Assignment # 9.1a Sample 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)REPEATS = { f | for some x and y, x ≠ y, f(x)↓, f(y)↓ and f(x) == 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) DOUBLES = { f | for all x, f(x)↓, f(x+1)↓ and f(x+1)=2*f(x) }

$\forall$x $\exists$t [STP(f,x,t) & STP(f,x+1,t) & (2*VALUE(f,x,t) = (VALUE(f,x+1,t))]

Non-RE, Non-Co-RE

Assignment # 9.1c Key

c) DIVEVEN = { f | for all x, f(2*x)↑ }

∀ <x,t> [~STP(f,2*x,t)]

Co-RE

Assignment # 9.1d Key

d) QUICK10={ f | f(x), for all 0≤x≤9, converges in at most x+10 steps }

STP(f,0,10) & STP(f,1,11) & … & STP(f,9,19)

or

$\forall x_{0\leq x\leq 9}$ [STP(f,x,x+10)]

REC

Assignment # 9.21 Key

1. Let sets A be recursive (decidable) and B be re non-recursive (undecidable).
   Consider C = { z | min(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 non-recursive?

YES. Consider A = { 2x | x $\in$ *N* }. B = { 2x+1 | x $\in$ Halt}. C = A $\cup$ B. This is semi-decidable but non rec as Halt is reducible to C.

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