Solutions to Cpt S 317 Homework #12

1. (1). Show that if both L and (the complement) $\overline{L}$ are r.e., then L is recursive. (2). Show that r.e. languages are not closed under complement.

Proof. (1). Assume L and $\overline{L}$ are r.e.. That is, we have two TMs M and $\bar{M}$ such that $L = L(M)$ and $\overline{L} = L(\bar{M})$. Now we will construct a TM M' recognizing (i.e., always halts and gives yes/no answers) L. Given any input word w, M' works as follows. M' simulates M and $\bar{M}$ on input w in parallel. Since either $w \in L$ (i.e., M gives yes on w) or $w \in \overline{L}$ (i.e., $\bar{M}$ gives yes on w), whenever M' sees a yes answer, M' halts (and obviously, always halts). M' accepts (says yes) w if the yes was given by M, M' rejects (says no) w if the yes was given by $\bar{M}$. Clearly, M' recognizes L, i.e., L is recursive.

(2). Consider the universal language $L_u = \{\langle M, w\rangle : w \in L(M)\}$. We know that L_u is r.e. If the complement $\overline{L_u}$ were also r.e., then L_u, from (1) above, would be recursive. But this is impossible since L_u is not recursive.

2. Show that recursive languages are closed under complement.

Proof. Let L be recursive and be recognized by M. By switching yes/no for M, it can be used to recognize $\bar{L}$. Thus, $\bar{L}$ is also recursive.

3. Recall that we use M_i to denote the "i-th Turing Machine". That is, the string encoding $\langle M_i\rangle$ of M_i is exactly w_i, the i-th word in the dictionary ordering. Show that, for each Turing machine M, there are infinitely many i such that $L(M) = L(M_i)$ (i.e., M and M_i accept the same language.). (If a program is understood as a Turing machine, what does this exercise say about the program?)

Proof. Given a TM M. Of course, we can add any number of "garbage" instructions to it. For instance, if M has n states $q_1, \cdots, q_n$, then expand the states of M by adding k new states $q_{n+1}, \cdots, q_{n+k}$ and instructions

$$\delta(q_{n+1}, a) = (q_{n+1}, a, R)$$
$$\vdots$$
$$\delta(q_{n+k}, a) = (q_{n+k}, a, R)$$

Denote the new machine as M^k. Of course the new machine, when running, will not execute any of the above new instructions. It is noticed that M and M^k behave the same, i.e., $L(M) = L(M^k)$. Let M^k be the i_k-th Turing machine M_{i_k}. Since the encodings of M^k are distinct for each k. That is,

i_k are also distinct for each k. The result follows by the fact that we have infinitely many choices for k.

This exercise says that for each program P, there are infinitely many equivalent programs. Some of them are good, many of them are bad. In particular, by adding a number of garbage instructions (like $x := x + 1 - 1$), any program can be made as long as you want.

4. For a TM M, a L-instruction (resp. R-instruction) is a move in the form of $\delta(q, a) = (p, b, L)$ (resp. $\delta(q, a) = (p, b, R)$). Show that the following problem is **decidable**:

Given a TM M, whether M contains the same number of L-instructions and R-instructions?

Proof. Define $L = \{\langle M\rangle : M$ contains the same number of L-instructions and R-instructions$\}$. Then L is a context-free language. Therefore, the problem in the exercise is equivalent to check $\langle M\rangle \in L$, i.e., the membership problem for a context-free language. Therefore, the result follows by noticing that the membership problem is decidable for context-free languages.

5. Show that the following problem is also **decidable**:

Given a TM M, whether there exists a TM M' such that M' contains the same number of L-instructions and R-instructions, and $L(M') = L(M)$?

Proof. We claim any TM M can be modified to a TM M' that contains the same number of L-instructions and R-instructions. If this can be done, the answer to the above problem is always yes. Of course, it is decidable.

The modifications is as follows. Given a TM M, if M already contains the same number of L-instructions and R-instructions, then take $M' = M$. If however, there are k more L-instructions than R-instructions in M for some k, then we add k number "garbage" R-instructions to M and the result is M'. Assume M has n states $q_1, \cdots, q_n$. We expand the states of M by adding k new states $q_{n+1}, \cdots, q_{n+k}$ and the following garbage R-instructions:

$$\delta(q_{n+1}, a) = (q_{n+1}, a, R)$$
$$\vdots$$
$$\delta(q_{n+k}, a) = (q_{n+k}, a, R)$$

It is obvious that $L(M') = L(M)$ since these newly added instruction will not be executed by M'. The case when M has more R-instructions than

L-instructions is symmetric, i.e., by adding k number of L-instructions to M.