfect compiler" is one that takes any source code and produces the shortest machine code. Unfortunately, such a perfect compiler does not exist, as you will prove here. Formally, a perfect compiler is a TM that takes in (the encoding of) a TM M, and outputs (the encoding of) an equivalent TM M' with the minimum number of states, such that for any w, $M(w) = \nearrow$ iff $M'(w) = \nearrow$. Show that such a perfect compiler does not exist.

An attempt to reduce from a RELEVANT problem → 5 pts

a correct proof (transform + proof) → 10 pts

Note that: suppose M_1 is equivalent to M_2 and the perfect compiler produces M_1' and M_2' we only know that M_1' and M_2' have the same number of states, but their encodings can be ~~different~~

e.g. $> \sqcup R$ and $> R R$

Hint: to Solve $\{M : M \text{ halts on any } w \in \Sigma^*\}$

the TM with minimum ~~st~~number of states, which halts on all inputs is: $> h$

which is unique

5. (25 pts) In this question, we consider the class coNPC, the hardest problems in coNP. Recall that $\mathsf{coNP} = \{L : \overline{L} \in \mathsf{NP}\}$.

   (a) (10 pts) Show that the following two definitions of coNPC are equivalent. Note that you need to prove two directions.

      i. $\mathsf{coNPC} = \{L : \overline{L} \in \mathsf{NPC}\}$.

      ii. coNPC consists of all languages L such that $L \in \mathsf{coNP}$, and for any $L' \in \mathsf{coNP}$, $L' \leq_{\mathsf{P}} L$.

5pts for each direction

Hint:

$\overline{L} \in NPC$

$\Leftrightarrow \forall L' \in NP \Rightarrow L' \leq_p \overline{L}$

$\Leftrightarrow \forall L' \in coNP \Rightarrow L' \leq_p L$

$\forall L' \in coNP$

$\overline{L'} \in NP$ $\quad \overline{L'} \leq_p \overline{L}$ $\quad$ Y N

$\downarrow$ $\wedge_P$ V/P

$L' \leq_p L$ $\quad$ N Y

$w \in \overline{L}$

$\Updownarrow$

$w \notin L$

(b) (5 pts) Show that if $L \in \mathsf{coNP}$, and there exists an $L' \in \mathsf{coNPC}$ such that $L' \leq_{\mathsf{P}} L$, then $L \in \mathsf{coNPC}$. You may use the result of (a) even if you cannot prove it.

Hint: $\overline{L'} \in NPC$, $\overline{L'} \leq_P \overline{L} \wedge L \in NP \Rightarrow L \in NPC$

5 pts for a correct proof

(c) (10 pts) A *tautology* is a Boolean formula that is always true for any assignment of values to the variables. The TAUTOLOGY problem is, for a given Boolean formula ϕ, to decide whether ϕ is a tautology. Show that TAUTOLOGY $\in \mathsf{coNPC}$. You may use the results of (a) and (b) even if you cannot prove them.

Hint & NOTE:

SAT: $\{\phi : \exists a.\ \phi|_a = T\}$

$\overline{\text{SAT}}$: $\{\phi : \forall a.\ \phi|_a = F\}$ ✓

TAUTOLOGY: $\{\phi : \forall a.\ \phi|_a = T\}$ ✓

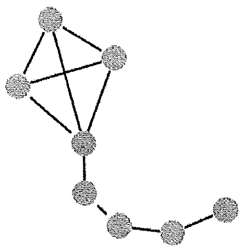
$\overline{\text{TAUTOLOGY}}$: $\{\phi : \exists a.\ \phi|_a = F\}$

TAUTOLOGY $\in$ coNP OR $\overline{\text{TAUTOLOGY}} \in NP$ $\longrightarrow$ 2 points

a CORRECT transformation/reduction $\longrightarrow$ 4 points

a CORRECT proof of BOTH directions $\longrightarrow$ 4 points

6. (15 pts) A *kite* is a graph with an even number of vertices, say $2k$, in which k of the vertices form a clique and the remaining k vertices are connected in a tail that consists of a path joined to one of the vertices of the clique. The figure below shows a kite of size 8. Given a graph G and k, the KITE problem asks whether G contains a subgraph which is a kite of size $2k$. Prove that KITE is NP-complete.



~~Hint~~

KITE $\in$ NP $\rightarrow$ 3 points

a CORRECT reduction $\rightarrow$ 4 points

proof of both directions $\rightarrow$ 4 points + 4 points.

Hint/Note:

reduce DCLIQUE to KITE:

add a tail of ~~length~~ length k to each vertex.

need to disscuss the case when $k<3$, since the CLIQUE you find may be the extra edges you added $\rightarrow$ 1 point

You cannot find a ~~KITE~~ kite by finding a clique then try to find a tail connecting to it.