COT 6410 Spring2019 More Midterm Sample Exam Questions

1. In each case below, consider **R1** and **R2** to be Regular and **L1** and **L2** to be non-regular CFLs. Fill in the three columns with **Y** or **N**, indicating what kind of language **L** can be. No proofs are required.
Read ⊆ as "is contained in and may equal." Put **Y** in all that are possible and **N** in all that are not. I did one example, so you get the idea.

Definition of L	Regular?	CFL, non-Regular?	Not even a CFL?
L = L1 ∪ L2	*Y*	*Y*	*N*
L = R1 ∩ R2	*Y*	*N*	*N*
L = L1 – R2	*Y*	*Y*	*N*
L = L1 / L2 *	*Y*	*Y*	*Y*
L ⊆ R1 / R2	*Y*	*Y*	*Y*

* ***L1** = $\{ a^nb^n \mid n > 0 \}$ **L2** = $\{ b^na^n \mid n > 0 \}$ **L1/L2** = { } (Regular)*
***L1** = $\{ a^nb^{n+1}c^*d^* \mid n > 0 \}$ **L2** = $\{ bc^nd^n \mid n > 0 \}$ **L1/L2** = $\{ a^nb^n \mid n > 0 \}$ (CFL)*
***L1** = $\{X_0 \$ \# X_0 \# X_1 \# X_2 \# X_3 \# X_4 \# \ldots \# X_{2k-1} \# X_{2k} \# Z \#\}$ where $X_{2i-1} \Rightarrow X_{2i}$, $1 \le i \le k$*
***L2** = $\{ \$ \# Y_0 \# Y_1 \# Y_2 \# Y_3 \# \ldots \# Y_{2j} \# Y_{2j+1} \# \}$ where $Y_{2i} \Rightarrow Y_{2i+1}$, $0 \le i \le j$,*
X_0 is a legitimate starting configuration and Z is a unique halting configuration.
L1/L2 = (X_0 | X_0 is a configuration for which the machine being traced halts. (RE)

2. 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 = Σ* ?	*D*	*U*	*U*
L = ϕ ?	*D*	*D*	*U*
L = L^2 ?	*D*	*U*	*U*

3. 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
Mid Loss with Regular Sets, where **L** ∊ ***C***, **R** is Regular, **L** and **R** are over the alphabet **Σ**, and
L | R = { xz | ∃y ∊ R, such that **xyz ∊ L }.**
You may assume substitution **f(a) = {a, a'},** and homomorphisms **g(a) = a'** and **h(a) = a, h(a') = λ**. Here **a∊Σ** and **a'** is a new character associated with each such **a∊Σ**.
You only need give me the definition of **L | R** in an expression that obeys CFL closure properties; you do not need to prove or even justify your expression.

L | R = ***h(f(L) ∩ Σ* g(R) Σ*)***

5. Specify True (**T**) or False (**F**) for each statement.

Statement	**T or F**
Every Regular Language is also a Context Free Language	***T***
The Context Free Languages are closed under Complement	***F***
The Quotient of a Context Free and Regular Language is Context Free	***T***
An algorithm exists to determine if a Context Free Language is infinite	***T***
Phrase Structured Languages are the same as RE Languages	***T***
The Quotient of two Context Free Languages is Context Free	***F***
The Ambiguity problem for Context Free Languages is decidable	***F***
There is an algorithm to determine if **L**=**Σ***, for **L** a Context Free Language	***F***
An algorithm exists to determine if a Context Sensitive Language is infinite	***F***
PCP is undecidable even for one-letter systems	***F***
Membership in Context Sensitive Languages is undecidable	***F***
~~Every RE language is Turing reducible to its complement~~	~~***T***~~
Emptiness is undecidable for Context Sensitive Languages	***T***
There is an algorithm to determine if $\mathbf{L}=\mathbf{L}^2$, for **L** a Regular Language	***T***
The complement of a trace language is Context Free	***T***
The word problem for two-letter Semi-Thue Systems is decidable	***F***

6. Let $\mathbf{P} = \langle\langle x_1,x_2,\ldots,x_n\rangle, \langle y_1,y_2,\ldots,y_n\rangle\rangle$, $x_i,y_1 \in \Sigma^+$, $1 \le i \le n$, be an arbitrary instance of **PCP**. We can use **PCP's undecidability** to show the undecidability of the problem to determine if a **Context Sensitive Grammar** generates a non-empty language. I will present the grammar, **G**. You must explain how it maps an instance of PCP to the non-emptiness problem for this $\mathcal{L}$**(G)**.

Define **G = ({S, T}** $\cup$ **Σ, {*}, S, R)**, where **R** is the set of rules:

$S \rightarrow x_i\ S\ y_i^R \mid x_i\ T\ y_i^R \quad 1 \le i \le n$ **(Note: the superscripted R means Reversed)**

$a\ T\ a \rightarrow *\ T\ *$

$*\ a \rightarrow a\ *$

$a\ * \rightarrow *\ a$

$T \rightarrow *$

a) What are the syntactic forms (strings with a variable in them) generated from **S** at the time it is rewritten as a string with a **T** in it.

$x_{i1} \ldots x_{ik}\ T\ y_{ik}^R \ldots y_{i1}^R$ ***where*** $1 \le ij \le n$ ***for*** $1 \le j \le k$

b) What do the terminal strings look like when and if any are produced and under what circumstances are such terminal string produced? When answering this question, you should be referring back to part (a).

$*^{2n+1}$ ***where*** $n = | x_{i1} \ldots x_{ik} |$ ***and*** $x_{i1} \ldots x_{ik} = y_{i1} \ldots y_{ik}$

c) There are two possible cardinalities (sizes) of the language, $\mathcal{L}$**(G)**. What are these?

$|\ \mathcal{L}(G)\ |$ ***is either 0 or countably infinite***