COT 6410 Spring 2015 Midterm#1 Name: __KEY__

12 **1**. Choosing from among **(REC) recursive**, **(RE) re non-recursive, (coRE) co-re non-recursive**, **(NRNC) 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 | f is a Fibonacci function, i.e. f(0)=f(1)=1 and f(x+2)=f(x)+f(x+1) } ___*NRNC*___

Justification: ***∀x∃t [STP(f,0,t) && VALUE(f,0,t)=1 && STP(f,1,t) && VALUE(f,1,t)=1 && STP(f,x,t) && STP(f,x+1,t) && STP(f,x+2,t) && (VALUE(f,x+2,t) = VALUE(f,x+1,t) + VALUE(f,x,t))]***

b.) { f | if f(x) converges, it does so in more than (2^x) units of time } ___*coRE*___

Justification: ***∀x ~ STP(f,x,2^x)***

c.) { <f,x> | if f(x) converges, it does so in more than (2^x) units of time } ___*REC*___

Justification: ***~ STP(f,x,2^x)***

d.) { f | f(x) = f(x+1) for at least one value of x } ___*RE*___

Justification: ***∃<x,t> [STP(f,x,t) && STP(f,x+1,t) && (VALUE(f,x,t) = VALUE(f,x+1,t))]***

2 **2.** Looking back at Question 1, which of these are candidates for using Rice's Theorem to show their unsolvability? Check all for which Rice Theorem might apply.

a) __√__ **b)** ____ **c)** ____ **d)** __√__

6 **3**. 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 d) by listing **all** possible categories. No justification is required.

a.) D = C − A (set difference) ___*REC, RE, NR*___

b.) A ⊆ D (set containment) ___*REC, RE, NR*___

c.) D = A × B (cross product) ___*RE, REC (only when A = ∅)*___

d.) D = A − B (set difference) ___*REC, NR*___

4. Define **NON_TRIVIAL_RANGE = (f | |range(f)| > 1 }**.

2 **a.)** Show some minimal quantification of some known recursive predicate that provides an upper bound for the complexity of this set. (Hint: Look at **c.)** and **d.)** to get a clue as to what this must be.)

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

5 **b.)** Use Rice's Theorem to prove that **NON_TRIVIAL_RANGE** is undecidable.

First show NON_TRIVIAL_RANGE is non-trivial:

$I(x) = x \in NON\_TRIVIAL\_RANGE \quad C_0(x) = 0 \notin NON\_TRIVIAL\_RANGE$
and thus, the set and its complement are non-empty as required.

Let f, g be two arbitrary indices (functions) such that range (φ_f) = range (φ_g).

$f \in NON\_TRIVIAL\_RANGE$ ***iff*** $|range(\varphi_f)| > 1$ ***Definition of NON_TRIVIAL_RANGE***

iff $|range(\varphi_g)| > 1$ $range(\varphi_f) = range(\varphi_g)$

iff $g \in NON\_TRIVIAL\_RANGE$ ***Definition of NON_TRIVIAL_RANGE***

This weak form of Rice's Theorem shows NON_TRIVIAL_RANGE to be undecidable.

4 **c.)** Show that $K_0 \leq_m$ **NON_TRIVIAL_RANGE**, where $K_0 = \{ <x,y> \mid \varphi_x(y)\downarrow \}$.

Let <x,y> be an arbitrary pair of natural numbers.
Define $f_{x,y}(z) = \varphi_x(y) - \varphi_x(y) + z$

$<x,y> \in K_0$ ***implies*** $\forall z\, f_{x,y}(z) = z$ ***implies*** $f_{x,y} \in NON\_TRIVIAL\_RANGE$

$<x,y> \notin K_0$ ***implies*** $\forall z\, f_{x,y}(z)\uparrow$ ***implies*** $f_{x,y} \notin NON\_TRIVIAL\_RANGE$

Thus, $<x,y> \in K_0 \Leftrightarrow f_{x,y} \in NON\_TRIVIAL\_RANGE$

And so, $K_0 \leq_m NON\_TRIVIAL\_RANGE$

4 **d.)** Show that **NON_TRIVIAL_RANGE** $\leq_m$ K_0.

Let f be an arbitrary index (function)
Define $g_f(z) = \exists <x,y,t> [STP(f,x,t) \&\& STP(f,y,t) \&\& (VALUE(f,x,t) \neq VALUE(f,y,t))]$

$f \in NON\_TRIVIAL\_RANGE$ ***implies*** $\forall z\, g_f(z) = 1$ ***implies*** $<gf,0> \in K_0$

$f \notin NON\_TRIVIAL\_RANGE$ ***implies*** $\forall z\, g_f(z)\uparrow$ ***implies*** $<gf,0> \notin K_0$

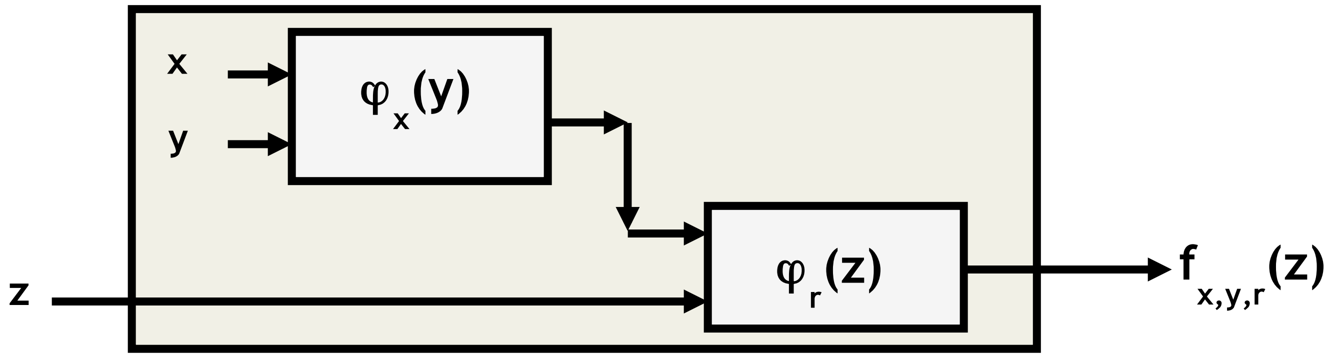
Thus, $f \in NON\_TRIVIAL\_RANGE \Leftrightarrow <gf,0> \in K_0$

And so, $NON\_TRIVIAL\_RANGE \leq_m K_0$

2 **e.)** From **a.)** through **d.)** what can you conclude about the complexity of **NON_TRIVIAL_RANGE (Recursive, RE, RE-COMPLETE, CO-RE, CO-RE-COMPLETE, NON-RE/NON-CO-RE)**?

RE-COMPLETE

5. Rice's Theorem deals with properties **P** of partial recursive functions and their corresponding sets of indices $\mathbf{S_P}$. The following image describing a function $\mathbf{f_{x,y,r}}$ that is central to understanding Rice's Theorem.



Given the hypotheses **P** is non-trivial and is an I/O behavior and that we assume, without loss of generality that all functions with empty domains/ranges do not have property **P**, explain the meaning of this diagram by doing the following:

2 **a.)** Indicate what **r** is, how it is chosen and how we can guarantee its existence.

r is some arbitrary member of the set S_P. As P is non-trivial S_P must be non-empty and so such an r must exist.

2 **b.)** Using recursive function notations, write down precisely what $\mathbf{f_{x,y,r}}$ computes for the Strong Form of Rice's Theorem.

$f_{x,y,r}(z) = \varphi_x(y) - \varphi_x(y) + \varphi_r(z)$

5 **c.)** Specify how the function $\mathbf{f_{x,y,r}}$ behaves with respect to **x**,**y** and **r**, and how this relates to the original problem, **P**, and set, $\mathbf{S_P}$.

If $\varphi_x(y)\uparrow$ then $\forall z\, f_{x,y,r}(z)\uparrow$ and so $f_{x,y,r}$ is in the complement of the set S_P

If $\varphi_x(y)\downarrow$ then $\forall z\, f_{x,y,r}(z) = \varphi_r(z)$ and so $f_{x,y,r}$ is in the set S_P

Combining these, we have that <x,y> is in K_0 iff $f_{x,y,r}$ is S_P

This means that a solution to membership in S_P implies a solution to membership in K_0.

As K_0 is the set associated with the Halting Problem, its membership is undecidable and thus so is membership in S_P. Hence the problem P is at least as hard as the Halt Problem and hence must also be undecidable.

6 **6.** Let **S** be an arbitrary non-recursive semi-decidable set. This means that **S** is the domain of some partial recursive function $\mathbf{f_s}$, whose domain is infinite. Using $\mathbf{f_S}$, show that **S** has an infinite recursive subset, call it **R**. To be complete you will need to create a characteristic function for **R**, $\chi_{\mathbf{R}}$, and argue that the set **R** you defined is infinite. **Hint**: Inductively define a monotonically increasing algorithm that enumerates **R**. I'll even do this part for you.

$\mathbf{f_R(0) = < \mu<x,t> [\ STP(f_S, x, t)\ ] >_1}$ **// Extract first component of <x, t>**
$\mathbf{f_R(y+1) =} < \mu<x,t> [\ STP(f_S, x, t)\ ]\ \&\&\ (x > f_R(y))\ ] >_1$ **// You fill this part in**
You now need to argue that $\mathbf{f_R}$ is total and monotonically increasing. From that you must argue that the set **R** enumerated by $\mathbf{f_R}$ is an infinite subset of **S** and then you must define the characteristic function $\chi_{\mathbf{R}}$ for **R**. I started the hardest part.

First, since the domain of f_s is infinite, it is non-empty. Thus, $f_R(0) = < \mu<x,t> [\ STP(f_S, x, t)\ ] >_1$ will converge and return some value in the domain of f_s. Thus, $f_R(0)$ converges and returns an element in S.
Assume that f_R (y) converges and returns a value in S greater than any value previously enumerated by f_R.
As the domain of f_s, is infinite, there is value in its domain that is larger than $f_R(y)$. Thus, our search at y+1 will always find an <x,t> such that $STP(f_S, x, t)\ ]\ \&\&\ (VALUE(f_S, x, t) > f_R(y))$ and that vale will be greater than $f_R(y)$ and also in S. This inductively shows that $f_R(y)$is monotonically increasing and enumerates a set R that is a subset of S.

$\chi_R(x) = \exists y \le x\ [\ f_R(y) == x\ ]$ decides membership in R.

3 **7.** We proved that $\mathbf{TOTAL = \{ f \mid \forall x\ \varphi_f(x)\downarrow \}}$ is not recursively enumerable. The proof is straightforward in that we assume the property to be so and that implies there is an algorithm **A** that enumerates the indices of all algorithms. Using the universal machine, φ, where $\boldsymbol{\varphi(f,x) = \varphi_f(x)}$, we have that $\boldsymbol{\varphi(A(f),x) = \varphi_{Af}(x)}$, that is, the value of the **f**-th algorithm at the input **x**. We then can define a new algorithm $\mathbf{D(x) = \varphi(A(x),x) + 1}$. Now you must finish the arguments that show that **D** contradicts its own existence and hence of the existence of the enumerating algorithm **A**.

As D is an algorithm its index must be enumerated by A. Assume then that D is the d-th algorithm enumerated by A and thus $D(x) = \varphi(A(d),x)$. Now, consider D(d), which must be defined since D is an algorithm.

D(d)	***=***	$\varphi(A(d),d) + 1$	***by definition of D***
	=	$D(d) + 1$	***since D is the d-th algorithm***

But then D(d) = D(d) + 1, which cannot be so if D converges on d, which it must.

The consequence is that D cannot exist, but then A cannot exist and hence TOTAL is not recursively enumerable, as assuming that to be so was the only non-constructive part of this reasoning.