# CSCI-UA.0453-001: Honors Theory of Computation

## Problem Set 3

All problems are worth 10 points.

Collaboration is allowed, but you must write your own solutions. Write the names of your collaborators (and your own!).

Unless stated otherwise, you must show all intermediate steps and give proper justification or proof.

### Problem 1

A *Turing machine with stay put instead of left* is similar to an ordinary Turing machine, but the transition function is of the form:

$$Q \times \Gamma \mapsto Q \times \Gamma \times \{R, S\}$$

At each point the machine can move its head to right or let it stay in the same position. Show that this Turing machine variant is *not* equivalent to the usual version. What class of languages do these machines recognize ?

### Problem 2

Show that the collection of Turing-recognizable languages is closed under the operation of (a) union (b) concatenation (c) star and (d) intersection. What about complementation operation ?

### Problem 3

Show that the following language is decidable:

$$INFINITE_{CFG} = \{\langle G\rangle \mid G \text{ is a context-free grammar such that } L(G) \text{ is infinite}\}$$

### Problem 4

Show that the following language is decidable:

$$L = \{\langle R, S\rangle \mid R \text{ and } S \text{ are regular expressions such that } L(R) \subseteq L(S)\}$$

### Problem 5

Show that the following language is undecidable:

$$A = \{\langle M\rangle \mid M \text{ is a Turing machine that accepts } w \text{ whenever it accepts } w^R\}$$

Here $w^R$ denotes the reverse of string $w$.

**Problem 6** In this problem, we explore the notion of *oracle reducibility.* If $A$ is a language, then a *Turing machine with oracle $A$* is a Turing machine with a "magical" subroutine that decides membership in $A$. In other words, the subroutine, when given a string $w$, tells the machine whether or not $w \in A$. Let

$$HALT_{TM} = \{\langle M, x\rangle \ \mid M \text{ is a Turing machine that halts on } x\}$$

Show that there is a Turing machine with oracle $HALT_{TM}$ that decides the following problem with only *two* questions to the oracle: Given three (machine, input) pairs $\langle M_1, x_1\rangle, \langle M_2, x_2\rangle, \langle M_3, x_3\rangle$, decide for each pair whether the Turing machine halts on the corresponding input.

Note: This is trivial if one allows *three* questions. Just ask the oracle whether $\langle M_i, x_i\rangle \in HALT_{TM}$ for $i = 1, 2, 3$.