COT 6315/CIS 4930 Sample Test 2 (Solutions)

1. (a) The set QUITEEQUAL is decidable (a simple algorithm counts the numbers of a's, b's and c's in the input string, always halts, and says yes when the three numbers are the same and no otherwise), but not context free. Proof of latter by contradiction. Suppose that QUITEEQUAL is context free. Then the intersection of QUITEEQUAL and regular language $a^*c^*b^*d^*$ should be context free as well. However this intersection $a^nc^mb^nd^m$ is not context free (see problem 2 (iii) of the second homework). Therefore QUITEEQUAL is not context free.

   (b) The set WEIRD is regular. It can be represented as $L_1 \cap L_2 \cap L_3$ where $L_1 = (1^*01^*01^*)^*$ is a regular language of strings that have even number of zeroes, $L_2 = \overline{(0^*10^*10^*10^*10^*10^*)^*}$ is a regular language of strings that have a number of 1's that is not a multiple of 5, and $L_3 = ((0^*1^*)^*1)^*(00)^*000$ is a regular language of strings that end with an odd number of 0's (atleast 3). Since set of regular languages is closed under intersection, set WEIRD is regular.

   (c) The language VERYDIFFERENT(L) is regular if L is regular, since VERYDIFFERENT(L)= $\overline{\Sigma^* \circ L \circ \Sigma^*}$ and set of regular languages is closed under operations of complement and concatenation.

   (d) The set BALANCED is context free but not regular. It is context free because there is PDA for it (We want to check whether $k-l=n-m$. This can be done as follows. Push all as, pop an a for every input b, push remaining bs (if any). At this stage stack contains either $k-l$ as or $l-k$ bs. There are two cases. (1) If stack contains as then pop an a for every input c, when stack becomes empty push all remaining cs, pop a c for every input d, accept if stack is empty after last input d. (2) If stack contained bs then push all c on a stack, pop a symbol (c or b) for every d, accept if stack is empty after last input d.

   The set BALANCED is not regular because BALANCED $\cap a^*b^* = a^nb^n$ is not a regular language, and set of regular languages is closed under intersection.

2. (a) This language is recursively enumerable but not recursive. It is r.e because we can run both M_1 and M_2 on input x and *accept* if both M_1 and M_2 halt and accept x in the same number of instruction steps. This language is not recursive because $A_{TM} = \{< M, w >\}$ can be reduced to it (take $M_1 = M_2 = M, x = w$).

   (b) This language is recursive - run M_1 on x for exactly $|x|^2$ steps, run M_2 on x for exactly $|x|^2$ steps, *accept* if both M_1 and M_2 halted and accepted x in exactly $|x|^2$ steps.

(c) This language is recursively enumerable but not recursive. It is not recursive because $E_{TM} = \{< M >\}$ can be reduced to it (take $M_1 = M_2 = M$).

In order to show that language is recursively enumerable we will enumerate all strings $x_1, x_2, x_3 \ldots \in \Sigma^*$. Run M_1 and M_2 on x_1 for 1 step. Run M_1 and M_2 on x_2 for 1 step and continue on x_1 for 1 step. Run M_1 and M_2 on x_3 for 1 step and continue on x_2 and x_1 for 1 step, etc. If there is an x_k such that both M_1 and M_2 both halt on x_k and accept in the same number of steps, then this x_k will be found by this procedure.