COSC 417 Test 2 Name:Henry Osei

Instructions:

Time 120 minutes. All the questions have equal weight. If you use a fact that was discussed in class or in our meetings you can state it and no further proof is needed. The test is open books, open notes, open internet. You are not allowed to ask any person or internet forum.

Your signature above implies the following Honor Statement "***I pledge on my honor as a student, that I have not received or provided any unauthorized help on this exam.”***

Please do not compromise academic integrity. If cheating, you will get 0 for the final exam and I will report you for academic dishonesty.

Do not write statements that are not relevant to the question. If you leave a question blank you get 20 (out of 100 for the question), but if you write an unrelated text, you get 0/100.

You will submit this .doc file on Blackboard. You can type your answers, or hand-write them and insert digital pictures of them. If you type, you can use “latex”-type shortcuts. For example, you can use \alpha, \beta, \phi, for greek letters, *5^{194}* for 5 raised to the power 194, *x_1* for subscript, etc. Alternatively, you can submit one pdf file with all your answers.

During the exam, I will be on webex and you can ask questions using the chat function.

Question 1. Let L = {$a_n\, b_m\, a_n\, b_m$ | n, m integers, $n \geq 0, m \geq 0$}. Choose the correct answer (no justification is required).

(a) Is L regular? YES or NO ANSWER:

NO

(b) Is L context-free? YES or NO ANSWER:

YES

(c) Is L decidable? YES or NO ANSWER:

YES

Question 2. Use the Pumping Lemma for context-free languages to show that the language $L = \{a^{n^2} \mid n \geq 0\}$ is not context-free. Use the given steps of the proofs and fill-in the argument for Step III by analyzing the possibilities resulted from the splitting in STEP II.

STEP I. Take $s = a^{p^2}$, where p is an arbitrary natural number.

STEP II. The adversary splits $s = uvxyz$, with $|vy| > 0$ and $|vxy| \leq p$.

STEP III. Take i = 2. Then $s' = u\, v_2\, x\, y_2\, z$ is not in L, because <here you need to explain why the length of s' cannot be a perfect square, no matter what the adversary did at step II. Hint: As a function of p, what is the length of s? What is the minimum length s' can have? What is the maximum length s' can have? From this, conclude that the length of s' cannot be a perfect square.>

ANSWER:

- **Since can only contains only one symbol which is a. So when I =2, increases the number of occrences only one symbol changes the number of occurrences. This will result in s never being able to become a perfect square. The length of s excedess the length of |vxy|<=p, s' would have to have a length which was less than uv^2xy^2z. Therefore we are able to conflude, that the length of s' cannot be a perfect square**

Question 3. Show that if a language A is context-free, then its complement is decidable.

ANSWER:

If a Language L is context free then it would be decidable. Therefore since the class of decidable languages is under complementation L is decidable

Question 4. Explain what is incorrect in the following ``proof" that claims to show that if a language A is Turing-recognizable, then its complement is also Turing-recognizable (a fact we know is not true).

``Proof". Let M be TM that Turing-recognizes A. We build M' a TM that Turing-recognizes complement of A.

(1) For every string x, $x \in A$ iff M on input x enters q_{accept}
(2) We take M' to be the TM which is identical to M except that we flip the states q_{accept} and q_{reject}.
(3) Then an arbitrary string x ∈ complement of A ⇔ M does not accept x
(4) M does not accept x ⇔ M' accepts x, because in M' we flipped q_{accept} and q_{reject}
(5) The last 2 statements imply that M' Turing-recognizes the complement of A

Indicate which of the statement or statements (1) – (4) is incorrect, and explain why.

ANSWER:

There is a roblem in steps 3-5 although for step 2 they do flip the accept and reject state then never flip it back to a state where a string could be accept therefore it could ot be turning recognizable.

Question 5. Consider the following problem P:

Input: M, a Turing machine.
Question: Is L(M) equal to Σ^* ?

Show that the above problem is undecidable by proving that $A_{TM} \leq_m$ P . Use the following proof skeleton.

On input <M,w> , we construct a TM M_1 such that the following statements (1) and (2) hold.
(1) if M accepts w, then $L(M_1) = \Sigma^*$ (YES maps to YES) and (2) if M does not accept w, then $L(M_1)$ is the empty set (so, as desired, NO maps to NO).

Description of M_1 on an arbitrary input x:

Simulate M on w, and if M is accepted then enter q_{accept}. If M enters its rejection state then enter q_{reject}

Simulate M on w.

If M enters on w and $L(M_1) = \Sigma^*$ this means that w would be in the set of both $L(M_1)$ and Σ^* because we are texting to see if the Σ^* & $L(M1)$ contains the string w by checking if M is equivalent to Σ^*

If M on w enters q_{accept} **then M₁ contains a string which is in the Language meaning that Σ^* *also contains the* string .**

If M on w enters q_{reject} then **TM will Halt because M₁ does not contain any strings that are in Σ^* meaning that M is not in** Σ^*

Else M₁ **does contain strings but w is not a string of Σ^***

Question 6. We define the following operation DELETE-FIRST for languages. Given a language L over the binary alphabet $\Sigma = \{0,1\}$, DELETE-FIRST(L) is the language consisting of all strings in L in which we have removed the first letter (the empty string remains unchanged). For example, if $L = \{10^n | n \geq 0\}$, then DELETE-FIRST(L) = $\{0^n | n \geq 0\}$. Show that if L is Turing recognizable, then DELETE-FIRST(L) is also Turing-recognizable.

Hint: Use a TM M that Turing-recognizes L, to obtain a TM M' that Turing-recognizes DELETE-FIRST(L). Note that x is in DELETE-FIRST(L) if and only if at least one of 0x or 1x is in L. M' has to simulate M on certain inputs and use the dovetailing technique.

ANSWER:

M' on input x

If both L and complement L are RE, we let x be the Turning Recognizer L and M' be the TM recognizer for complement L

- **M and M' are both dovetailing for the input x**
- **Simulate x on input iff at least one of 0x or 1x is in L**
- **If x enters q_{accept}, accept**
- If x rejects, M rejects. If M' accepts then M rejects

Therefore M' Turning-recognizes L because L accepts M' iff there is at least one 0x or 1x

Question 7. For each the following problems, indicate whether they are decidable or undecidable. No justification is required.

(1) Input: (M, w), where M is a Turing machine and w is an input string.
Question: Does M accept w?

Choose one of DECIDABLE, UNDECIDABLE.

ANSWER:

UNDECIDABLE

(2) Input: P, where P is a Java program.
Question: Does P contain an *if* statement?
Choose one of DECIDABLE UNDECIDABLE.

ANSWER: **DECIDABLE**

(3) Input: (P, w), P is a Java program, w is an input for the program P.
Question: Does P go into an infinite loop on input w?
Choose one of DECIDABLE, UNDECIDABLE.

ANSWER:

UNDECIDABLE

Question 8. For each of the following problems, indicate whether they are NP-complete, or not NP-complete (under the assumption P $\neq$ NP).

(1) Given a boolean formula ϕ determine if ϕ is satisfiable.
Choose one of NP-complete , not NP-complete.
ANSWER:

- **Not NP Complete**

(2) Given a binary string x, determine if x is of the form $0^n 1^n$
Choose one of NP-complete , not NP-complete.
ANSWER:

- **Not NP Complete**

(3) Given a graph G and a positive integer k, determine whether there are k vertices in G with no edges between them.
Choose one of NP-complete , not NP-complete.
ANSWER:

- **NP Complete**

(4) Given a graph G, and two vertices s and t, determine if there is a path in the graph from s to t.

Choose one of NP-complete , not NP-complete.
ANSWER:

- **Not NP-Complete**

Question 9. A Java program P prints on the screen in order a sequence of binary strings $x_1, x_2, \ldots, x_n, \ldots$ (the program may never halt and so the sequence can be infinite). The strings are printed in increasing order of length, i.e., for all i, $|x_i| < |x_{i+1}|$.

Consider the following decision problem:

Input: binary string u.
Question: Is u printed by P?

Show that this decision problem is decidable.
ANSWER: <present here the algorithm that determines if an arbitrary string u is printed or not by P. You have to let P run but at some moment you need to stop it and take the right decision.>

ADFA{<P,u>| P is a DFA that accepts u}

The TM is given as input <P,u>. The DFA and a string u. The TM checks to make sure Pis a valid representation. The TM then simulates P on u If P reaches a Final state at the end of u, then the TM will accept or reject therefore proving the program will halt meaning it is decidable.

Question 10. MORE-SAT is the following decision problem.

Input: Boolean formula ϕ.
Question: Does ϕ have at least two satisfying assignments?

The goal of the problem is to show that MORE-SAT is NP complete.

(a) Consider the following two Boolean formulas:

$$\phi_1 = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

$$\phi_2 = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (y \vee \neg y)$$

How many satisfying assignments does ϕ_1 have ?

How many satisfying assignments does ϕ_2 have? (Note that ϕ_2 is obtained from ϕ_1 by adding the last clause).
ANSWER:

$\phi 1$, is a nondeterministic polynomial time machine can guess two assignments and accept if both assigments satisfy ϕ

$\phi 2$ has three assignemts due to the reasoning of $(y \vee \neg y)$

(b) Show that MORE-SAT is in NP. You need to indicate what is a certificate of a YES-instance, and describe a Verifier.
ANSWER:

$\phi_1 = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ =X_1=**T, X_2=T X_3= F**

$\phi_2 = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (y \vee \neg y)$= X_1=**T, X_2=F X_3= T, y=F**

MORE-Sat is satisfied by the assignments

(c) Now we show that 3-SAT $\leq_p$ MORE-SAT. You need to describe a transformation $\phi \mapsto \phi'$ that (1) is poly-time computable , and (2) if ϕ is a YES-instance of 3-SAT then $\phi'$is a $\mathrm{YES-instance\ of\ MORE-SAT}$, and (3) if $\phi'$is a $\mathrm{YES-instance\ of\ MORE-SAT}$ then ϕ is a YES-instance of 3-SAT. (Hint: get inspiration from part (a). What is the effect of adding the clause $(y \vee \neg y)$?) Show the three assertions (1), (2), and (3) using the given template.
ANSWER:

Adds one more instance which turns y from False to True

The transformation $\phi \mapsto \phi'$ is <describe the transformation and argue that (1) is true>

The transformation includes the complement of ϕ

Statement (2) is true because:

ϕ' being a complement of ϕ

Statement (3) is true because: