Problem Set 10

Due Date ..... November 29, 2022
Name ..... **Alex Ojemann**
Student ID **109722375**
Collaborators .....

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. (See this short intro to LaTeX plus other resources on Canvas.)
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

(I agree to the above, Alex Ojemann). □

3 Standard 23: Complexity Reductions

Problem 1. There is a Canvas quiz. You do not have to submit anything here.

4 Standard 22: Computability Classification

Problem 2. Classify the following languages as below, and justify your answer:

- (RE) recursive enumerable (r.e.),
- (CO-RE) co-r.e. (i.e. the complement is r.e.),
- (BOTH) both r.e. and co-r.e., or
- (NEITHER) neither r.e. nor co-r.e.

Note this means showing two things each: L is r.e. or not, and $\sim L$ is r.e. or not.

(a) $L = \{M\#x \mid \text{on input } x, M \text{ never moves its head to the left}\}$

Proof. RE:

If M halts before reaching the end of the input string without moving left, we accept. Otherwise, we count the number of steps taken to reach the end of the input. If that number is greater than the length of the input string, M must've moved left somewhere within the string, so we reject. If that number is the same as the length of the input string, we then count the number of steps taken in reading the blank tape squares that follow the input. If we reach a number of blanks equal to the number of states in our finite control without moving left, we accept since there must be no possible states that could move the tape left. Therefore, L is r.e.

co-RE:

$\sim L = \{M\#x \mid \text{on input } x, M \text{ moves its head to the left at least once}\}$

We run M on the input string infinitely until M moves its head left. If it does, we accept. Otherwise we loop. So, $\sim L$ is r.e. Therefore, L is co-r.e. □

(b) $L = \{M \mid M \text{ accepts exactly 3 inputs}\}$

Proof. not RE:

Once M has accepted 3 inputs and is now searching through infinite strings to try to find another, we reduce $\sim HP$ to L. Let $M\#x$ be an instance of $\sim HP$. We construct a TM M' such that M' simulates M on any input string. We map $M\#x -> M'\#x \in L$. As $\sim HP$ isn't RE, neither is L.

This is the case because L will reject if a fourth accepted is found, but will loop forever if it isn't.

not co-RE:

We will show that $\sim L = \{M \mid M \text{ accepts more or fewer than 3 times}\}$ is not RE.

In the case in which M accepts more than 3 inputs, L will reject once the fourth accepted input is found. However, we can reduce $\sim HP$ to the case in which M accepts fewer than three inputs. Let K be the subset of $\sim L$ where M accepts fewer than 3 times. Let $M\#x$ be an instance of $\sim HP$. We construct a TM M' such that M' simulates M on any input string. We map $M\#x -> M'\#x \in K$. As $\sim HP$ isn't RE, neither is K or $\sim L$, so L must not be co-RE.

This is the case because K will reject once the third accepted input is found but will loop forever if it isn't. □