CSCI 320 Midterm Test III SOLUTIONS NAME:________________________

April 2, 2019 Out of 55 marks 5 Questions 5 pages

Note that we use "TM" to mean "Turing Machine". The following languages are known to be **Undecidable**

$A_{TM} = \{< M, w > \mid M$ is a TM that accepts string $w\}$.
$Halt_{TM} = \{< M, w > \mid M$ is a TM that halts on string $w\}$.
$E_{TM} = \{< M > \mid M$ is a TM that accepts no strings$\}$.
$EQ_{TM} = \{< M_1, M_2 > \mid M_1$ and M_2 are TMs and $L(M_1) = L(M_2)\}$.
Regular$_{TM} = \{< M > \mid M$ is a TM and $L(M)$ is a regular language $\}$.
Halt$_{any}\{< M > \mid M$ is a TM and $\exists$ string w such that M halts on $w\}$.

1. (10 marks) Show the the following language is **not decidable**:
   $CF_{TM} = \{< M > \mid M$ is a TM and $L(M)$ is context-free $\}$. Do not use Rice's Theorem. For full marks, give a brief argument that shows your reduction works.

2. (a) (4 marks) Prove that the class of Turing-decidable languages is closed under intersection. [Hint: "Let L_1 and L_2 be two Turing-decidable languages, decided by X_1 and X_2 respectively. Then we can construct a TM X that decides the language $L_1 \cap L_2$ as follows: ..."]

   Solution:

   (4 marks) Prove that the class of Turing-decidable languages is closed under complement. [Hint: "Let L be a Turing-decidable language, decided by X. Then we can construct a TM $\overline{X}$ that decides the language $\overline{L}$ as follows: ..."]

   Solution:

   (b) (6 marks) Prove that the class of Turing-recognizable languages is closed under union.

   Solution:

3. Are the following languages recognizable? Prove your answer.

(a) (10 marks) Reject$_{any}$ $= \{< M > | M$ is a TM, and there is a string w that M rejects $\}$.
Solution:

Claim: Reject$_{any}$ is recognizable.
Proof: The following TM R recognizes Reject$_{any}$

$R =$ "on input $< M >$:
1. Set i = 1.
2. For each string w of length i or less,
run M on w for i steps; if M rejects any string, halt and ACCEPT.
3. Set $i = i + 1$. Go to 2."

Note that for any given value of i, a finite amount of work is being done. This solutions works, because if M rejects some string, R will eventually get to the value of i that finds this out. If there is no such string, R will loop forever – but R is just a recognizer, so that's okay.
(Note to students: It is necessary that M run only for a limited (though ever increasing) number of steps on each string; we need to ensure that M is not allowed to loop forever on some string, when another string would make M reject and therefore prove that M should be accepted by R.)

(b) (10 marks) OneAccepts= $\{< M_1, M_2, w > | M_1$ and M_2 are TMs, and exactly one of M_1 and M_2 accepts $w.\}$.

Solution:
Claim: OneAccepts is not recognizable.
Proof: Suppose $\exists$ a TM Y that recognizes OneAccepts. Then the following TM S **decides** A_{TM}

$S =$ "on input $< M, w >$:
1. Create M_{all}, where M_{all} operates as follows:

M_{all} = "on input $< x >$, ACCEPT."
2. In parallel, run M on $< w >$ and Y on $< M, M_{all}, w >$.
3. Depending on which one returns a value first, do the following:

3.1 if M accepts, ACCEPT. If M rejects, REJECT.

3.2 if Y accepts, REJECT. If Y rejects, ACCEPT."

Note that if M halts, then so will S and it will return the correct answer. If M loops forever, Y will accept, and so S will reject. Hence S is a decider.

Since no such decider for A_{TM} can exist, Y cannot exist, and OneAccepts is not recognizable.

4. Two versions (5 marks)
   1. Prove that the following language is in NP.
   SET_PARTITION $= \{< S > |S$ is a multiset of non-negative integers, and S can be partitioned into two sets with equal sum.$\}$.
   "Partition" means divide the set into two parts.
   "Multiset" means there can be duplicate values.
   An example of a multiset that is in SET_PARTITION is $\{1, 2, 5, 5, 7\}$.

   Solution:

   1. Non-deterministically guess sets X and Y.
   2. Test whether the multiset union of X and Y is S; if not, REJECT.
   3. Sum the elements of X, and the elements of Y; if equal, ACCEPT; otherwise REJECT.

   Step 1 can be done in O(n) time non-detemrinisticallly (making lucky guesses).
   Step 2 can be done in O(n) time (cross of the elements of S as you scan X and Y).
   Step 3 can be done in O(n^2) time (summing n integers).

   2. Prove that the following language is in NP.
   LongCommonSubsequence $= \{< R, K > |R$ is a collection strings from Σ^*, and K is a positive integer, and there is a string of length K that is a substring of every string in R.$\}$.

   An example of a collection that is in LongCommonSubseqence is $(\{abba, babb, ababb\}, 3)$.

   Solution:

   1. Non-deterministically guess a string v.
   2. Count the characters in v; if $< K$, REJECT.
   3. Scan each string in R in turn to find if v is a substring of R. If there is a string in R that does not contain v, REJECT.
   4. If all strings have been scanned and v found in all, ACCEPT.

   Step 1 can be done in O(n) time non-deterministically (making lucky guesses).
   Step 2 can be done in O(n) time.
   Step 3 can be done in O(n^2) time (check each K-length substring of each string).

5. (a) (6 marks) Give a 3SAT clause that is satisfiable if and only if the following SAT formula is satisfiable, using the construction given in class: $(x_1 \vee \overline{x_2} \vee x_3 \vee x_4 \vee \overline{x_5} \vee x_6)$.

   Solution: $(x_1 \vee \overline{x_2} \vee A) \wedge (\overline{A} \vee x_3 \vee B) \wedge (\overline{B} \vee x_4 \vee C) \wedge (\overline{C} \vee \overline{x_5} \vee x_6)$.

   (b) (4 marks) Give some assignment of truth values to the variables $x_1, \ldots, x_6$ that non-redundantly satisfies the SAT formula above (i.e., that makes only one of the literals in the "OR" clause true). Then show how that assignment can be extended (by giving values to the newly introduced variables) so that the corresponding 3SAT formula is also satisfied.

   Solution: Pick any literal in the boolean formula above to be true, say $\overline{x_2}$, and all the other **literals** will be false. That means $x_1 = F, x_2 = F, x_3 = F, x_4 = F, x_5 = T, x_6 = F$. This forces the following assignment: $A = F, B = F, C = F$.
   We could write this as "00010 $\rightarrow$ 000". The other implications are below – any one of them would do for an answer to this question.

$110010 \to 000$
$000010 \to 000$
$011010 \to 100$
$010110 \to 110$
$010000 \to 111$
$010011 \to 111$