6 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 "contains and may equal."
Put **Y** in all that are possible and **N** in all that are not.

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

3 2. Choosing from among **(D) decidable**, **(U) undecidable**, categorize each of the following decision problems. No proofs are required. **L** is a language over Σ; **w** is a word in Σ*

Problem / Language Class	Regular	Context Free	Context Sensitive	Phrase Structured
w ∈ L ?	*D*	*D*	*D*	*U*
L is infinite ?	*D*	*D*	*U*	*U*

4 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 **Double Interior Loss with Regular Sets**, denoted by the operator ‖, where **L ∈ C**, **R** is Regular, **L** and **R** are both over the alphabet Σ, and
L‖R = { uwy | ∃v,x ∈ R, such that **uvwxy ∈ 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 the above closure properties; you do not need to prove or even justify your expression.

L‖R = _____*h(f(L) ∩ Σ* g(R) Σ* g(R) Σ*)*_____

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

Statement	T or F
Rice's Theorem demonstrates the undecidability of the Halting Problem	*F*
The Context Free Languages are closed under intersection	*F*
The Ambiguity problem for Context Free Languages is undecidable	*T*
The Quotient of two Context Free Languages is Context Sensitive	*F*
An algorithm exists to determine if a Context Free Language is Σ*	*F*
Every RE set can be generated by a Phrase Structured Grammar	*T*
The set difference of two Context Free Languages is Context Sensitive	*T*
There is an algorithm to determine if **L = ∅**, for **L** a Context Sensitive Language	*F*

4   **5.** Let $\mathbf{P} = <<\mathbf{x_1,x_2,\ldots,x_n}>, <\mathbf{y_1,y_2,\ldots,y_n}>>$, $\mathbf{x_i,y_1} \in \Sigma^+$, $\mathbf{1{\le}i{\le}n}$, be an arbitrary instance of **PCP**. We can use **PCP's undecidability** to show the undecidability of the problem to determine if a **Context Free Grammar** is ambiguous. Present the grammar, **G**, associated with an arbitrary instance of PCP, **P**, such that $\mathcal{L}$**(G)** is ambiguous if and only if there is a solution to **P**.

Define $\mathbf{G = (\{ S, X, Y \}, \Sigma \cup \{ [i] \mid 1 \le i \le n \}, S, R)}$, where **R** is the set of rules:

$S \rightarrow X \mid Y$
$X \rightarrow x_i\, X\, [i] \mid X \rightarrow x_i\, [i] \qquad 1 \le i \le n$
$Y \rightarrow y_i\, Y\, [i] \mid Y \rightarrow y_i\, [i] \qquad 1 \le i \le n$

12   **6.** 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 | domain(f) = range(f) }**   *NRNC*

     Justification: *$\forall <x, t> \exists <y, s> [STP(f, x, t) \Leftrightarrow (STP(f, y, s) \,\&\&\, VALUE(f, y, s) = x)]$*

b.) **{ <f,x> | f(x) = x }**   *RE*

     Justification: *$\exists t [STP (f, x, t) \,\&\&\, VALUE(f, x, t) = x)]$*

c.) **{ f | f(x) converges in x steps for at least one value of x }**   *RE*

     Justification: *$\exists x [STP (f, x, x)]$*

d.) **{ f | whenever f converges, f(x) = x }**   *coRE*

     Justification: *$\forall <x, t> [STP(f, x, t) \Rightarrow (VALUE(f, x, t) = x)]$*

2   **7.** Looking back at Question 6, which of these are candidates for using Rice's Theorem to show their unsolvability? Check all for which Rice Theorem might apply.

     a) *X*   **b)** *X*   **c)** _____   **d)** *X*

6   **8.** Let **S** be an arbitrary, non-empty re/semi-decidable set. One definition is that **S** is the range of some total recursive f_S. Using f_S, show that **S** is the domain of some partial recursive function g_S. Here the function g_S that you define based on the existence of f_S semi-decides **S**.

$$g_S(x) = \exists y \, [\, f_S(y) = x \,]$$

Let **S** be an arbitrary, non-empty re/semi-decidable set. One definition is that **S** is the domain of some partial recursive function g_S. Using g_S and the fact that **S** is non-empty (you may assume **c** is some element guaranteed to be in **S**), show that **S** is the range of some total recursive f_S. Here the function f_S that you define based on the existence of g_S enumerates the elements of **S**. Hint: Each element of **S** is enumerated a countably infinite number of times by your function f_S.

$$f_S(\langle x,t \rangle) = x * STP(g_S, x, t) + c * (1\text{-}STP(g_S, x, t))$$

6   **9.** Show example sets **A** and **B**, where **A** is non-empty and recursive, and **B** is re non-recursive and

     a.) **Max(A,B)** = { z | z = max(x,y) where x ∈ **A** and y ∈ **B** } is recursive

         A = ℵ, B = K, Max(A,B) = ℵ-{ x | x < least value in K} = { x | x ≥ least value in K}

     b.) **Max(A,B)** = { z | z = max(x,y) where x ∈ **A** and y ∈ **B** } is re non-recursive

         A = {0}, B = K, Max(A,B) = K

     Hint: Consider **B** = **K** = { f | $\varphi_f(f) \downarrow$ }

     Note: You must specify the results of **Max(A,B)** for each case above.

10. Define **SuccessorLike (SL)** = (**f** | for some input **x, f(x) = x+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.)

$\exists <x,t> [STP(f, x, t)$ && $(VALUE(f, x, t) = x+1)]$

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

$S(x) = x+1 \in SL$; $I(x) = x \notin SL$ // SL is non-trivial

Let f and g be two arbitrary function indices such that $\forall x [f(x) = g(x)]$.

$f \in SL \Leftrightarrow \exists x\ f(x) = x+1 \Rightarrow$ *for some* $x_0, f(x_0) = x_0+1 \Rightarrow g(x_0) = x_0+1 \Rightarrow \exists x\ g(x) = x+1 \Rightarrow g \in SL$

$f \notin SL \Leftrightarrow$ *for no x does* $f(x) = x+1 \Leftrightarrow$ *for no x does* $g(x) = x+1 \Leftrightarrow g \notin SL$ *// Can just do this one*

4 **c.)** Show that $K \leq_m SL$, where $K = \{ f | f(f)\downarrow \}$.

Let f be an arbitrary function index. Define $\forall x\ F_f(x) = f(f) - f(f) + x + 1$

$f \in K \Leftrightarrow f(f)\downarrow \Leftrightarrow \forall x\ F_f(x) = x + 1 \Rightarrow F_f \in SL$

$f \notin K \Leftrightarrow f(f)\uparrow \Leftrightarrow \forall x\ F_f(x)\uparrow \Rightarrow F_f \notin SL$

4 **d.)** Show that $SL \leq_m K$.

Let f be an arbitrary function index.
Define $\forall y\ F_f(y) = \exists <x,t> [STP(f, x, t)$ && $(VALUE(f, x, t) = x+1)]$

$f \in SL \Leftrightarrow \exists <x,t> [STP(f, x, t)$ && $(VALUE(f, x, t) = x+1)] \Leftrightarrow \forall y\ F_f(y)\downarrow \Rightarrow F_f \in K$

$f \notin SL \Leftrightarrow \sim \exists <x,t> [STP(f, x, t)$ && $(VALUE(f, x, t) = x+1)] \Leftrightarrow \forall y\ F_f(y)\uparrow \Rightarrow F_f \notin K$

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

RE-COMPLETE