CS:5340 Homework 2 Hints

1. We say that a Turing machine M *accepts* a string $w \in \{0,1\}^*$ if on input w, M halts and outputs 1. A Turing machine M is said to have property R if whenever M accepts w it accepts w^R. (**Note:** w^R denotes the string obtained by reversing string w; e.g., $(\texttt{011})^R$ is `110`.) Define a function $R : \{0,1\}^* \to \{0,1\}$ as follows: $R(\alpha) = 1$ if M_α has property R and $R(\alpha) = 0$ otherwise. Prove that the function R is uncomputable.

   Hint: For any binary string $x \in \{0,1\}^*$, consider the function $\texttt{HALT}_x$ defined as follows: $\texttt{HALT}_x(\alpha) = 1$ if M_α halts on input x and $\texttt{HALT}_x(\alpha) = 0$ if M_α does not halt on input x.

   Reduce the problem of computing $\texttt{HALT}_x$, for any $x \in \{0,1\}^*$, to the problem of computing R. In other words, if there is a TM M_R that computes R then one could use this TM to contruct a new TM M_x that could solve $\texttt{HALT}_x$. This step is the heart of the proof and conceptually most challenging.

   Finally, argue that since `HALT` is uncomputable, $\texttt{HALT}_x$ is uncomputable for some $x \in \{0,1\}^*$. This contradicts the existence of M_R.

2. Problem 1.12 (Chapter 1, Page 35).

   Hint for Part 3(a): Fix $x \in \{0,1\}^*$ and again consider $\texttt{HALT}_x$ and show that Rice's Theorem implies that $\texttt{HALT}_x$ is uncomputable. This is fairly straightforward if you understand the statement of Rice's Theorem. Then argue that if `HALT` were computable, then $\texttt{HALT}_x$ would also be computable (for every $x \in \{0,1\}^*$) and this contradicts the implication about $\texttt{HALT}_x$ derived from Rice's Theorem.

   Hint for Part 3(b): The textbook provides a hint for this on page 531 and again $\texttt{HALT}_x$ plays an important role. Here I expand on this hint. Consider a non-trivial set S of partial functions and without loss of generality suppose that Φ (the function that is not defined anywhere) is not in S. Since S is non-empty it contains a partial function f that is computed by some TM M_f. Also, since $f \neq \Phi$, f is defined for some $x \in \{0,1\}^*$.

   Now *reduce* $\texttt{HALT}_x$ to f_S. In other words, assume that the function f_S is computable by some TM M_S and use the existence of M_S to design a TM M_x for $\texttt{HALT}_x$. As in the previous problem, this would contradict the uncomputability of `HALT`.