COT 4210 Fall 2016 Sample Midterm 2 Key Name: KEY

1. Assuming you have computed the sets, $\mathbf{R^k_{i,j}}$, for each pair of states, $(\mathbf{q_i},\mathbf{q_j})$, in some DFA. How is $\mathbf{R^{k+1}_{i,j}}$ calculated, where **k+1** is no greater than the number of states in the associated DFA?

$$R^{k+1}_{i,j} = R^k_{i,j} + R^k_{i,k+1} \left(R^k_{k+1,k+1} \right)^* R^k_{k+1,j}$$

2. Write a **Context Free Grammar** for the language
$\mathbf{L = \{ a^k\ b^m\ c^n \mid k = n + m, \text{ or } m = k + n, \text{ or } n = k + m, k>0, m>0, n>0 \}}$.

S → aAc | aA'bbC'c
A → aAc | aA'b | bC'c
A' → aA'b | λ
C' → bC'c | λ

3. Use Myhill-Nerode to show that the language of Problem 2 is not Regular. That language is
$\mathbf{L = \{ a^k\ b^m\ c^n \mid k = n + m, \text{ or } m = k + n, \text{ or } n = k + m, k>0, m>0, n>0 \}}$.

Consider Classes $[a^i]_{R_L}$
$a^i\, b\, c^{i+1} \in L$
However,
$a^j\, b\, c^{i+1} \notin L$ ***whenever*** $j \neq i$
Thus, $[a^i] = [a^j]$ ***iff*** $i = j$
This shows an infinite number of equivalence classes are induced by R_L ***and so L is not Regular.***

4. Consider the language
$\mathbf{L = \{ a^n\, b^{n!} \mid n>0 \}}$.
Use the Pumping Lemma for Context-Free Languages to show that **L** is not context-free.

PL: Provides N>0
We: Choose $a^N b^{N!} \in L$
PL: Splits $a^N b^{N!}$ ***into uvwxy,*** $|vwx| \leq N$, $|vx| > 0$, ***such that*** $\forall i \geq 0\ uv^iwx^iy \in L$
We: Choose i=2
Case 1: vwx contains only b's, then we are increasing the number of b's while leaving the number of a's unchanged. In this case uv^2wx^2y ***is of form*** $a^N b^{N!+c}$, $c>0$ ***and this is not in L.***
Case 2: vwx contains some a's and maybe some b's. Under this circumstances uv^2wx^2y ***has at least N+1 a's and at most N!+N-1 b's. But*** $(N+1)! = N!(N+1) = N!*N+N \geq N! + N > N!+N-1$ ***and so is not in L.***
Cases 1 and 2 cover all possible situations, so L is not a CFL.

5. Present the **CKY** recognition matrix for the string **bbabb** assuming the Chomsky Normal Form grammar, **G = ({S,A,B,C,D }, {a,b}, R, S)**, specified by the rules **R**:

S → **AB | BA | BD**
A → **CS | CD | a**
B → **DS | b**
C → **a**
D → **b**

	b	b	a	b	b
1	**BD**	**BD**	**AC**	**BD**	**BD**
2	**S**	**S**	**SA**	**S**	
3	**B**	**SB**	**SA**		
4	**SB**	**SB**			
5	**SB**				

6. Choosing from among **(Y) yes**, **(N) No**, categorize each of the following closure properties. No proofs are required.

Problem / Language Class (C)	**Regular**	**Context Free**
Closed under union with Context Free languages?	N	Y
Closed under quotient with languages of its own class (C), i.e., L1/L2	Y	N
Closed under difference with languages of its own class (C), i.e., (difference (L1, L2) = L1 – L2)?	Y	N
Closed under intersection with Context Free languages?	Y	N

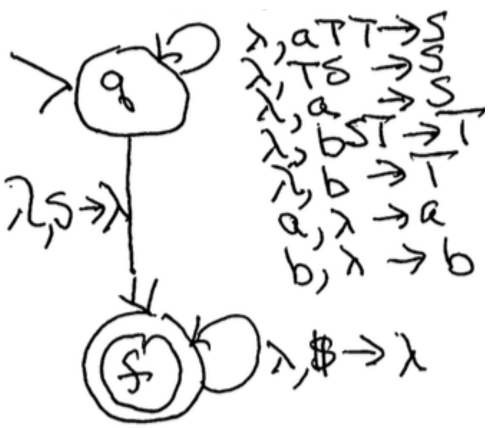
7. Prove that any class of languages, ***C***, closed under union, concatenation, intersection with regular languages, homomorphism and substitution (e.g., the Context-Free Languages) is closed under **Erase Middle with Regular Sets (em)**, where **L** ∈ ***C***, **R** is Regular, **L** and **R** are over the alphabet **Σ**, and **L em R = { xz | x,z** ∈ Σ^+ and **∃y** ∈ **R,** such that **xyz** ∈ **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 distinct new character associated with each **a∈Σ**.
You must be very explicit, describing what is produced by each transformation you apply.

L em R = h(f(L) ∩ Σ^+ g(R) Σ^+)
f(L) = { w | w ∈ L } where w has some (or none) of its letters primed. f(L) is a CFL since CFLs are closed under substitution.
g(R) = { y' | y ∈ R } where y' has all of its letter primed. g(R) is Regular since Regular languages are closed under homomorphism.
Σ^+ g(R) Σ^+ = { xy'z | x,z ∈Σ^+ and y ∈ R, This is a Regular language since Regular languages are closed under concatenation.
f(L) ∩ Σ^+ g(R) Σ^+ = { xy'z | xyz ∈ L and y ∈ R} . This is a CFL since CFLs are closed under intersection with Regular.
L em R = h(f(L) ∩ Σ^+ g(R) Σ^+ = { xz | ∃y ∈ R where xyz ∈ L } is a CFL since CFLs are closed under homomorphism.

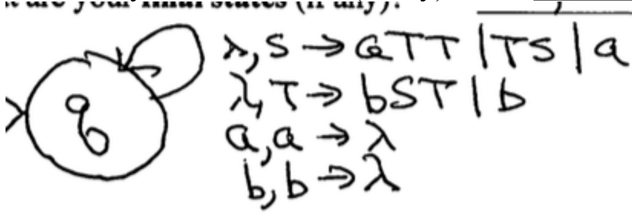
8. Consider the CFG **G = ({ S, T }, { a, b }, R, S)** where **R** is:

S → a T T | T S | a
T → b S T | b

a.) Present a pushdown automaton that accepts the language generated by this grammar. 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: I am encouraging you to use extended stack operations.



What parsing technique are you using? (Circle one) **top-down** or **bottom-up**
How does your PDA accept? (Circle one) **final state** or **empty stack** or **final state and empty stack**
What is the **initial state**? q
What is the **initial stack content**? $
What are your **final states** (if any)? f



Or can have **a,S → TT | λ; λ,S → TS**
b,T → ST | λ

What parsing technique are you using? (Circle one) <u>**top-down**</u> or **bottom-up**
How does your PDA accept? (Circle one) **final state** or <u>**empty stack**</u> or **final state and empty stack**
What is the **initial state**? ***q***
What is the **initial stack content**? ***S***
What are your **final states** (if any)? ***None***

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

[q, w, \$] ⇒* [f, λ, λ] if by final state and empty stack (my solution on (a) Bottom-Up)

[q, w, S] ⇒* [q, λ, λ] if by empty stack (my solution on (a) Top-Down)

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

S → SAB | BA
A → AB | a
B → bS | b | λ

a.) Remove all **λ**-rules from **G**, creating an equivalent grammar **G'**. Show all rules.
Nullable = {B}
G':
S → SAB | SA | BA | A
A → AB | a
B → bS | b

b.) Remove all **unit** rules from **G'**, creating an equivalent grammar **G''**. Show all rules.
Unit(S)=Chain(S)={S,A}; Unit(A)={A}; Unit(B)={B}
G'':
S → SAB | SA | BA | AB | a
A → AB | a
B → bS | b

c.) Convert grammar **G''** to its **Chomsky Normal Form** equivalent, **G'''**. Show all rules.
G''':
S → S<AB> | SA | BA | AB | a
A → AB | a
B → <b>S | b
<AB> → AB
<b> → b

In exam I may have some Unproductive non-terminals and some Unreachable ones.