CS 271: Automata and Computation Theory **Spring 2014**

Problem Set #3

Due: Thursday, April 3, 2014

Problem 1 Prove that $L = \{\langle M \rangle \mid M \text{ is a DFA that accepts } w^{\mathcal{R}} \text{ whenever it accepts } w\}$ is decidable.

Problem 2 Prove that a language L is decidable if and only if $L^{\mathcal{C}}$ is decidable.

Problem 3 Consider the problem of determining whether a computer program written in Python ever prints out "`Hello world!`" when run on some input w. Prove that this problem is undecidable. Formally, consider the language

$$\{\langle P, w \rangle \mid P \text{ is a Python program that, on input } w, \text{ prints } \texttt{Hello world!}\}$$

and show that it is undecidable.

Problem 4 Consider the problem of determining whether a TM M on input w ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and prove that it is undecidable.

Problem 5 Show that the class of Turing-recognizable languages is not closed under complement.

Problem 6 Consider the language

$$L = \{\langle M, w, q \rangle \mid M \text{ is a TM that when run on input } w \text{ never enters state } q\}.$$

If L is decidable, describe a TM that decides it. If L is not decidable, prove it by giving a reduction from an undecidable language L'. That is, show $L' \leqslant L$.

Problem 7 In class, we proved that $A_{\text{TM}} \leqslant \text{HALT}_{\text{TM}}$ (although we didn't use the terminology of reductions). Prove that $\text{HALT}_{\text{TM}} \leqslant A_{\text{TM}}$.

Problem 8 Prove that EQ_{CFG} is co-Turing-recognizable by describing a TM that recognizes the complement.

Problem 9 Prove that EQ_{CFG} is undecidable.

Problem 10 Use the results of Problems 8 and 9 to show that EQ_{CFG} is not Turing-recognizable.