COT 5310 Fall 2006 Midterm#1 Sample Name: KEY

16 **1**. Choosing from among **(REC) recursive**, **(RE) re non-recursive**, **(NR) non-re**, categorize each of the sets in a) through d). Justify your answer by showing some minimal quantification of some known recursive predicate or by another clear and convincing short argument.

a.) { <f,x>| ∀(t>x) STP(x,f,t) } *REC*

Justification: ***This is equivalent to STP(x, f, x+1).***
That's because STP(x, f, s) implies that STP(x, f, t), where t ≥ s.

b.) { f | ∀x f(x) is a prime number **}** *NR*

Justification: ***This is ∀x ∃t [STP(x, f, t) && isPrime(Value(x, f, t))]***
Some of you thought it was REC because primality is recursive, but that's not enough since we are looking at all x, and divergence is necessary.

c.) { f | ∃x f(x)↓ } *RE*

Justification: This is ∃<t,x> [STP(x, f, t)]. It's just L_{ne}.

d.) { x | ∀f f(x)↑ } *REC*

Justification: ***No x has this property. An easy way to see this is by f(x) = x (a base primitive recursive function). This shows that all x are included in the domain of some function.***

15 **2**. Let set **A** be recursive, **B** be re non-recursive and **C** be non-re. Choosing from among **(REC) recursive**, **(RE) re non-recursive**, **(NR) non-re**, categorize the set **D** in each of a) through e) by listing **all** possible categories. No justification is required.

a.) D = ~A ∪ B *REC, RE*

b.) D = A ∩ ~C *REC, RE, NR*

c.) D = B – A *REC, RE*

d.) D = C ∪ ~C *REC*

e.) A ⊆ D ⊆ B *REC, RE, NR*

7 **3.** Using the definition that **S** is semi-decidable iff **S** is the domain of some procedure g_S (partial recursive function), prove that if both **S** and its complement **~S** are semi-decidable then **S** is decidable. Note: To get full credit, you must show the characteristic function for **S**, χ_S, and argue convincingly (not formally) that the function you specified is the correct one for **S**. Hint: The **STP** predicate is useful here. You may also use any other known total recursive functions to attack this.

$\chi_S(x) = STP(x, g_S, \mu t\ [\ STP(x, g_S, t] \ || \ STP(x, g_{\sim S}, t)\ ]$

Since x is either in S or its complement, then x is either in the domain of g_S or of $g_{\sim S}$, then there is some t such that either STP(x, g_S, t) or STP(x, $g_{\sim S}$, t). Of course, x can only be in the domain of one of these. Now, once we get t, we can just see if STP(x, g_S, t). If so, x ∈ S, else x ∉ S.

10 **4.** Prove that the **Uniform Halting Problem** (the set **TOTAL**) is not recursively enumerable within any formal model of computation. You may assume the existence of a universal machine. (Hint: A diagonalization proof is required and the most direct approach is via recursive functions.)

Assume TOTAL is re, then it is the range of some total function, T. Define C by C(x) = Univ(T(x), x) + 1. Now, since T(x) enumerates the indices of total functions, then for each x, Univ(T(x), x) is defined (converges). Thus, C is total. Thus, there is some c, C's index in the enumeration T, such that C(x) = Univ(T(c), x). Consider C(c). By the definition of C,
C(c) = Univ(T(c), c) + 1 by the definition of C. But then
C(c) = Univ(T(c), c) + 1 = C(c) + 1 by the meaning of Univ (see above). This means that
C(c) = C(c) + 1 this is a contradiction since C is total.
Since the only non-constructive step was assuming that TOTAL is re, we can conclude that TOTAL is not re, as was desired.

6 **5.** Using reduction from the known undecidable **Non-Empty Problem, NE = { f | ∃x f(x)↓ }**, show the non-recursiveness (undecidability) of the problem to decide if an arbitrary effective procedure **g** has a fixed point; that is, whether or not **g(x) = x**, for some **x**.

Let f be an arbitrary effective procedure (actually the index of one).
Define $g_f(x) = x + f(x) – f(x)$ or more formally $g_f(x) = G(f, x) = x + Univ(f, x) – Univ(f, x)$.
If f converges somewhere then g_f has a fixed point since it is the identity function. Otherwise, g_f diverges everywhere and has no fixed point. This can also be approached with
$g_f(x) = signum(\mu t\ [STP(x, f, t)] + 1) * x$. Think about it.

9 **6.** Let **S** be an re non-recursive set, and let **T** be an infinite recursive set. Define **E = { z | z = x - y**, where **x ∈ S** and **y ∈ T** and **x – y = 0** if **x<y** }.You may assume that **S** is the range of a total recursive function $\mathbf{f_S}$, and domain of a partial recursive function $\mathbf{g_S}$. You may also assume **T** has a total characteristic function χ_T. Can **E** be re non-recursive?

Consider S = { 2*x | x ∈ HALT }. This set is re, non-recursive since a decision procedure for either the Halt or S implies one for the other.
Let T = {0} ∪{2x+1 | x∈ℵ}. T is infinite and recursive – it's the union of the odd numbers and 0.
E = S ∪{0} ∪{2x+1 | x∈ℵ}. The reason all odd numbers are in E is that S is infinite and hence has no largest number and every odd number is the difference between an even number and an odd number. {0} is there because all we have to do is subtract a larger odd number from some even number to get 0. Now x ∈ S iff x is even and (x==0 or x ∈ E).

Domination (x , y, z) = 1 if **(x+y) > z; 0** otherwise.

6 **7.** Assuming only the recursiveness of the base functions

$C_a(x_1,\ldots,x_n) = a$: constants; $I_i^n(x_1,\ldots,x_n) = x_i$: identity (projection); $S(x) = x+1$: increment

and the standard arithmetic operators +, -, * and // and that the recursive functions are closed under composition, primitive recursion and minimization, show the recursiveness of **Domination**

Non-Zero(0) = C_0; Non-Zero(x+1) = C_1(x, Non-Zero(x)) : Primitive Recursion
>(x, y) = Non-Zero(-(x, y)) : Composition
Add3(x, y, z) = +(I_1^3(x, y, z), I_2^3(x, y, z)) : Composition
Domination(x, y, z) = >(Add3(x, y, z), I_3^3(x, y, z)) : Composition

6 **8**. Demonstrate a **Register Machine** where the numbers **x**, **y** and **z are** in registers **R2**, **R3** and **R4**, respectively, all other registers being zero. **Domination (x , y, z) = (1** if **(x+y) > z; 0)** otherwise ends up in register **R1**. The final contents of all registers except **R1** are unimportant.

1. DEC_2(2, 3)
2. DEC_4(1, 5)
3. DEC_3(4, 6)
4. DEC4(3, 5)
5. INC_1(6)
6.

6 **9**. Present a **Factor Replacement System** (ordered rules of form **ax → bx**) that, when started on the number $\mathbf{3^x\ 5^y\ 7^z}$, terminates on the number $\mathbf{2=2^1}$, if **x + y > z**, and on $\mathbf{1=2^0}$, otherwise (**Domination** encoded). When you end, no prime factor other than perhaps **2** appears in the final number.

3 • 7 x → x
5 • 7 x → x
3 x → 2x
5 x → 2x
4 x → 2x
7 x → x