COT 4210 Spring 2017 **Midterm 2** **Name: KEY**

Total Points Available 67 **Your Raw Score _______** **Grade:____________**

2 **1.** Let **A = ({q1, … q10}, {0,1}, q1, {q7})** be some DFA. Assume you have computed the sets, $\mathbf{R^k_{i,j}}$, for $\mathbf{0 \le k \le 9, 1 \le i \le 10, 1, 1 \le j \le 10}$. How do you compute $\mathbf{L(A) = R^{10}_{1,7}}$, based on the previously computed values of the $\mathbf{R^k_{i,j}}$'s?

$$\mathbf{R^{10}_{1,7} = R^{9}_{1,7} + R^{90}_{1,10} \, (R^{9}_{10,10})^* \, R^{9}_{10,7}}$$

4 **2.** Write a Context Free Grammar for the language **L**, where
$\mathbf{L = \{ a^i b^j c^k \mid i \le (j + k) \}}$

S → a S c | S c | A
A → a A b | A b | λ

5 **3.** Assume **A** and **B** are arbitrary Context Free languages. Indicate, for each of the following operations, whether the language **L** is guaranteed to be Context Free (Note: Regular languages are Context Free). No proofs or examples are required.

Operation	**Is L guaranteed to be a CFL? (Y or N)**
L ⊂ A (Subset)	**N**
L = A ∩ B (Intersection)	**N**
L = A • B (Concatenation)	**Y**
L = A ⊕ B ({ x | x is in either A or B, but not both }	**N**
L = Max(A) ({ x | x∈A but no xy∈A, |y|>0 }	**N**

8 **4.** Use the Pumping Lemma for CFLs to show that the following language **L** is not Context Free.
L = { $a^n b^{2^n}$ | n>0 }. Be explicit as to why each case you analyze fails to be an instance of **L**.
I will do the first two steps for you.

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
ME: Choose $w = a^N b^{2^N} = uvwxy$, $|vwx| \le N$, $|v| + |x| > 0$, and $\forall i\ uv^iwx^iy \in L$
PL:
Case1: Assume vwx is over a's and perhaps b's. This means that vwx must contain at least one a and at most N-1 b's. Let i=2. Assuming the case where it has just one a, the string uv^2wx^2y would start with N+1 a's and so must have 2^{N+1} b's.
Now, 2^N is always greater than N, for N>0, so $2^{N+1} = 2^N + 2^N$ is greater than $2^N + N$ which is greater than $2^N + N-1$. Thus, there are not sufficient number of b's to accommodate number of a's and so $uv^2wx^2y \notin L$.
Case2: Assume vwx is over only b's. Let i=2. Then $uv^2wx^2y = a^N b^{2^N+|vx|}$, where $|vx|>0$ and so there are too many b's relative to the number of a's and so $uv^2wx^2y \notin L$.

Cases 1 and 2 cover all possibilities, so L is not a CFL.

6 **5.** Consider some languages **A** and **B** that are both Context Free, and neither is Regular. Define **L = A ∪ B**. Give explicit examples of languages **A** and **B**, and explicitly describe **L**, or argue that this is impossible based on some well-known result, for each of the following.

a.) **L** is Regular

$A = \{ a^n b^m \mid m \ge n, m,n \ge 0 \}$; $B = \{ a^n b^m \mid m \le n, m,n \ge 0 \}$; $L = A \cup B = a^*b^*$

b.) **L** is Context Free, non-Regular.

$A = \{ a^n b^n \mid ,n \ge 0 \}$; $B = A$; $L = A \cup B = A = = \{ a^n b^n \mid ,n \ge 0 \}$

c.) **L** is Context Sensitive, non-Context-Free.

That is impossible as CFLs are known to be closed under union. The proof is trivial when employing CFGs.

10 **6.** Present the **CKY** recognition matrix for the string **abbccc** assuming the Chomsky Normal Form grammar, **G = ({S, A, B, C, X, Y, Z }, {a,b,c}, R, S)**, specified by the rules **R**: Note: **abbccc** is in **L(G)** so that should help you if you make an error and don't see **S** at bottom of matrix.

S → AB
A → XA | a
B → CZ | BZ | b | c
C → YB
X → a
Y → b
Z → c

	a	b	b	c	c	c
1	*AX*	*BY*	*BY*	*BZ*	*BZ*	*BZ*
2	*S*	*C*	*BC*	*B*	*B*	
3		*BC*	*BC*	*B*		
4	*S*	*BC*	*BC*			
5	*S*	*BC*				
6	*S*					

A little help from your friends

Non-Terminal	First Symbol in Rules	Second Symbol in Rules
S	None	None
A	S → AB	A → XA
B	B → BZ	S → AB; C → YB
C	B → CZ	None
X	A → XA	None
Y	C → YB	None
Z	None	B → BZ; B → CZ

8 **7**. Prove that Context-Free Languages are closed under **div3** where **L** is a CFL over the alphabet **Σ**, and **div3(L) = { x | xy ∈ L** and **|x| modulo 3 = 0** and **|y| ∈ {0,1,2} }.**
In words, we remove as few characters as needed from the end of a string in **L**, so the resulting string's length is a multiple of **3**.
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**.
You must be very explicit, describing what is produced by each transformation you apply and what kind of language results.

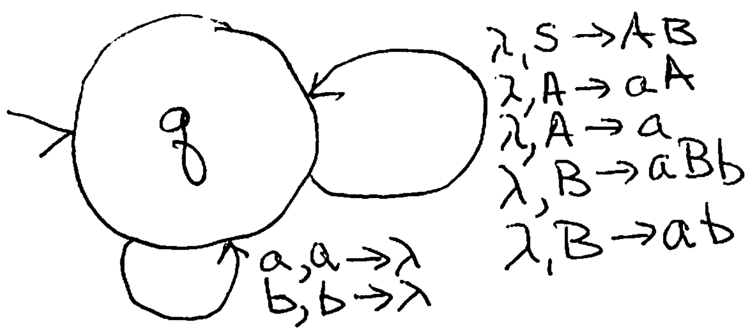
div3(L) = h(f(L) ∩ ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ)))
First, all finite sets are Regular and Regular are closed under concatenation and union, so ΣΣΣ is Regular as is ({λ} ∪ Σ ∪ ΣΣ)). Next Regular are closed under Kleene star, homomorphism and, again concatenation, so ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ)) is Regular.
Second, Context Free are closed under homomorphism, substitution and intersection with regular sets, so f(L) ∩ ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ)) and h(f(L) ∩ ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ))) are both Context Free.
Now, ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ)) = { xy' | xy ∈ Σ* *and* ***|x| modulo 3 = 0*** *and* ***|y| ∈ {0,1,2} }***
f(L) = { f(<u>w</u>) | w ∈ L }.
So f(L) ∩ ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ)) = { xy' | xy ∈ L *and* ***|x| modulo 3 = 0*** *and* ***|y| ∈ {0,1,2} }***
Thus, h(f(L) ∩ ((ΣΣΣ)* g({λ} ∪ Σ ∪ ΣΣ))) = { x | xy ∈ L *and* ***|x| modulo 3 = 0*** *and* ***|y| ∈ {0,1,2} }.***
This is precisely div3(L) so CFLs are closed under div3.

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

S → AB
A → aA | a
B → aBb | ab

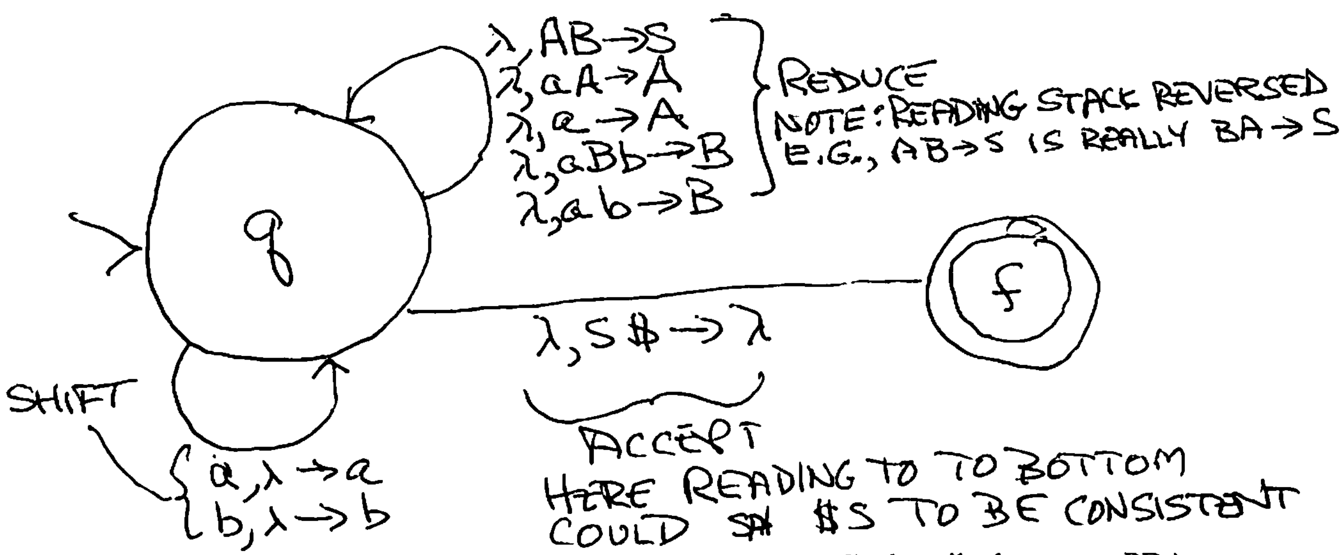
In the PDAs below, you may (and are encouraged) to use a transition diagram where transitions have arcs with labels of form **a, α → β** where **a** ∈ **Σ∪{λ}**, **α, β** ∈ **Γ***. Note: This just means that you can use extended stack operations that push more than one symbol onto stack.

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



b.) Now, using the notation of **IDs** (Instantaneous Descriptions, **[q, x, z]**), describe how your PDA 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 accepts strings generated by **G**.
[q, w, $] |--* [f, λ, λ]

9. Consider the context-free grammar **G = ({ S, A, B, C, D }, { a,b,c }, R, S)**, where **R** is:

S → BC | AC | ABC
A → aA | λ
B → ABb | Bb
C → bCc | bc
D → bBc | Dc | λ

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

S → BC | AC | ABC | C
A → aA | a
B → ABb | Bb
C → bCc | bc
D → bBc | Dc | c

2 **b.)** Remove all **unit** rules from **G'**, creating an equivalent grammar **G**''. Show all rules.
Chain(S) = { S, C }; Chain(A) = { A }; Chain(B) = { B }; Chain (C) = { C }; Chain(D) = { D }
S → BC | AC | ABC | bCc | bc
A → aA | a
B → ABb | Bb
C → bCc | bc
D → bBc | Dc | c

2 **c.)** Remove all unproductive symbols, creating an equivalent grammar **G'''**. Show all rules.
Productive = { S, A, C, D }; Unproductive = { B }
S → AC | bCc | bc
A → aA | a
C → bCc | bc
D → Dc | c

2 **d.)** Remove all unreachable symbols, creating an equivalent grammar $\mathbf{G^{iv}}$. Show all rules.
Unreachable = { D }
S → AC | bCc | bc
A → aA | a
C → bCc | bc

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

S → AC | <bC><c> | <b><c>
A → <a>A | a
C → <bC><c> | <b><c>
<bC> → <b>C
<a> → a
<b> → b
<c> → c