COT 4210 Fall 2017 Midterm 2 Name: ________________
Total Points Available <u>72</u> Your Raw Score _______ Grade: M2:____ Combined:_____

4 **1.** Write a Context Free Grammar for the language **L**, where
L = { $a^i b^j c^k$ | i, k > 0 , j = i + k }, that is **#b's = #a's + #c's**

8 **2.** Use the Pumping Lemma for CFLs to show that the following language **L** is not Context Free.
L = { $a^i b^j c^k$ | i, k >0, j = i * k }, that is **#b's = #a's * #c's**
Be explicit as to why each case you analyze fails to be in **L**. I will do the first two steps.

Me: Assume L is Context Free
PL: Provides a whole number N>0 that is the value associated with L based on the Pumping Lemma for CFLs

12 **3.** Consider some language **L**. For each of (a) and (b), and for each of the three possible complexities of **L**, indicate whether this is possible (**Y** or **N**) and present evidence, if you answered **Y**, by providing an example language **A**, with a homomorphism **σ** for **(b)**, and the resulting **L**, or, if you answered **N**, state some known closure property that reflects a bound on the complexity of **L**.

a.) **L = min(A)** where **A** is context-free, not regular.
Can **L** be Regular**?** Circle **Y** or **N.**
If yes, show **A** and argue **min(A)** is Regular; if no, why not?

Can **L** be a non-regular CFL? Circle **Y** or **N.**
If yes, show **A** and argue **min(A)** is a CFL; if no, why not?

Can **L** be more complex that a CFL? Circle **Y** or **N.**
If yes, show **A** and argue **min(A)** is not a CFL; if no, why not?

b.) Let **σ** **b**e a homomorphism from **Σ** into regular languages, such that, for each **a∈Σ**, **σ(a) =** $\mathbf{w_a}$, for some string $\mathbf{w_a}$. Let **A** be a context free, non-regular language and let **L = σ(A).**
Can **L** be Regular? Circle **Y** or **N**.
If yes, show **A** and **σ**, and argue **σ(A)** is Regular; if no, why not?

Can **L** be a non-regular CFL? Circle **Y** or **N.**
If yes, show **A** and **σ**, and argue **σ(A)** is a CFL; if no, why not?

Can **L** be more complex that a CFL? Circle **Y** or **N.**
If yes, show **A** and **σ**, and argue **σ(A)** is not a CFL; if no, why not?

10 **4.** Present the **CKY** recognition matrix for the string **a-(a)+a** assuming the Chomsky Normal Form grammar, **G = ({ E, L, P, R, T, Y, Z }, { a, +, -, (,) }, R, E)**, specified by the rules **R**:.

E → a | EY | LZ | PE
Z → ER
Y → PT
T → a | LZ
P → + | -
L → (
R →)
Note: Do not be surprised if many of the cells are blank when you complete this.

	a	**-**	**(**	**a**	**)**	**+**	**a**
1							
2							
3							
4							
5							
6							
7							

A

A little help

Non-Terminal	**First Symbol in Rules**	**Second Symbol in Rules**
E	**E → EY; Z → ER**	**E → PE**
Z	**None**	**E → LZ; T → LZ**
Y	**None**	**E → EY**
T	**None**	**Y → PT**
P	**E → PE; Y → PT**	**None**
L	**E → LZ; T → LZ**	**None**
R	None	**Z → ER**

Is **a-(a)+a** in **L(G)?** ____________

What is the order of execution of this approach to determine if some **w**, **|w| = N**, is in **L?** _________

What is the algorithmic strategy, e.g., greedy, divide and conquer, dynamic programming, backtracking or randomized, associated with this **CKY** algorithm called? ___________________

8 **5**. Prove that **Context-Free Languages** are closed under **double quotient with Regular Languages**. Double Quotient of a CFL **L** and a Regular Language **R**, both of which are over the alphabet **Σ**, is denoted **DQ(L, R)**), and defined as **DQ(L, R) = { x | xyz ∈ L and y, z ∈ R }.**
That is, we select a prefix **x** of **xyz** in **L**, provided **y** and **z** are both in the Regular Language **R**.
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 distinct new character associated with each **a∈Σ**.

DQ(L, R) = ____

You must now be very explicit as to why each language you produce by your various transformations (substitutions and other operations) is a CFL or a Regular language. Do this for **all** parts of the expression you write, with **appropriate justifications** based on known closure properties. You do **not** need to describe what is produced by each such transformation

5 **6.** Let **L1**, **L2** be Non-Regular CFLs; **R1**, **R2** be Regular; Answer is about **S** and there should be just one cell per row that has a check mark, or **X** if your prefer.

Definition of S / Characterization of S	***Always Regular***	***At worst CFL***	***Might not be CFL***
S ⊆ L1 ∪ L2			
S = L1 ∩ R1			
S = R1 – R2			
S ⊇ R1			
S = L1 ∪ R1			

12 **7.** Consider the CFG **G = ({ S, A, B } , { a , b } , R , S)** where **R** is:
S → SAB | BA
A → AB | a
B → bS | b
In the PDAs below, you may (and are encouraged) to use a transition diagram where transitions have arcs with labels of form **a, α → β** where **a** ∈ $\Sigma\cup\{\lambda\}$, **α, β** ∈ **Γ***. Note: This just means that you can use extended stack operations that push or pop arbitrary length strings.

a.) Present a pushdown automaton that parses the language **L(G)** using a top down strategy.
INITIAL CONTENTS OF STACK = ______

b.) Now, using the notation of **ID**s (Instantaneous Descriptions, **[q, x, z]**), describe how your PDA in (a) accepts strings generated by **G**.

c.) Present a pushdown automaton that parses the language **L(G)** using a bottom up strategy. Note: I am fine with your showing strings that are on top of the stack in either reversed or non-reversed form.
INITIAL CONTENTS OF STACK = ______

d) Now, using the notation of **ID**s (Instantaneous Descriptions, **[q, x, z]**), describe how your PDA in (c) accepts strings generated by **G**.

8. Consider the context-free grammar **G = ({ S, A, B, C, D, E }, { 0, 1 }, R, S)**, where **R** is:

S → A | B
A → 1CD
B → 1CE
C → 1C1 | 01 | λ
D → 0C
E → 0B | A

3 **a.)** Remove λ-rules, creating an equivalent grammar **G'**. Show **all** rules.
Nullable = { }

3 **b.)** Remove all **unit** rules, creating an equivalent grammar **G''**. Show **all** rules.
Unit(S) = { }; Unit(A) = { }; Unit(B) = { };
Unit(C) = { }; Unit(D) = { }; Unit(E) = { }

2 **c.)** Remove all unproductive symbols, creating an equivalent grammar **G'''**. Show **all** rules.
Productive = { }; Unproductive = { }

2 **d.)** Remove all unreachable symbols, creating an equivalent grammar $\mathbf{G^{iv}}$. Show **all** rules.
Unreachable = { }

3 **e.)** Convert grammar $\mathbf{G^{iv}}$ to its **Chomsky Normal Form** equivalent, $\mathbf{G^{V}}$. Show **all** rules.