COT 5310 Fall 2006 More Sample Final Exam Questions

Generally useful information.

- When I say a set **S** is re, unless I explicitly say otherwise, you may assume any of the following equivalent characterizations:
  1. **S** is either empty or the range of a total recursive function $\mathbf{f_S}$.
  2. **S** is the domain of a partial recursive function $\mathbf{g_S}$.
- If I say a function **g** is partially computable, then there is an index **g** (I know that's overloading, but that's okay as long as we understand each other), such that $\mathbf{\Phi_g(x) = \Phi(x, g) = g(x)}$. Here **Φ** is a universal partially recursive function.
  Moreover, there is a primitive recursive function **STP**, such that
  STP(x, g, t) is **1** (true), just in case **g**, started on **x**, halts in **t** or fewer steps.
  STP(x, g, t) is **0** (false), otherwise.
  Finally, there is another primitive recursive function **VALUE**, such that
  VALUE(x, g, t) is **g(x)**, whenever **STP(x, g, t)**.
  VALUE(x, g, t) is defined but meaningless if **~STP(x, g, t)**.
- The notation **f(x)↓** means that **f** converges when computing with input **x**, but we don't care about the value produced. In effect, this just means that **x** is in the domain of **f**.
- The notation **f(x)↑** means **f** diverges when computing with input **x**. In effect, this just means that **x** is **<u>not</u>** in the domain of **f**.
- The Post Correspondence problem is known to be undecidable. This problem is characterized by instances that are described by two n-ary sequences of non-empty words $\mathbf{<x_1,x_2,\ldots,x_n>, <y_1,y_2,\ldots,y_n>}$. The question is whether or not there exists a sequence, **i1,i2,…,ik**, such that $\mathbf{x_{i1}x_{i2}\cdots x_{ik} = y_{i1}y_{i2}\cdots y_{ik}}$
- The language **{ww | w** is a word in some alphabet with more than one letter**}** is not a **CFL**.
  The language $\mathbf{\{a^nb^nc^n \mid n\geq 0\}}$ is not a **CFL**. The language $\mathbf{\{a^nb^n \mid n\geq 0\}}$ is not **Regular**.
- When I ask for a reduction of one set of indices to another, the formal rule is that you must produce a function that takes an index of one function and produces the index of another having whatever property you require. However, I allow some laxness here. You can start with a function, given its index, and produce another function, knowing it will have a computable index. For example, given f, a unary function, I might define g_f, another unary function, by
  $g_f(0) = f(0)$
  $g_f(y+1) = g_f(y) + f(y+1)$
  This would get $g_f(x)$ as the sum of the values of f(0)+f(1)+…+f(x), which could be useful.

1. Choosing from among **(D) decidable**, **(U) undecidable**, **(?) unknown**, categorize each of the following decision problems. No proofs are required.

Problem / Language Class	Regular	Context Free	Context Sensitive
$L = \Sigma^*$?			
$L = \phi$?			
$L = L^2$?			
$x \in L^2$, for arbitrary x ?			

2. Choosing from among **(Y) yes**, **(N) No**, **(?) unknown**, categorize each of the following closure properties. No proofs are required.

Problem / Language Class	Regular	Context Free
Closed under intersection?		
Closed under quotient?		
Closed under quotient with Regular languages?		
Closed under complement?		

3. Consider the two operations on languages **max** and **min**, where
max(L) = { x | x ∈ L and, for no **y** does **xy ∈ L }** and
min(L) = { x | x ∈ L and, for no prefix of **x**, **y**, does **y ∈ L }**
Describe the languages produced by **max** and **min**. for each of the following:

$L_1 = \{ a^i b^j c^k \mid k \le i \text{ or } k \le j \}$

max(L_1) = ________________________________

min(L_1) = ________________________________

$L_2 = \{ a^i b^j c^k \mid k > i \text{ or } k > j \}$

max(L_2) = ________________________________

min(L_2) = ________________________________

4. Prove that any class of languages, ***C***, closed under union, concatenation, intersection with regular languages, homomorphism and substitution (e.g., the Context-Free Languages) is closed under **MissingMiddle**, where, assuming L is over the alphabet Σ,
MissingMiddle(L) = { xz | ∃y ∈ Σ* such that xyz ∈ L }
You must be very explicit, describing what is produced by each transformation you apply.

5. Use **PCP** to show the undecidability of the problem to determine if the intersection of two context free languages is non-empty. That is, show how to create two grammars $\mathbf{G_A}$ and $\mathbf{G_B}$ based on some instance $\mathbf{P = \langle\langle x_1,x_2,\ldots,x_n\rangle, \langle y_1,y_2,\ldots,y_n\rangle\rangle}$ of **PCP**, such that $\mathbf{L(G_A) \cap L(G_B)} \neq \phi$ iff **P** has a solution. Assume that **P** is over the alphabet **Σ**.You should discuss what languages your grammars produce and why this is relevant, but no formal proof is required.

6. Consider the set of indices **CONSTANT** = $\{ \mathbf{f} \mid \exists \mathbf{K}\ \forall \mathbf{y}\ [\ \varphi_{\mathbf{f}}(\mathbf{y}) = \mathbf{K}\ ]\ \}$. Use Rice's Theorem to show that **CONSTANT** is not recursive. Hint: There are two properties that must be demonstrated.

7. Show that **CONSTANT** $\equiv_{\mathbf{m}}$ **TOT**, where **TOT** = $\{ \mathbf{f} \mid \forall \mathbf{y}\ \varphi_{\mathbf{f}}(\mathbf{y})\downarrow \}$.

8. Why does Rice's Theorem have nothing to say about each of the following? Explain by showing some condition of Rice's Theorem that is not met by the stated property.

a.) **AT-LEAST-LINEAR = { f | $\forall$y φ_f(y) converges in no fewer than y steps }**.

b.) **HAS-IMPOSTER = { f | $\exists$ g [g$\neq$f** and **$\forall$y [φ_g(y) = φ_f(y)]] }**.

9. The trace language of a computational device like a Turing Machine is a language of the form **Trace = { $C_1\#C_2\# \ldots C_n\#$ | $C_i \Rightarrow C_{i+1}$, $1 \le i < n$ }**

Trace is Context Sensitive, non-Context Free. Actually, a trace language typically has every other configuration word reversed, but the concept is the same. Oddly, the complement of such a trace is Context Free. Explain what makes its complement a **CFL**. In other words, describe the characteristics of this complement and why these characteristics are amenable to a **CFG** description. **Note**: Reversing the second word in a pair is important here if you're thinking about Turing Machines but is irrelevant for **FRS with Residue**. Thus, don't make reversal an issue in your discussion. Also, here's a start for you, assuming **Σ** is the alphabet of configuration words.

Let **R** = $(\Sigma + \#)^*\#\#(\Sigma + \#)^* + \Sigma^* + \#(\Sigma + \#)^* + (\Sigma + \#)^*\Sigma^+$

R is a regular expression that describes all words that do not look like sequences of configurations. Your job now is to describe **BadTrace**, the rest of the complement of **Trace** and discuss why it's a **CFL**.