COT 4210 Fall 2018 Midterm 2 Name: KEY

Total Points Available 68 **Your Raw Score** ______ **Grade: M2:** ______ **Combined:** ______

5 4 **1.** Write a Context Free Grammar for the language **L**, where
L = { $a^i b^j c^k$ | k = (i – j), if i ≥ j, else k= 0 }. Hint: Splitting into two cases makes your job easier.

$S \rightarrow A \mid C$
$A \rightarrow aAc \mid B$ } $i \geq j$
$B \rightarrow aBb \mid \lambda$
$C \rightarrow aCb \mid Cb \mid b$ } $j > i$

3 **2.** 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 an **X**.

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

2 **3.** Which of the following are correct definitions of an ambiguous grammar? In each case, **w** is a terminal string. Write **T**(true) or **F**(false) in the underlined area following each statement

There are two distinct parse trees for some string **w** derived by the grammar T

There are two distinct derivations of some string **w** derived by the grammar F

There are two distinct rightmost derivations of some string **w** derived by the grammar T

There are two distinct leftmost derivations of some string **w** derived by the grammar T

4 **4.** Show that **Context-Free Languages** are closed under **Non-Empty-Left-Right Quotient with Regular Languages**. Non-Empty-Left-Right Quotient of a CFL **L** and a Regular Language **R**, both of which are over the alphabet **Σ**, is denoted **NELRQ(L, R)**), and defined as
NELRQ(L, R) = { y | xyz ∈ L; x, z ∈ R; and **y ≠ λ }.**
That is, we select a non-empty substring **y** of **xyz** in **L**, provided **x** 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**. No justification is required.

NELRQ(L, R) = $h(f(L) \cap g(R)\Sigma^{+}g(R))$

10 **5.** 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. Recall that
max(A) = { w | w ∈ A and for no x≠λ does wx ∈ A }
If you answer **Y**, you must provide an example language **A** and the resulting **L**. In the case of part (b) you must also present a homomorphism **σ**. If you answer **N**, state some known closure property that reflects a bound on the complexity of **L**. Note: I did the first of each of the three parts for you.

a.) **L = max(A)** where **A** is context-free, not regular.
Can **L** be Regular? Circle **Y** or **N.**
If yes, show **A** and argue **max(A)** is Regular; if no, why not?
YES. Let A = { $a^i b^j$ | i, j > 0 and j >i }
L = max(A) = ∅, a regular set, as every string in A can be extended with more b's.

Can **L** be a non-regular CFL? Circle **Y** or **N.**
If yes, show **A** and argue **max(A)** is a CFL; if no, why not?
$A = \{a^n b^n \mid n>0\}$
$L = max(A) = A = \{a^n b^n \mid n>0\}$

Can **L** be more complex that a CFL? Circle **Y** or **N.**
If yes, show **A** and argue **max(A)** is not a CFL; if no, why not?
$A = \{a^i b^j c^k \mid k \le i \text{ OR } k \le j\}$
$L = max(A) = \{a^i b^j c^k \mid k = max(i, j)\}$
L IS KNOWN (PROVEN) TO BE A CSL, NON-CFL LANGUAGE

b.) Let **σ** be a homomorphism from **Σ** into regular languages, such that, for each **a∈Σ**, **σ(a) = w_a**, for some string **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?
YES. Let A = { $a^n b^n$ | n > 0 }
Define σ(a) = λ and σ(b) = λ
L = σ (A) = { λ }, a regular set.

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?
LET $A = \{a^n b^n \mid n>0\}$
DEFINE $\sigma(a) = a$ AND $\sigma(b) = b$
$L = \sigma(A) = A = \{a^n b^n \mid n>0\}$ WHICH IS A CFL

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?
CFLS ARE KNOWN TO BE CLOSED UNDER SUBSTITUTION AND HOMOMORPHISM

5. Use the Pumping Lemma for CFLs to show that the following language **L** is not Context Free.
L = { a^n $b^{sum(1..n)}$ | n>0 }. Here **sum(1..n)** = $\sum_1^n i$.
Be explicit as to why each case you analyze fails to be an instance of **L** and, of course, make sure your cases cover all possible circumstances. I have done the first two steps for you.

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

CHOOSE $z = a^N b^{(N+1)N/2} \in L$

$\therefore z = uvwxy$, $|vwx| \le N$, $|vx| > 0$ AND $\forall i \ge 0$ $uv^iwx^iy \in L$

∃:

CASE 1: vwx CONTAINS ONLY b'S.
SET i = 2 THEN PL SAYS
$a^N b^{(N+1)N/2 + |vx|} \in L$
BUT $|vx| > 0$ AND SO WE HAVE TOO MANY b'S, THUS $uv^2wx^2y \notin L$
NOTE: CAN ALSO USE i = 0

CASE 2: vx CONTAINS AT LEAST ONE a
THEN vwx CONTAINS AT MOST N−1 b'S. SET i = 2
IN BEST CASE, WE HAVE N+1 a'S
AND $(N+1)N/2 + (N-1)$ b'S
BUT N+1 a'S REQUIRES $(N+1)N/2 + (N+1)$
b'S AND $(N+1)N/2 + (N-1) < (N+1)N/2 + (N+1)$
SO $uv^2wx^2y \notin L$

AS CASES 1 AND 2 COVER ALL POSSIBILITIES
WE HAVE A CONTRACTION SHOWING $uv^2wx^2y \notin L$
IN ALL CASES. HENCE L IS NOT A CFL

10 **7.** Present the **CKY** recognition matrix for the string **aababb** assuming the Chomsky Normal Form grammar, **G = ({ S, T, U, V, W, A, B }, { a, b }, R, S)**, specified by the rules **R**:.

S →	**A T | B U**
T →	**b | B S | A V**
U →	**a | A S | B W**
V →	**T T**
W →	**U U**
A →	**a**
B →	**b**

	a	**a**	**b**	**a**	**b**	**b**
1	UA	UA	TB	UA	TB	TB
2	W	S	S	S	V	
3	U	U	T	T		
4	W	S	V			
5	U	T				
6	S					

A little help

Non-Terminal	First Symbol in Rules	Second Symbol in Rules
S	None	T → B S ; U → A S
T	V → T T	S → A T ; V → T T
U	W → U U	S → B U ; W → U U
V	None	T → A V
W	None	U → B W
A	S → A T ; T → A V ; U → A S	None
B	S → B U ; T → B S ; U → B W	None

Is **aababb** in **L(G)**? Y

What is the order of execution of this approach to determine if some **w**, **|w| = N**, is in **L**? N^3

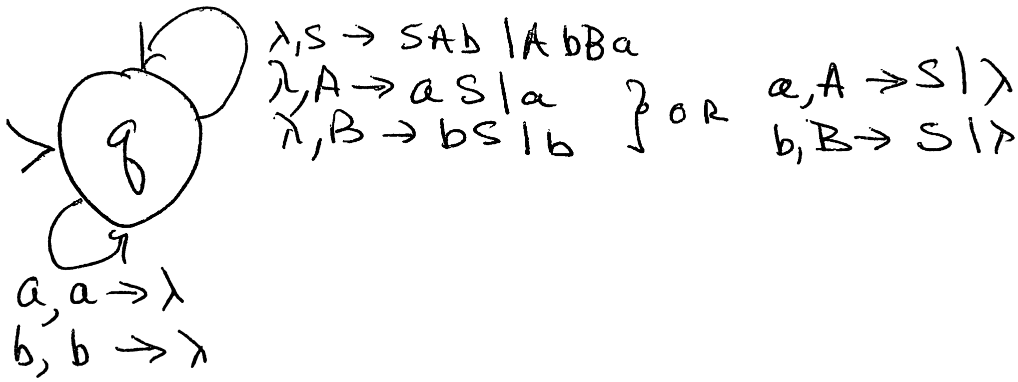
What is the algorithmic strategy, e.g., greedy, divide and conquer, dynamic programming, backtracking or randomized, associated with this **CKY** algorithm called? DYNAMIC PROGRAMMING

10/12 **8.** Consider the CFG **G = ({ S, A, B } , { a , b } , R , S)** where **R** is:

S → SAb | AbBa
A → aS | 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** ∈ **Σ ∪ {λ}, 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 = S

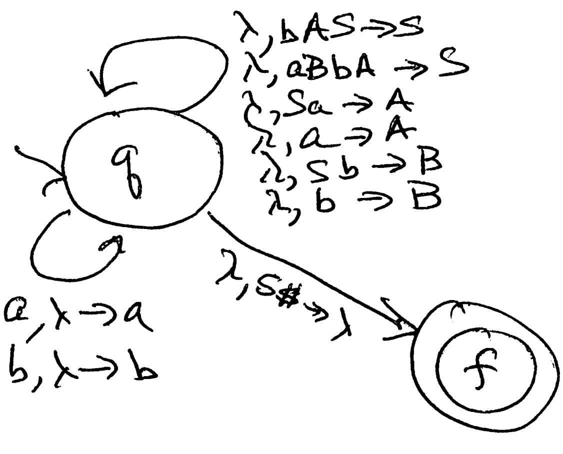


YOU ONLY DID ONE!!

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

[q, w, S] ⊢* [q, λ, λ]

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 **IDs** (Instantaneous Descriptions, **[q, x, z]**), describe how your PDA in (c) accepts strings generated by **G**.

[q, w, $] ⊢* [f, λ, λ]

9.

3 **a.)** Consider the context-free grammar **G1 = ({ S, A, B }, { 0, 1 }, R1, S)**, where **R1** is:

S → AB
A → 0A0 | λ
B → 1B1 | λ

Remove all **λ**-rules, except possibly for a start symbol, creating an equivalent grammar **G1'**. Show **all** rules.

Nullable = { S, A, B }

S' → λ | A | B | AB
S → A | B | AB
A → 0A0 | 00
B → 1B1 | 11

3 **b.)** Consider the context-free grammar **G2 = ({ S, A, B }, { 0, 1 }, R2, S)**, where **R2** is

S → AB | B
A → 1A0 | 10
B → A | AA

Remove all **unit** rules, creating an equivalent grammar **G2'**. Show **all** rules.

Unit(S) = {S, B, A}; ***Unit(A)*** = { A }; ***Unit(B)*** = {B, A}

S → AB | 1A0 | 10 | AA
A → 1A0 | 10
B → AA | 1A0 | 10

9.

3 **c.)** Consider the context-free grammar **G3 = ({ S, A, B }, { 0, 1 }, R3, S)**, where **R3** is

S → AB | BB
A → 1A0
B → 0B1 | 01

Remove all non-productive non-terminals, creating an equivalent grammar G3'. Show **all** rules.

***Productive* = { S, B }; *Unproductive* = { A }**

$S \to BB$

$B \to 0B1 \mid 01$

4 **d.)** Consider the reduced context-free grammar **G4 = ({ S, A, B }, { 0, 1 }, R4, S)**, where **R4** is

S → AABB
A → 1B0 | 10
B → 0A1 | 01

Convert an equivalent grammar G4'. Show **all** rules.

$S \to \langle AAB\rangle B$ OR $S \to \langle AA\rangle\langle BB\rangle$

$A \to \langle 1B\rangle\langle 0\rangle \mid \langle 1\rangle\langle 0\rangle$

$B \to \langle 0A\rangle\langle 1\rangle \mid \langle 0\rangle\langle 1\rangle$

$\langle AAB\rangle \to \langle AA\rangle B$

$\langle AA\rangle \to AA$

$\langle AA\rangle \to AA$

$\langle BB\rangle \to BB$

$\langle 1B\rangle \to \langle 1\rangle B$

$\langle 0A\rangle \to \langle 0\rangle A$

$\langle 0\rangle \to 0$

$\langle 1\rangle \to 1$