Winter 2024 CSE322 – Theory of Computation: Final Exam *70 points*

Instr.: Syamantak Das **8th May 2024, 10:00am-12:00pm**

Name: **Roll :**

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total

Mark **every** correct answer by writing $\boxed{\text{Yes}}$, and **every** incorrect answer by writing $\boxed{\text{No}}$ in the box to the left of each question. Each correct choice is worth 1 point; each incorrect choice is worth -1/2 point.

Q1 [6x2 points] Which of the following are surely decidable?**[Sarthak+Yashwant]**

- [ ] $\{xy \ : \ yx \text{ is a palindrome}\}$ (T)
- [ ] $\{\langle M\rangle \ : \ \text{TM } M$ accepts the language consisting of the binary representations of prime numbers$\}$ (F)
- [ ] The language consisting of all regular expressions that represent the language $\{0,1\}^*$ (T)
- [ ] L which is mapping reducible from $3SAT$ (F)
- [ ] L which is mapping reducible from A_{TM} (F)
- [ ] L^* if L is decidable (T)

Q2 [4x2 = 8 points] Which of the following are recognizable?**[Sarthak+Yashwant]**

- [ ] Encoding of Turing machines that accept their own encodings (T)
- [ ] $\overline{E_{TM}}$ is mapping reducible to L **Ambiguous question** (everyone gets score)
- [ ] L which is mapping reducible to A_{TM} (T)
- [ ] complement of EQ_{TM} (F)

Q3 [10 points] Consider the language $PRIME_{TM} = \{\langle M \rangle \mid |L(M)| \text{ is a prime number }\}$. You are going to prove that this language is not recognizable by showing a mapping reduction from $\overline{A_{TM}}$ through the following steps.**[Sagnik]**

(a) (2 points) Describe the inputs and outputs of the mapping reduction. **Solution.** Input : $\langle M, w \rangle$ where M is a TM and w is an arbitrary string (+1)
Output : $\langle M' \rangle$, where M' is a TM (+1)

(b) (6 points) Describe a TM computing the mapping reduction from $\overline{A_{TM}}$ to $PRIME_{TM}$ and explain precisely why this TM correctly computes the mapping.(Hint: As usual, try to 'construct' a new TM whose language will depend on the instance for $\overline{A_{TM}}$. Try some language other that $\Sigma^\star$ or $\emptyset$. You may use the fact that $\Sigma^\star$ is a countable set)

Solution. The TM which computes the reduction works as follows on input $\langle M, w \rangle$: Construct a new TM M' : (+1)
"On input x
If x is the any of first three strings in a lexicographic ordering of $\Sigma^\star$ (for example 0, 1, 01) (+1)
then accept
Else
If x is the fourth string in the above ordering
run M on w (+1)
accept x if M accepts w, reject x if M rejects w
Else reject x "
Output $\langle M' \rangle$

Explanation:

- $\langle M, w \rangle \in \overline{A_{TM}} \implies L(M')$ accepts three strings which is a prime number (+1)
- $\langle M, w \rangle \in \overline{A_{TM}} \implies L(M')$ accepts four three strings which is not a prime number (+1)

(In case you have done a reduction from A_{TM} to $\overline{PRIME_{TM}}$, a similar rubric works)

(c) (2 points)Explain using (b) why this implies that $PRIME_{TM}$ is not recognizable.

Solution. Assume for contradiction that $PRIME_{TM}$ is recognizable. But then by the above mapping, $\overline{A_{TM}}$ is also recognizable, which is a contradiction (any similar answer works). (+2)

Q4 [10 points] A language B is excited if every string in B takes the form ww for some $w \in 0, 1$. For example, $\{00, 1111, 1010\}$ is excited, but $\{00, 10\}$ is not excited. Prove via generic reduction from A_{TM} (not necessarily a mapping reduction) that the language $EX_{TM} = \{\langle M \rangle \mid L(M) \text{ is excited}\}$ is undecidable.

Solution.[Navidha+Shreyas] We will do a Turing reduction in this case (not a mapping reduction. If you have done a mapping reduction correctly (which is a little tricky here), you get full score. Assume EX_{TM} is decidable and D is a decider for the same. Now we will build a 'decider' for A_{TM} which will lead to a contradiction. (+1 for just writing this)
'Decider' for A_{TM}: $\langle M, w \rangle$ (+1 for correct input)
Construct another TM M : (+1 for just saying this)
" On input x :
If $x = 00$ (we have also given score for $x = \varepsilon$ or even writing 'if x is of the form ww') (+2)
then accept
Else
run M on w (+1)
accept x if M accepts (+1), reject x if M rejects (+1) "

Run Decider D on $\langle M' \rangle$ (+1)
accept if D *rejects* (+1), reject if D *accepts* (+1)

Please note here that this is *not* a mapping reduction since if $\langle M, w \rangle$ is a member of A_{TM} then $L(M') = \Sigma^\star$ which is *not* in EX_{TM} and vice-versa.

Q5 [5+3+2 points] Prove that $3SAT$ is mapping reducible to $HALT_{TM}$. Prove that the above reduction actually works in polynomial time. Does this prove that $HALT_{TM}$ is NP-complete ? Justify.

Solution.[Farhan] Given a 3-SAT instance ϕ, construct an encoding of the following TM M:
"On input $\langle\phi\rangle$:

1. Loop over all possible assignments of ϕ.
2. If a satisfying assignment is found, then HALT and ACCEPT.
3. Otherwise loop forever."

This is a valid mapping reduction since $\phi \in$ 3-SAT $\iff \langle M, \langle\phi\rangle\rangle \in$ HALT.
The following breakup is given for the reduction:

- Only 1 mark for trying to map a boolean formula ϕ to $\langle M, w\rangle$ instance for $HALT_{TM}$
- 1-3 marks for invalid reductions from $3-SAT$ to $HALT_{TM}$ (marks given based on closeness to actual answer and conceptual mistakes made)
- 5 marks for correct reductions (with only minor syntax errors at max)

The above encoding has constant size (since it is just the encoding of a TM and not the actual runtime of the TM on any input), so constructing the encoding takes polynomial time. **(Binary marking +3)**
Since $HALT_{TM}$ is undecidable, it cannot be in NP and hence, it can trivially can never be NP-complete (irrespective of the reduction). **(Binary marking +2)**

Q6 [10 points] Consider the language $L = \{\langle D\rangle \mid D$ is a DFA such that $L = \{\langle D\rangle \mid D$ is a DFA such that D accepts w^R wheneve

Either prove or disprove that L is decidable. For proving decidability, give an algorithm that decides this language. For proving undecidability, reduce from a known problem. You can use any result done in class/tutorials.

Solution.[Syamantak] L is decidable (+2). Decider for L ($\langle D\rangle$) :
Given DFA D, build an NFA N^R which accepts all string w^R such that D accepts w. This can be done as follows. The start state of D becomes an accept state in N^R. Turn the accept states of D in to non-accepting states (except if the start state itself is an accept state). Create a new start state which will have ε transitions to every such state. Finally, reverse the transitions. Now convert N^R in to a DFA D^R. (+3)
Finally run a decider for EQ_{DFA} on $\langle D, D^R\rangle$. (+3)
Accept if the decider accepts (+1), reject otherwise (+1)

Q7 [10 points] Let QUADRUPLE-SAT denote the following decision problem: given a Boolean formula ϕ, decide whether ϕ has at least four distinct satisfying assignments. Prove that QUADRUPLE-SAT is NP-complete.(Hint: reduction from SAT. Use a gadget that uses dummy variables.)

Solution.[Vivek+Abhilash] Firstly, QUADRUPLE-SAT is in NP since a certificate would be 4 different assignments of the variables (+2)

Now we prove QUADRUPLE-SAT is NP-hard by reducing SAT (or 3SAT) to QUADRUPLE-SAT. Suppose we are given an instance ϕ is a given instance of SAT. Now we take three additional variables x, y, z and create an instance ϕ' for QUADRUPLE-SAT as follows

$$\phi' = \phi \wedge (x \vee y \vee z) \wedge (\bar{x} \vee y \vee z)$$

(+1 for correctly defining new variables and +2 for correctly writing the new instance)

Now we prove the reduction is correct. Suppose ϕ is an YES instance of SAT. Then it has a satisfying assignment. We take that assignment along with the following four distinct assignments for x, y, z - $(0,0,1), (0,1,0), (1,1,0), (1,1,1)$. All these are clearly satisfying ϕ' since both the additional clauses evaluate to 1 under each of these assignments. (+2 for this part)

Conversely, if ϕ' is a YES instance of QUADRUPLE-SAT, then it has a satisfying assignment. The point is since the variables x, y, z do not appear anywhere except the last two clauses, the same assignment without considering x, y, z satisfies all the rest of the clauses in ϕ' which implies ϕ is satisfiable.(+2 for this part)

(There are several alternative solutions but all of them are mostly around the same idea of taking a few additional dummy variables. All of then get scores if correct. Just throwing in the term 'gadget' just like Salt Bae spinkles salt on his steaks does not yield any score.)