

Student ID: 304990072

Collaborators: \_\_\_\_\_

## CS181 Winter 2019 – Problem Set 5

Due Friday, March 8, 11:59 PM

- This homework is **OPTIONAL**. If completed, it can only help increase your overall homework score. Although this homework is optional, we encourage everyone to look at and attempt the problems as they will be good practice for the final.
- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using  $\text{\LaTeX}$ . Here is one place where you can create  $\text{\LaTeX}$  documents for free: <https://www.overleaf.com/>. The link also has tutorials to get you started. There are several other editors you can use.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- If you don't know the answer, write "I don't know" along with a clear explanation of what you tried. For example: "I couldn't figure this out. I think the following is a start, that is correct, but I couldn't figure out what to do next. [[Write down a start to the answer that you are sure makes sense.]] Also, I had the following vague idea, but I couldn't figure out how to make it work. [[Write down vague ideas.]]" At least 20% will be given for such an answer.  
Note that if you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 8 to 14 hours. You are advised to start early.
- Submit your homework online on Gradescope.

## Problem 1

We follow what the textbook does in Section 5.30 to prove the statement: we prove  $A_{TM} \leq_m \overline{COMPL_{TM}}$ . This is equivalent to  $\overline{A_{TM}} \leq_m COMPL_{TM}$ . Since in the textbook we have show  $A_{TM}$  is Turing-unrecognisable, by Corollary 5.29,  $COMPL_{TM}$  is Turing-unrecognisable. The reducing function  $f$  works as follows (In the textbook,  $A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$ ):

$F =$  "On input  $\langle M, w \rangle$ , where  $M$  is a TM and  $w$  a string:"

- Construct the following two machines,  $M_1$  and  $M_2$ :  
     $M_1$ : Reject on any input  
     $M_2$ : Run  $M$  on  $w$ , if it accepts, reject
- Output  $\langle M_1, M_2 \rangle$

Thus,  $M_1$  accepts nothing, If  $M$  accepts  $w$ ,  $M_2$  also accepts nothing, hence  $L(M_1) \neq \overline{L(M_2)}$ . Conversely, if  $M$  does not accept  $w$ ,  $M_2$  accepts everything, hence  $L(M_1) = \overline{L(M_2)}$ . Thus  $f$  reduces  $A_{TM}$  to  $\overline{COMPL_{TM}}$ .

## Problem 2

In the textbook, we have proved  $E_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } L(M) = \emptyset \}$  is undecidable. We let  $R$  decide  $SUBSET_{TM}$  and construct TM  $S$  to decide  $E_{TM}$  as follows:

$S =$  "On input  $\langle M \rangle$ , where  $M$  is a TM:"

- Run  $R$  on input  $\langle M, M_1 \rangle$ , where  $M_1$  is a Tm that rejects all inputs.
- if  $R$  accepts, accept, if  $R$  rejects, reject.

If  $R$  decides  $SUBSET_{TM}$ , it decides  $E_{TM}$  (because if  $L(M) \subset L(M_1) = \emptyset$ ,  $L(M) = \emptyset$ ). However, since  $E_{TM}$  is undecidable,  $SUBSET_{TM}$  must be undecidable.

### Problem 3

(a) We can define the certificate  $y$  for a  $N \in HALT_\epsilon$  as follows: when  $N$  gets a input  $\epsilon$ , each transition will be recorded as 1. Then we define  $M$  as follows:

$M =$  "On input  $(\langle N \rangle, y)$ , where  $M, N$  are TM:"

- process  $\langle N \rangle$  and  $y$  in parallel:  
simulate  $N$  with input  $\epsilon$ , for each transition, remove/mark the first 1 in  $y$  that has not been removed/marked
- if reach the end of  $y$ , and  $N$  is in an accept state, accept. Otherwise reject.

Hence, for all  $N \in HALT_\epsilon$ , there exists a qualified  $y$ . For all  $N \notin HALT_\epsilon$ , the qualified  $y$  is infinitely long. Hence, any input  $(\langle N \rangle, y)$  would be rejected.

(b) This is because the above method cannot find a proper input to feed in  $N$  in order to find a qualified certificate. Any input chosen can only prove  $N$  halts on a particular input/inputs. Since there are infinitely many possible inputs, even if such a machine defined by the above method accepts, we cannot conclude that  $N$  halts for all inputs.

(c) If  $HALT_{all}$  is a certified language, there exists a Turing machine  $M$  satisfying the conditions specified in the problem.

We can construct a machine  $B$  that takes in input  $x \subseteq \{0, 1\}^*$  as follows:

$B =$  "On input  $x$ :

- By recursion theorem, let  $\langle B \rangle$  be a description of  $B$
- Run  $M(B, x)$   
if  $M$  accepts, loop forever  
if  $M$  rejects, halt.

Thus, when  $M$  accepts,  $B$  actually never halts, which is a contradiction. When  $M$  rejects, there are two situations:

- if  $B \notin HALT_{all}$ , there is no such  $x$  that can make  $M$  accept and halt, and  $B$  halts on all input, a contradiction.
- if  $B \in HALT_{all}$ , that means  $x$  is not the certificate. However, if  $x$  is the certificate, this falls in the situation when  $B$  loops, which is still a contradiction.

Hence, our construction proves  $HALT_{all}$  is not a certified language.