12 1. Choosing from among (REC) recursive, (RE) re non-recursive, (coRE) co-re non-recursive, (NR) non-re/non-co-re, categorize each of the sets in a) through d). Justify your answer by showing some minimal quantification of some known recursive predicate.

   a.) \{ f | \text{ whenever } x>y \text{ and } f(x)\downarrow \text{ and } f(y)\downarrow \text{ then } f(x)>f(y) \} \text{ coRE}

   Justification: \( \forall x \forall y \forall t \left( \text{STP}(x,f,t) \land \text{STP}(y,f,t) \land (x>y) \implies (\text{VALUE}(x,f,t)>\text{VALUE}(y,f,t)) \right) \)

   b.) \{ f | \text{ size of range}(f) \text{ is at most 1} \} \text{ coRE}

   Justification: \( \forall x \forall y \forall t \left( \text{STP}(x,f,t) \land \text{STP}(y,f,t) \implies (\text{VALUE}(x,f,t) = \text{VALUE}(y,f,t)) \right) \)

   I allowed \( \exists K \forall x \forall t \left[ \text{STP}(x, f, t) \implies (\text{VALUE}(x, f, t) = K) \right], \text{ which is NR} \)

   c.) \{ <f,x> | f(x) \text{ converges in at most } x^2 \text{ steps} \} \text{ REC}

   Justification: \( \text{STP}(x, f, x^2) \)

   d.) \{ f | \text{ domain}(f) \text{ contains the value 0} \} \text{ RE}

   Justification: \( \exists t \text{STP}(0, f, t) \)

12 2. Let set \( A \) be recursive, and both \( B \) and \( C \) be re non-recursive. Choosing from among (REC) recursive, (RE) re non-recursive, (NR) non-re, categorize the set \( D \) in each of a) through d) by listing all possible categories. No justification is required. You may find it useful to know that, if \( E \) is recursive (re, non-re) then so is \( E \) even = \{ 2x | x \in E \} and \( E \) odd = \{2x+1 | x \in E\}.

   a.) \( D \supseteq B \) \text{ REC, RE, NR}

   b.) \( D = \sim A \) \text{ RE}

   c.) \( D = B \cup C \) \text{ REC, RE}

8 3. Let \( S \) be an arbitrary non-empty re set. Furthermore, let \( S \) be the range of some partial recursive function \( f_s \). Show that \( S \) is the range of some primitive recursive function, call it \( h_s \).

   Let \( h_s(<x,t>) = a \ast (1 - \text{STP}(x, f_s, t)) + \text{VALUE}(x, f_s, t) \ast \text{STP}(x, f_s, t) \)

   where \( 'a' \) is some arbitrary element of \( S \). Such an \( 'a' \) exists since \( S \) is non-empty.

   First, \( h_s \) is primitive since the constants \( C_a \) and \( C_1 \), \( \text{STP} \), \( \text{VALUE} \), \( + \), \( \ast \) and \( – \) are all primitive recursive and the primitive recursive functions are closed under composition.

   Now, given any element \( y \) of \( S \), there must exist some \( x \) such that \( f_s(x) = y \). For such an \( x \), there must exist a \( t \) such that \( \text{STP}(x, f_s, t) \) and \( \text{VALUE}(x, f_s, t) = y \). Of course, if such an \( <x,t> \) exists, there are an infinite number of these \( <x,t'> \) for \( t' > t \). Clearly, for each such \( <x,t> \), \( h_s(<x,t>) = y \).

   If, on the other hand, \( \sim \text{STP}(x, f_s, t) \), then \( h_s(<x,t>) = a \). Thus, all and only the elements in \( S \) are enumerated by \( h_s \), exactly what we need.

8 4. Prove that the Halting Problem (the set \( \text{HALT} = K_0 = L_0 \)) is not recursive (decidable) within any formal model of computation. (Hint: A diagonalization proof is required here.)

   Look at notes.

8 5. Using reduction from the known undecidable set \( \text{HasZero}, HZ = \{ f | \exists x f(x) = 0 \} \), show the non-recursiveness (undecidability) of the problem to decide if an arbitrary partial recursive function is in the set \( \text{Identity}, ID = \{ f | \forall x f(x) = x \} \).

   \( HZ = \{ f | \exists x \exists t \left[ \text{STP}(x, f, t) \land \text{VALUE}(x, f, t) = 0 \right] \} \)

   Let \( f \) be the index of an arbitrary effective procedure.

   Define \( g_f(y) = y \ast \exists x \exists t \left[ \text{STP}(x, f, t) \land \text{VALUE}(x, f, t) = 0 \right] \)

   If \( \exists x f(x) = 0 \), we will find an \( x \) and a run-time \( t \), and so we will return \( y \ast 1 \)

   If \( \forall x f(x) \neq 0 \), we will diverge in the search process and never return a value.

   Thus, \( f \in HZ \) iff \( g_f \in ID \).

7 6. Assuming only the primitive recursiveness of \( C_a(x_1,\ldots,x_n) = a : \text{constants} \), \( \Pi_1^P (x_1,\ldots,x_n) = x_i : \text{projection} \), and \( S(x) = x+1 : \text{increment} \), and that the prf’s are closed under composition and primitive recursion (iteration), show the primitive recursiveness of the function \( \text{Max}(x,y,z) \).

   \( \text{pred}(0) = 0, \text{pred}(y+1) = y; -x(0) = x, -(x,y+1) = \text{pred}(-x,y); \text{signum}(0) = 0, \text{signum}(y+1) = 1; > (x,y) = \text{signum}(-x,y); +(x,0) = x, +(x,y+1) = S(x,y); \text{Max}(x,y) = x > (x,y) + y > (y,x) \)
7. Demonstrate a **Register Machine** where the numbers \( x \), \( y \) and \( z \) are in registers \( R2 \), \( R3 \) and \( R4 \), respectively, all other registers being zero. \( \text{Max}(x, y, z) \) ends up in register \( R1 \). The final contents of all registers except \( R1 \) are unimportant.

1. \( \text{DEC2}[2,5] \)
2. \( \text{DEC3}[3,3] \)
3. \( \text{DEC4}[4,4] \)
4. \( \text{INC1}[1] \)
5. \( \text{DEC3}[6,8] \)
6. \( \text{DEC4}[7,7] \)
7. \( \text{INC1}[5] \)
8. \( \text{DEC4}[9,10] \)
9. \( \text{INC1}[8] \)
10. 

5. Present an ordered **Factor Replacement System** (ordered rules of form \( ax \rightarrow bx \)) that, when started on the number \( 3^x 5^y 7^z \), terminates on the number \( 2^{\text{Max}(x,y,z)} \). When you end, no prime factor other than perhaps 2 should appear in the final number.

\[
\begin{align*}
3 \cdot 5 \cdot 7 \cdot x &\rightarrow 2 \cdot x \\
3 \cdot 5 \cdot x &\rightarrow 2 \cdot x \\
3 \cdot 7 \cdot x &\rightarrow 2 \cdot x \\
5 \cdot 7 \cdot x &\rightarrow 2 \cdot x \\
3 \cdot x &\rightarrow 2 \cdot x \\
5 \cdot x &\rightarrow 2 \cdot x \\
7 \cdot x &\rightarrow 2 \cdot x
\end{align*}
\]