COT 6410 Spring 2020 Midterm#1 Name: KEY

Raw Score ___ / 60 Grade:_________

6 **1.** In each case below, consider **R1** to be Regular, **R2** to be finite, 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 "contained in 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 / L2	*Y*	*Y*	*Y*
L = L1 – R1	*Y*	*Y*	*N*
L = Σ* – L1	*N*	*Y*	*Y*
L ⊆ R2	*Y*	*N*	*N*

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**
L = ∅ ?	*D*	*D*	*U*	*U*
L is Σ* ?	*D*	*U*	*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 Retention 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 = { vx | v,x ∊ R and ∃u,w ∊ Σ^+ such that **uvwx ∊ 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) ∩ $g(\Sigma^+)$ R $g(\Sigma^+)$ R*

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

Statement	**T or F**
An algorithm exists to determine if a Phrase Structured Grammar generates λ	*F*
If **P** is Unsolvable then Rice's Theorem can always show this	*F*
The Context Sensitive Languages are closed under complement	*T*
If **P** $\leq_m$ **Halt** then **P** must be RE	*T*
The **RE** sets are closed under intersection	*T*
The correct traces of a Turing Machine's Computations form a Context Free Language	*F*
The Post Correspondence Problem is decidable if |**Σ**| = **1**	*T*
There is an algorithm to determine if **L** is finite, for **L** a Context Sensitive Language	*F*

4 **5.** Let $\mathbf{P} = \langle\langle x_1,x_2,\ldots,x_n\rangle, \langle y_1,y_2,\ldots,y_n\rangle\rangle$, $x_i,y_1 \in \Sigma^+$, $1\leq i\leq n$, 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 grammars, **G1** and **G2**, associated with an arbitrary instance of PCP, **P**, such that $\mathcal{L}(\mathbf{G1}) \cap \mathcal{L}(\mathbf{G2})$ is non-empty if and only if there is a solution to **P**.

Define $\mathbf{G1} = (\{X\}, \Sigma \cup \{ [i] \mid 1 \leq i \leq n \}, R1, X)$, $\mathbf{G2} = (\{Y\}, \Sigma \cup \{ [i] \mid 1 \leq i \leq n \}, R1, Y)$, where **R1 and R2** are the sets of rules (this is your job):

$X \rightarrow x_i X [i] \mid x_i [i] \qquad 1 \leq i \leq n$

$Y \rightarrow y_i Y [i] \mid y_i [i] \qquad 1 \leq i \leq n$

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.) A = { f | range(φ_f) has no values greater than 10 }

$\forall \langle x, t\rangle [STP(f, x, t) \Rightarrow VALUE(f, x, t) \leq 10]$ — *CoRE*

b.) B = { <f, x> | φ_f converges on every value (input) greater than or equal to x}

$\forall y\, \exists t\, [y \geq x \Rightarrow STP(f, y, t)]$ — *NRNC*

c.) C = { f | φ_f converges for at least one value (input) of x in at most x steps }

$\exists x\, [STP(f, x, x)]$ — *RE*

d.) D = { f | if φ_f(f) converges it takes more than f steps to do so }

$\sim STP(f, f, f)$ — *REC*

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)** ____

8. Show that a set **S** is an infinite decidable (solvable/recursive) set if and only if it can be described as the range of a monotonically increasing algorithm. I will start the proof.

3 **a.)** Let **S** be an infinite recursive set. As **S** is decidable, it has a characteristic function χ_S where $\chi_S(x) = 1$, when $x \in S$, and $\chi_S(x) = 0$, otherwise. Using χ_S as a basis, we wish to define a monotonically increasing algorithm f_S whose range is **S**. Note that, since **S** is non-empty, it has a smallest element and, since it is infinite, it has no largest element. I have started the proof using primitive recursion (induction). You must complete it by writing in the formula to compute $f_S(y+1)$ given we know the value of $f_S(y)$.

Let $x \in S \Leftrightarrow \chi_S(x)$
// list the smallest element
Define $f_S(0) = \mu\, x\, \chi_S(x)$
// list the next item in monotonically increasing order. That's your job!!
$f_S(y+1) =$ ___ *$\mu\, x\, [\, \chi_S(x)\ \&\&\ x > f_S(y)\,]$* ___

3 **b.)** Let **S** be the range of some monotonically increasing enumerating algorithm f_S. Show that **S** must be an infinite recursive set. First **S** is infinite since $\forall x\ f_S(x+1) > f_S(x)$. You must now present a characteristic function χ_S that takes advantage of the infinite nature of **S** and the fact that f_S is monotonically increasing and so enumerates any item **x** in some known bounded amount of time.

$\chi_S(x) =$ ___ *$\exists y \le x\, [\, f_S(y) = x\,]$* ___

6 **9**. Let sets **A** be a **non-empty** recursive (decidable) set and let **B** be re non-recursive (undecidable). Consider $C = \{ z \mid z = y^x$, where $x \in A$ and $y \in B \}$.
Note: Here, we define 0^0 to be **1** (yeah, I know that's a point of debate in Mathematics, but not in this question). For (a)-(c), either show sets **A** and **B** and the resulting set **C**, such that **C** has the specified property (argue convincingly that it has the correct property) or demonstrate (prove by construction) that this property cannot hold.

a. Can **C** be recursive? Circle ***Y*** or **N.**
$A = \{ 0 \}$; $B = HALT$; $C = \{ x^0 \mid x \in HALT \} = \{ 1 \}$, which is recursive

b. Can **C** be re non-recursive? Circle ***Y*** or **N.**
$A = \{ 1 \}$; $B = HALT$; $C = \{ x^1 \mid x \in HALT \} = HALT$, which is re, non-recursive

c. Can **C** be non-re? Circle **Y** or ***N*.**

Let Range (f_A) = A; Range (f_B) = B
Define $f_C(x, y) = f_B(x) \wedge f_C(y) = \{ x^y \mid x \in A, y \in B \} = C$
But then C is enumerated by f_C and hence is re.

10. Define **CounterID (CI) = (f |** for all input **x**, **f(x)↓ & f(x) ≠ x** }.

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.)

$\forall x\ \exists t\ [\ STP(f, x, t\ )\ \&\&\ VALUE(f, x, t\ ) \neq x\ \}$

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

Non-Trivial as: $S \in CI$ and $C0 \notin CI$

Let f, g be arbitrary indices of procedures such that $\forall x\ f(x) = g(x)$

$f \in CI \quad \Leftrightarrow \quad \forall x\ f(x)\downarrow \&\&\ f(x) \neq x$

$\quad \Leftrightarrow \quad \forall x\ g(x)\downarrow \&\&\ g(x) \neq x \qquad$ ***since*** $\forall x\ f(x) = g(x)$

$\quad \Leftrightarrow \quad g \in CI$

Thus, using the Strong Form of Rice's we have that CI is undecidable.

5 **c.)** Show that **TOTAL $\leq_m$ CI**, where **TOTAL = { f | ∀x f(x)↓ }**.

Let f be an arbitrary index of a procedure

Define $\forall x\ G_f(x) = f(x) - f(x) + x + 1$

$f \in TOTAL \quad \Leftrightarrow \quad \forall x\ f(x)\downarrow$

$\quad \Leftrightarrow \quad \forall x\ G_f(x) = x+1$

$\quad \Rightarrow \quad G_f \in CI$

$f \notin TOTAL \quad \Leftrightarrow \quad \exists x\ f(x)\uparrow$

$\quad \Rightarrow \quad G_f \notin CI$

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

NON-RE/NON-CO-RE