Theory of Computation, Fall 2023
Assignment 8 Solutions

Q1. We first show L_1 is not recursive by reducing H to L_1. Suppose that some Turing machine M_1 decides L_1. Consider the following Turing machine M_H.

M_H = on input "M""w":

1.construct a Turing machine $\widetilde{M}$ as follows

$\widetilde{M}$ = on input x

1. run M on w

2. run M_1 on "$\widetilde{M}$" and the return the result

If M halts on w, then $\widetilde{M}$ halts on every input. If M does not halt on w, $\widetilde{M}$ halts on no input. Therefore, M halts on w if and only if $\widetilde{M}$ halts on at least 2023 strings (that is, M_1 accepts "$\widetilde{M}$"). M_H decides H. This completes the reduction.

Next we show that L_1 is recursive enumerable by presenting a Turing machine M_1' to semidecide it. We label the strings in Σ^* as $s_1, s_2, \ldots$ in incresing length.

M_1' = on input "M":

1.For $i = 2023, 2024, \ldots$

2. For $s = s_1, s_2, \ldots, s_i$

3. Run M on s for i steps

4. If M halts on at least 2023 strings

5. halt

Q2. Note that $L_2 = \overline{L_1}$ (the strings that are not encodings of Turing machines can be safely ignored. Can you see why?) Since L_1 is recursivley enumerable, but not recursive. By the conclusion of Q7 in assignment 7, we have that L_2 cannot be recursively enumerable.

Q3. We already know that $\overline{H}$ is not recursively enumerable. By the conclusion of Q5 in assignment 7, to prove that L_3 is not recursively enumerable, it suffices to reduce $\overline{H}$ to L_3. Given any Turing machine M and its input w, we construct the following Tuing machine

f("M""w") = on input "x":

1.run M on w

If M halts on w, then $f(M, w)$ halts on every input. If M does not halt on w, $f(M, w)$ halts on no input. Therefore, M does not halt on w if and only if there are at least 2023 strings on which $f(M, w)$ does not halt. In otherwords, "M""w" $\in \overline{H}$ if and only if f("M""w") $\in L_3$. f is a reduction from $\overline{H}$ to L_3. This completes the proof.

Remark: if you find the above proof hard to follow. You may try to prove in the following way. Assume that there is a Turing machine that semidecides L_3, and then use this Turing machine to construct another Turing machine to semidecides $\overline{H}$.