Assignment # 9.1a Key

1. Use quantification of an algorithmic predicate to estimate the complexity (decidable, re, co-re, non-re) of each of the following, (a)-(d):

a)NonTrivial = { f | for some x and y, x ≠ y, $\varphi_f(x)\downarrow$, $\varphi_f(y)\downarrow$ and $\varphi_f(x) \neq \varphi_f(y)$ }

∃ <x,y,t>[STP(f,x,t) & STP(f,y,t) & (x≠y) & (VALUE(f,x,t) ≠ (VALUE(f,y,t))]

RE

Assignment # 9.1b Key

b) NoRepeats = { f | for all x and y, $\varphi_f(x)\downarrow$, $\varphi_f(y)\downarrow$, and $x \neq y \Rightarrow \varphi_f(x) \neq \varphi_f(y)$ }

$\forall$<x,y> $\exists$t [STP(f,x,t) & STP(f,y,t) & (x≠y ⇒VALUE(f,x,t) ≠ (VALUE(f,y,t))]

Non-RE, Non-Co-RE

Assignment # 9.1c Key

c) FIN = { f | dom(f) is finite }

$\exists K \forall \langle x,t \rangle \, [x > K \Rightarrow \sim STP(f, x, t)]$

Non-RE, Non-Co-RE

Assignment # 9.2a Key

2. Let sets A be recursive (decidable) and B be re non-recursive (undecidable).
   Consider C = { z | z = x*y, where x ∈ A and y ∈ B }. For (a)-(c), either show sets A and B with the specified property or demonstrate that this property cannot hold.

a) Can C be recursive?

YES. Consider A = {0}. B = Halt. C = {0}

Assignment # 9.2b Key

b) Can C be re, non-recursive?

YES. Consider A = { 1 }. B = Halt. C = B = Halt. This is Halt which is the classic re, non-recursive set.

Assignment # 9.2c Key

c) Can C be non-re?

No. Can enumerate C as follows.

First if A is empty then C is empty and so RE by definition.

If A is non-empty then A is enumerated by some algorithm f_A as recursive sets are RE.

As B is non-recursive RE, then it is non-empty and enumerated by some algorithm f_B.

Define f_C by $f_C(\langle x,y \rangle) = f_A(x) * f_B(y)$. f_C is clearly an algorithm as it is the composition of algorithms. The range of f_C is then $\{ z \mid z = x * y$, where $x \in A$ and $y \in B \} = C$ and so C must be RE.