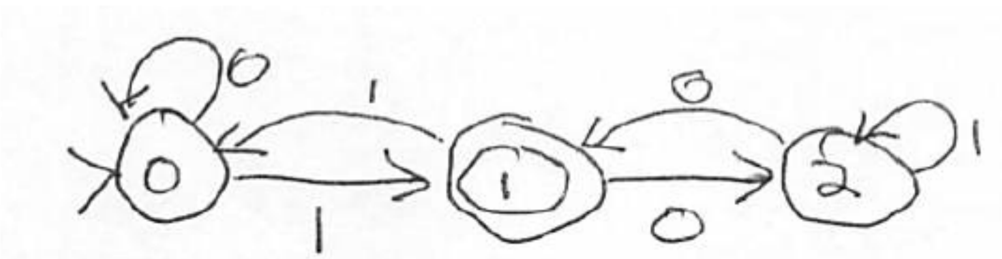
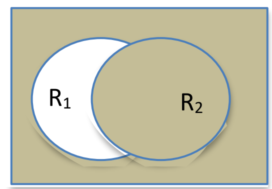
COT 4210 Fall 2019 Sample Midterm Key Name:________________________

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

1. Present the transition diagram for a **DFA** that accepts the set of binary strings that represent the magnitude of numbers that have a remainder of 1, when divided by 3. Numbers are read most to least significant digit, so 01 (1), 111 (7) and 10011 (19) are accepted, but 0000 (0), 110 (6) and 010001 (17) are not. Note: Leading zeroes are allowed.



2. Consider the following assertion:
Let $\mathbf{R_1}$ and $\mathbf{R_2}$ be regular languages that are recognized by $\mathbf{A_1}$ and $\mathbf{A_2}$, respectively, where
$\mathbf{A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)}$ and $\mathbf{A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)}$ are **DFA**s.
Show that $\mathbf{L = \sim(R_1 - R_2) = \{ w \mid w}$ is in the complement of $\mathbf{(R_1 - R_2) \}= \sim(R_1 \cap \sim R_2) = \sim R_1 \cup R_2}$
is also regular, where ~ means set complement and – means set difference.



Present a **DFA** construction $\mathbf{A_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)}$, where $\mathbf{L(A_3) = \sim(R_1 - R_2)}$. You must clearly define $\mathbf{Q_3, \delta_3, q_3,}$ and $\mathbf{F_3}$. However, I do not require you to prove or even justify your choice of final set, just to get it right.

$$Q_3 = Q_1 \times Q_2$$

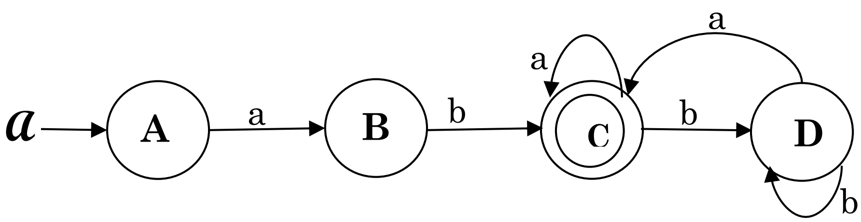
$$q_3 = \langle q_1, q_2 \rangle$$

$$\delta_3(\langle p, q \rangle, a) = \langle \delta_1(p,a), \delta_2(q,a) \rangle$$

$$p \in Q_1,\ q \in Q_2,\ a \in \Sigma$$

$$F_3 = (Q_1 - F_1) \times Q_2 \cup Q_1 \times F_2$$

3. Let **L** be defined as the language accepted by the finite state automaton a:



a.) Present the regular equations associated with each of a's states, solving for the regular expression associated with the language recognized by a.

$$A = \lambda \quad B = Aa = a$$
$$C = Bb + Da + Ca = ab + Da + Ca$$
$$D = Cb + Db = Cb^+$$
$$C = ab + Cb^+a + Ca$$
$$= ab(b^+a + a)^* = ab(b^*a)^*$$

b.) Assuming that we designate **A** as state **1**, **B** as state **2**, **C** as state **3** and **D** as state **4**. Kleene's Theorem allows us to associate regular expressions $R^k_{i,j}$ with a, where $i \in \{1..4\}$, $j \in \{1..4\}$, and $k \in \{0..4\}$.
The following are values of

$R^3_{1,3}$ = **aba*** , $R^3_{2,3}$ = **ba*** , $R^3_{3,3}$ = **a*** , $R^3_{3,4}$ = **a*b**

What are the values of the following?

$R^3_{4,3} = a^+$, $R^3_{4,4} = a^+b + b = a^*b$

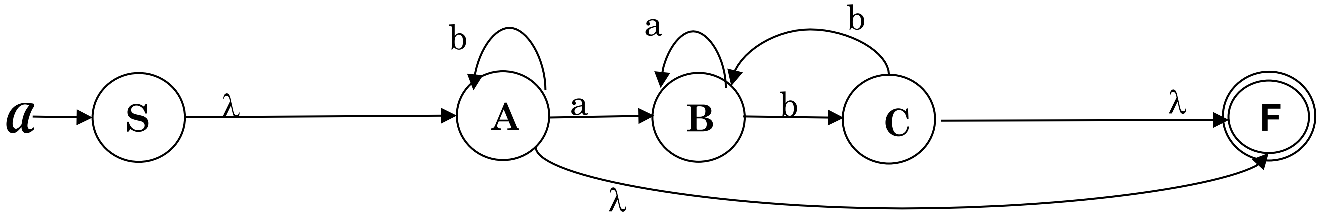
How is $R^4_{3,3}$ calculated from the set of $R^3_{i,j}$'s above? Give this abstractly in terms of the $R^3_{i,j}$'s

$$R^4_{33} = R^3_{33} + R^3_{34}\, R^{3*}_{44}\, R^3_{43}$$

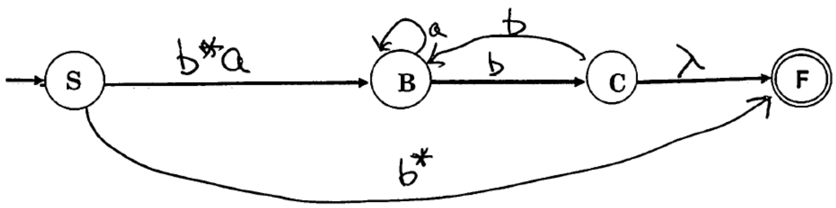
What expression does $R^4_{3,3}$ evaluate to, given that you have all the component values?

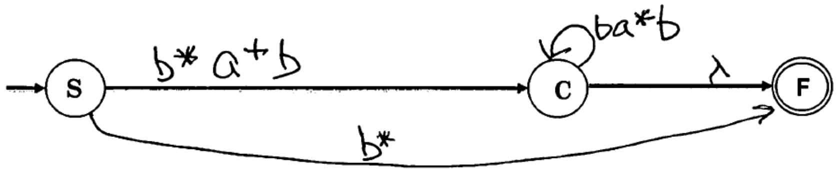
$$R^4_{33} = a^* + a^*b\,(a^*b)^*a^+$$

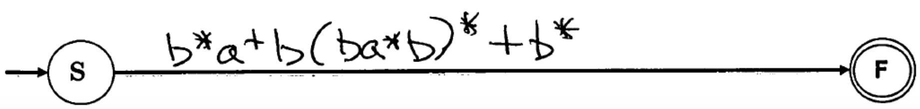
4. Let **L** be defined as the language accepted by the NFA $\mathcal{A}$:



Using the technique of replacing transition letters by regular expressions and then ripping states from a GNFA to create new expressions, develop the regular expression associated with the automaton $\mathcal{A}$ that generates **L**. I have included the states of GNFA's associated with removing states **A**, **B** and then **C**, in that order. You must use this approach of collapsing one state at a time, showing the resulting transitions with non-empty regular expressions.



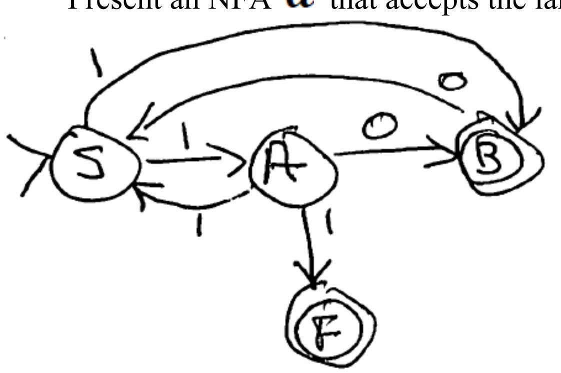




5. Consider the regular grammar **G = ({S, A, B}, {0, 1}, S, P)** where **P** is the set of rules:

S → **1 B** | **1 A**
A → **0 B** | **1 S** | **1**
B → **0 S** | λ

Present an NFA $\mathcal{A}$ that accepts the language generated by **G**:



6. Analyze the following language, **L**, proving it is **non**-regular by showing that there are an **infinite** number of equivalence classes formed by the relation $\mathbf{R_L}$ defined by:

$$\mathbf{x\ R_L\ y} \text{ if and only if } [\ \forall \mathbf{z} \in \{\mathbf{a,b}\}^*, \mathbf{xz} \in \mathbf{L} \text{ exactly when } \mathbf{yz} \in \mathbf{L}\ ].$$

where $\mathbf{L = \{ a^i b^j \mid i < j \}}$,

You don't have to present all equivalence classes, but you must demonstrate a pattern that gives rise to an infinite number of classes, along with evidence that these classes are distinct from each other.

CONSIDER EQUIV CLASSES $[a^i]_{R_L}$

$a^i b^{i+1} \in L$

BUT

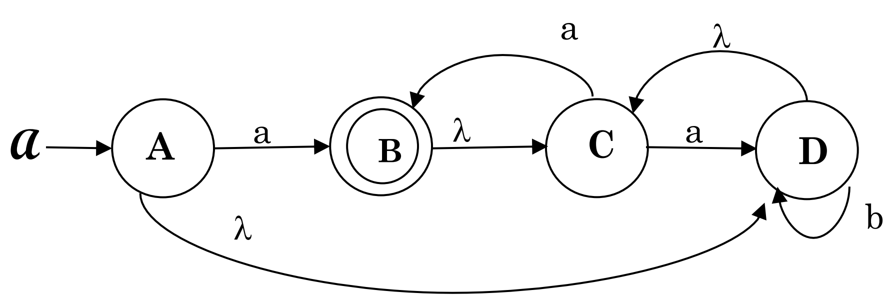
$a^j b^{i+1} \notin L$ FOR ALL $j > i$

SO $[a^i] \neq [a^j]$ $j > i$

AND HENCE EACH SUCH CLASS DIFFERS
AND THERE ARE AN INFINITE NUMBER OF
SUCH CLASSES

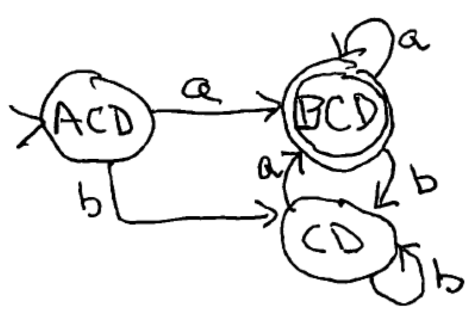
7. Let **L** be defined as the language accepted by the finite state automaton $\mathcal{A}$:

a.) Fill in the following table, showing the **λ**-closures for each of $\mathcal{A}$'s states.



State	**A**	**B**	**C**	**D**
λ-closure	***A, C, D***	***B, C***	***C***	***C, D***

b.) Convert $\mathcal{A}$ to an equivalent deterministic finite state automaton. Use states like **AC** to denote the subset of states **{A,C}**. Be careful -- **λ**-closures are important.



8. **OddLetters(L)={ $x_1\, x_3 \ldots x_{2n+1}$ | where $x_1x_2x_3 \ldots x_{2n}x_{2n+1} \in$ L or $x_1x_2x_3 \ldots x_{2n}x_{2n+1}x_{2n+2} \in$ L}, where each $x_i \in \Sigma$**

Show that any class of languages (in particular Regular and Context Free Languages) that is closed under substitution, concatenation and intersection with Regular Languages is also closed under **OddLetters**. A constructive solution is all I ask; no proof required. I'll help. Consider using substitutions **f(a) = {a, a'}; g(a) = a'; h(a) = a, h(a') = λ**,
where **a** ∈ **Σ** and **a'** is a new symbol uniquely associated with the symbol **a**.

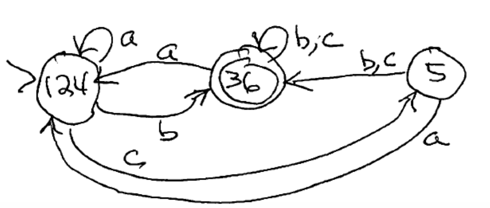
$OddLetters(L) = h(f(L) \cap (\Sigma \cdot g(\Sigma))^* (\Sigma \cup \Sigma \cdot g(\Sigma)))$

9. Given a DFA denoted by the transition table shown below, and assuming that **1** is the start state and **3** and **6** are final states, fill in the equivalent states matrix I have provided. Use this to create an equivalent, minimal state DFA.

	a	b	c
>1	4	6	5
2	1	6	5
<u>3</u>	2	3	3
4	1	3	5
5	1	3	6
<u>6</u>	1	3	6

2	1,4				
<u>3</u>	X	X			
4	3,6	3,6	X		
5	1,4 3,6 5,6 X	3,6 5,6 X	X	5,6X	
<u>6</u>	X	X	1,2	X	X
	1	2	<u>3</u>	4	5

{1,2,4} {3,6} {5}



10. 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

11. 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, $|vwx| \le N$, $|vx| > 0$, 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.

12. 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***				

13. 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, A, B }, { a, b }, R', S):
S → SAB | SA | BA | A
A → AB | a ***Note: There is a rule A → A but it was removed***
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, A, B }, { a, b }, R'', S):
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, A, B }, { a, b }, R''', S):
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.

14. 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 $\mid$ x is in either A or B, but not both }	***N***
L = Max(A) ({ x $\mid$ x∈A but no xy∈A, $\lvert y \rvert$>0 }	***N***