CS 4510 Automata and Complexity

Exam 3 Solutions: Turing Machines

Date: 2:40-5:30pm Tuesday, December 12 *Prof. Jaeger*

Name: ______________________ GTID: ______________________

- **When handing in this test, please go to a TA with your BuzzCard to verify the name on the test.** By submitting this exam, you agree that your actions during this exam conform with the Georgia Tech Honor Code.
- Do not open this exam until you are directed to do so. Read all the instructions first.
- **Write your name and GTID# on the top of every page.** Your GTID# can be found on your BuzzCard.
- This exam is out of 101 points (with one extra credit point for writing your name). Each problem is labeled with the number of points it is expected to be worth. A curve may be applied. The last question is the CIOS bonus question which (if the threshold is reached) will be applied directly to your course grade.
- Write your solutions in the space provided. If you run out of space, continue your answer on the back of the last page, and make a notation on the front of the sheet.
- There is a page of scratch paper at the end of the exam. We can provide additional scratch paper if you need it. The only outside material you may use on the exam is one (1) double-sided, hand-written, 8.5"x11" page of notes. Calculators are NOT permitted.
- You may use any of the theorems/facts/lemmas from the lecture notes, homeworks, or textbook without re-proving them unless explicitly stated otherwise. If you have a question, you may ask a TA.
- Do not spend too much time on any one problem! If you get stuck, move on and come back to that one later. Good luck!

For your reference, here are some examples of complete problems we saw.

Complexity Class	Complete Problem
NP	SAT, 3SAT, CLIQUE, VERTEX-COVER, DOMINOS,...
PSPACE	TQBF
NL	PATH

Name: ______________________________ GTID: ______________________________

0. **Following Instructions [1 point extra credit]**

   Please write your name and GTID# at the top of EACH page of this exam. Your GTID# can be found on your BuzzCard.

1. **Multiple Choice [16 points - 2 points each]**

   Circle the correct answer. No justification necessary.

   (a) For every Context Free Language L, there $\boxed{MUST}$ / MAY / DOESN'T exist a Turing Machine that decides L.
   CFL is a subset of $TDEC$.

   (b) If $SAT \in P$, then $P = NP$ $\boxed{MUST}$ / MAY / DOESN'T hold.
   SAT is NP-hard.

   (c) Suppose $A \leq_L PATH$, then A is NL-complete MUST / $\boxed{MAY}$ / DOESN'T hold.
   This tells us $A \in NL$, but we don't know if it is NL-hard

   (d) If TM M is a recognizer for a language L and $w \in L$, then M $\boxed{MUST}$ / MAY / DOESN'T halt when run on input w.
   M must accept (and so halt) on input w.

   (e) Suppose $A \leq_P B$ and B is in P, then A is in NP $\boxed{MUST}$ / MAY / DOESN'T hold.
   $A \leq_P B$ and $B \in P$ tells us $A \in P$. Note $P \subseteq NP$.

   (f) Suppose L is PSPACE-hard, then it MUST / $\boxed{MAY}$ / DOESN'T hold that L is undecidable.
   E.g., $TQBF$ and $HALT$ are both PSPACE-hard.

   (g) Suppose M is a $NPSPACE$-decider for L. Then it $\boxed{MUST}$ / MAY / DOESN'T hold that an accepting tableau for M is polynomially wide. (That is, the length of each row is bounded by a polynomial in the length of x.)
   An $NSPACE$ machine touches at most polynomially many cells.

   (h) The set of all languages is COUNTABLE / $\boxed{UNCOUNTABLE}$.
   We showed this for our first proof that there exist undecidable problems.

2. **Language Complexity [8 points - 2 points each]**

   Let L_1, L_2 be decidable, and L_3 be recognizable. For each language below, write D in the blank if it is Turing decidable, R in the blank if it is Turing recognizable but not decidable, and X in the blank if it is neither Turing decidable or recognizable.

   (a) $L_1 \circ L_2$ ______ $\boxed{D}$

   (b) $L_3 \setminus L_1$ ______ There are multiple possibilities, everyone gets full credit for this.

   (c) $L_1 - L_2$ ______ $\boxed{D}$

   (d) $L_1 - L_3$ ______ There are multiple possibilities, everyone gets full credit for this.

   Recall $A \circ B = \{ab \mid a \in A, b \in B\}$ and $A \setminus B = \{x \mid x \in A, x \notin B\}$.

Name: ______________________________ GTID: ______________________________

3. **Configurations and Tableaus**

   (a) **[8 points]**Suppose M is a TM where the following is the ONLY transition into q_0.

$$\begin{tikzcd} \boxed{q_0} \arrow[r, "0 \rightarrow 1{,} R"] & \boxed{q_1} \end{tikzcd}$$

For each of the following 2×3 windows (domino), circle them if they will NOT occur in any valid tableau for M.

a)

0	0	1
q_1	0	1

$b)$

1	0	0
0	q_1	0

$c)$

q_0	0	1
0	q_1	1

$d)$

1	q_0	1
q_1	1	0

$e)$

1	1	1
q_1	1	1

$f)$

1	0	0
0	1	q_1

$g)$

0	q_0	1
0	0	q_1

$h)$

0	1	0
1	0	q_0

   (b) **[5 points]** Suppose M is a TM with $|M.\Sigma| = 2$, $|M.\Gamma| = 3$, $|M.Q| = 5$. On input $x = 1111111 = 1^7$, it only touches the first 11 tape locations. Assuming M halts on input x, prove an upper bound on t, the number of times steps it takes. (Hint: relate this to the number of configurations.) Leave your answer in the form $t = 2^a \cdot 3^b \cdot 5^c \cdot 7^d \cdot 11^e$. Note that some of a, b, c, d, e will be 0.

Because M halts, it cannot repeat the same configuration twice when computing on input x. So we can bound the number of time steps by the number of possible configurations. Recall that a configuration specifies (a) the tape heads locations, (b) the current state, and (c) the current content of the tape. We bound (a) by 11 the number of cells touched by M. We bound (b) by $|M.Q|$. We bound (c) by noting that each of the 11 cells touched can have at most $|M.\Gamma|$ possible values, giving 3^11. Put together, our bound in $3^{11} \cdot 5^1 \cdot 11^1$.

The question asked for a proof of your bound. Students who write down (without justification) seemingly random upper bounds that happen to be correct do not receive credit.

4. **Open problems [8 points]**

The most famous open question in computer science is whether or not P=NP. In the latter part of this course we described a number of other open questions. Give an example of an open question not equivalent to P vs. NP and describe how proving something about a complete problem would answer the question.

The expected answer was something like "It is an open question whether of not $L = NL$. The problem $PATH$ is NL-complete, so if we were able to show $PATH$ is in L then we would know $L = NL$ must hold. If $PATH$ is not in L, then of course $L \neq NL$." Other possible examples include P vs. PSPACE, NP vs. PSPACE, and NP vs. coNP. PSPACE vs. NPSPACE is not an acceptable answer as we know they are equal. Many students were confused by what this question was asking, so we were generous with points.

Name: ______________________ GTID: ______________________

5. **Recognizability**

(a) **[8 points]** Give a deterministic Turing machine which recognizes

$$L_1 = \{\langle M \rangle \mid \text{TM } M \text{ accepts when run on input } \langle M, M \rangle\}.$$

We can define the following recognizer.

```
$R$ on input $\langle M \rangle$:
Run $M$ on $\langle M, M \rangle$.
If $M$ accepts $\langle M, M \rangle$, accept $\langle M \rangle$
```

(b) **[8 points]** Let f be a mapping reduction. Give a deterministic Turing machine which recognizes

$$L_2 = \{y \mid \text{there exists an } x \text{ such that } f(x) = y\}.$$

```
$R$ on input $y$:
For each $x \in \Sigma^*$ //in some fixed ordering:
  Compute $y' = f(x)$ //$f$ is computable because it's a mapping reduction
  If $y = y'$, accept $y$
```

Name: ______________________ GTID: ______________________

6. **Undecidability [8 points]**

Prove that $L = \{\langle M\rangle : |L(M)| = 2 \text{ or } |L(M)| = 3\}$ is undecidable.

We will prove that L is undecidable by showing $A_{TM} \leq_m L$. We define our reduction f on input $\langle N, w\rangle$ to output $\langle M\rangle$ where is M is a machine with input x that runs N on w and accepts only if N accepted w and if x is "CS" or "4510".

$\underline{\text{If } \langle N, w\rangle \in A_{TM}, \text{ then } \langle M\rangle \in L.}$ When $\langle N, w\rangle \in A_{TM}$, N will accept inside of M, making $L(M) =$ {"CS", "4510"}. Therefore, $L(M) = 2$, so $\langle M\rangle \in L$.

$\underline{\text{If } \langle N, w\rangle \notin A_{TM}, \text{ then } \langle M\rangle \notin L.}$ When $\langle N, w\rangle \notin A_{TM}$, N will reject or run forever inside of M, making M definitely not accept. Therefore, $L(M) = \emptyset$, so $\langle M\rangle \notin L$.

Since A_{TM} is undecidable and $A_{TM} \leq_m L$, we have that L is undecidable.

Name: ______________________________ GTID: ______________________________

7. **Complexity Classes [8 points]**

   Recall that $coNP$ is the complexity class defined by $coNP = \{\overline{L} \mid L \in NP\}$. It is not known whether $3SAT$ is in $coNP$. Suppose that $3SAT$ is not in $coNP$. Show that this fact implies P is not equal to NP. (Hint: Assume for contradiction that $P = NP$. Then consider closure properties you know.)

   Assume, for contradiction, that $P = NP$ and $3SAT$ is not in $coNP$. We know $3SAT$ is in NP and thus, under our assumption, $3SAT$ is in P. We also know that P is closed under compliment. Hence $\overline{3SAT}$ is in P and thus in NP. But then $\overline{3SAT}$ being in NP means that $\overline{\overline{3SAT}} = 3SAT$ is in $coNP$, which we assumed not to be the case.

Name: ______________________________ GTID: ______________________________

8. **Half Closure**

Consider the $HALF()$ operation, defined as follows: Let L be a language. Then $HALF(L) = \{x \mid xy \in L \text{ and } |x| = |y| \text{ for some } y\}$, i.e. the set of strings which are the first half of some string in L. For example, if $L = \{hamburgers, math\}$, then $HALF(L) = \{hambu, ma\}$.

(a) **[8 points]** Show that NP is closed under $HALF$, i.e. show that, for any arbitrary language $L \in NP$, $HALF(L) \in NP$.

To show that for any arbitrary language $L \in NP$, $HALF(L) \in NP$, we can make a nondeterministic poly-time decider for $HALF(L)$. Given an input x, we can nondeterministically guess y of length $|x|$. Then we can make the string xy and use the poly-time NTM decider for L (which exists because $L \in NP$) to check if $xy \in L$. We return the same answer that decider does. This is a decider because all of our steps are guaranteed to halt. We first guess y, then run the decider for L, and then accept or reject based on its output. Thus, the NTM is a decider. It is poly-time because every step runs in poly-time. Since we have made a poly-time NTM decider for $HALF(L)$, $HALF(L) \in NP$.

(b) **[8 points]** Show that $PSPACE$ is closed under $HALF$, i.e. show that, for any arbitrary language $L \in PSPACE$, $HALF(L) \in PSPACE$.

This is similar to the above. Let $L \in PSPACE$ then let D be the poly-space decider for it. We building a new poly-space decider M which on input x runs $D(xy)$ for all possible y of length $|x|$. M accepts if D accepts any of these. Note that M is still poly-space because each invocation of D can reuse the same tape cells.

Name: ______________________ GTID: ______________________

9. **Oracle Machines [8 points]**

If L is a language, then an oracle TM M^L is a normal TM except it is also given access to a magic device for L. The TM can repeatedly ask this device questions of the form "Is x in L?" and the device will give it the correct answer. For example, the language of the following machine is $\overline{DOMINOS}$.

```
$\underline{M^{DOMINOS}(x)}$
Ask "Is $x$ in $DOMINOS$?"
If yes: reject $x$
Else: accept $x$
```

Prove that there exists an oracle machine M^{TQBF} whose language is $\overline{CLIQUE}$.

One student noted we forgot to place a time bound on the machine M^{TQBF}, so one could just ignore the oracle and brute-force compute the solution. Ignoring that, the intended solution was to note that $CLIQUE \in NP \subseteq PSPACE$. Then $\overline{CLIQUE} \in PSPACE$ because $PSPACE$ is closed under compliment. Now, because $TQBF$ is $PSPACE$-hard $\overline{CLIQUE} \leq_p TQBF$ must hold. Let f be the corresponding poly-time mapping reduction. Then out machine M^{TQBF} on input x can compute $y = f(x)$, as its oracle "Is y in $TQBF$", and then return whatever answer the oracle says. Many students were confused by what this question was asking, so we were generous with points.

10. **ChatGPT BONUS**
[1 point to COURSE GRADE if CIOS $\geq 85\%$**]**

We gave this prompt to ChatGPT:

Is the language $\{\langle M\rangle, \langle N\rangle \mid M \text{ and } N \text{ are both TMs}\}$ Turing decidable or only Turing recognizable?

Here is the response it gave:

The language $\{\langle M\rangle, \langle N\rangle \mid M \text{ and } N \text{ are both TMs}\}$ is Turing recognizable but not Turing decidable.

To understand why, let's break down the language. The set $\{\langle M\rangle, \langle N\rangle \mid M \text{ and } N \text{ are both TMs}\}$ represents the set of pairs of Turing machine descriptions. In other words, it includes pairs of Turing machines, where each pair is denoted by the descriptions (encodings) of the Turing machines M and N.

Turing recognizability means that there exists a Turing machine that can accept (halt and output "yes") on all valid inputs (pairs of Turing machine descriptions where M and N are both TMs). In this case, you can design a Turing machine that simulates both M and N on all possible inputs and accepts if at least one of them accepts.

However, it's not Turing decidable because there is no general algorithm to decide whether, for any given pair of Turing machines, both machines will behave in the same way on all inputs. The famous Halting Problem is an example of an undecidable problem, and it can be reduced to this problem. If you could decide whether any two Turing machines are equivalent, you could solve the Halting Problem, which is known to be undecidable.

Is ChatGPT correct? Justify your answer.

ChatGPT is wrong because the language is decidable. Other explanations of how ChatGPT's response is nonsense/unintelligible may also be accepted. At time of writing, CIOS is at 80% so this question has not yet been graded.

Name: ______________________ GTID: ______________________

This page provides extra space if you need it. Clearly mark questions that use this space.