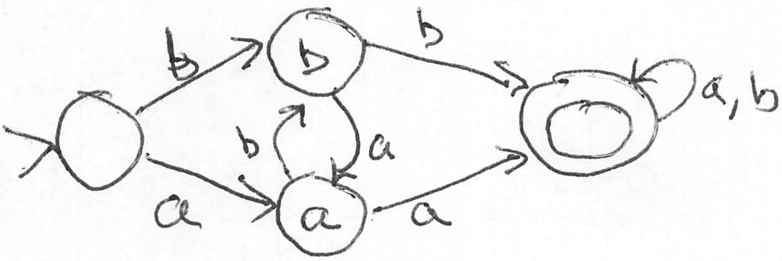
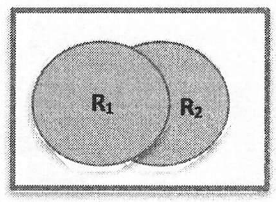
COT 4210 **Fall 2018** **Midterm#1** **Name:** KEY

Total Points Available 60 **Your Raw Score** ______ **Grade:** ________

5 1. Present the transition diagram for a **DFA** that accepts the set of strings over the alphabet **{a,b}** that are of length >0 and have either two consecutive **a**'s and/or two consecutive **b**'s. Examples that are in the language include **ababba** (there are two b's) and **baaab** (two a's in a row), but not **ababab**.



5 2. Consider the following assertion:
Let **R₁** and **R₂** be regular languages that are recognized by DFAs **A₁** and **A₂**, respectively, where $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are **DFA**s. Show that
$L = \sim R_1 \cap \sim R_2 = \{ w \mid w \text{ is not in either of } R1 \text{ or } R2 \}$
is also regular, where ~ means NOT and $\cap$ means set intersection.
Note: The figure on right shows the set as the **white** part.
Present a **DFA** construction $A_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$, where
$L(A_3) = \sim R_1 \cap \sim R_2 = \sim(R_1 \cup R_2)$. You do not need to present a formal proof that it works, but you must clearly define Q_3, δ_3, q_3, and F_3.



$Q_3 = Q_1 \times Q_2 \quad q_3 = \langle q_1, q_2 \rangle$

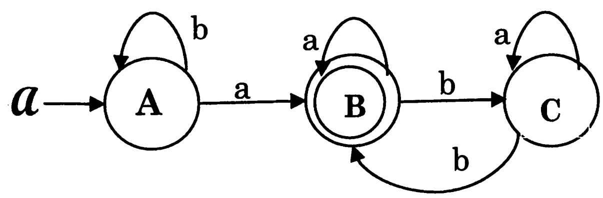
$F_3 = (Q_1 - F_1) \times (Q_2 - F_2)$

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

4 3. Define **SuperSub (L)** = { **xyz** | **x,y,z** $\in \Sigma^+$ **and** ($\exists$ **u,v** $\in \Sigma^+$) where **xuyvz** $\in$ **L** }
Assuming that Regular languages are already shown to be closed under **Substitution**, **Homomorphism**, **Concatenation** and **Intersection with Regular Languages**, show they are closed under **SuperSub.** You should find it useful to employ the substitution **f(a) = {a, a'}**, and the homomorphisms **g(a) = a'** and **h(a) = a, h(a') = λ**. Here **a** $\in \Sigma$ and **a'** is a new symbol associated with **a**. You do not need to show your construction works, but it must be based on the meta technique I showed in class for languages closed as above.

$h(f(L) \cap (\Sigma^+ g(\Sigma^+) \Sigma^+ g(\Sigma^+) \Sigma^+))$

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



8 **a.)** Present the regular equations associated with each of $\mathcal{A}$'s states, solving for the regular expression associated with the language recognized by $\mathcal{A}$. You must finish by showing the final expression for the language accepted by this automaton. Hint: That's the solution to **B**. Don't try to optimize.

$$A = \lambda + Ab = b^*$$
$$B = Aa + Cb + Ba = (b^*a + Cb)a^*$$
$$= b^*a^+ + Cba^*$$

$$C = Bb + Ca = Bba^*$$
$$B = b^*a^+ + Bba^*ba^*$$
$$= b^*a^+(ba^*ba^*)^*$$

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

$\boldsymbol{R^2_{1,1} = b^*}$ $\boldsymbol{R^2_{1,2} = b^*a^+}$ $\boldsymbol{R^2_{1,3} = b^*a^+b}$
$\boldsymbol{R^2_{2,1} = \phi}$ $\boldsymbol{R^2_{2,2} = a^*}$ $\boldsymbol{R^2_{2,3} = a^*b}$
$\boldsymbol{R^2_{3,1} = \phi}$ $\boldsymbol{R^2_{3,2} = ba^*}$ $\boldsymbol{R^2_{3,3} = \lambda + a + ba^*b}$

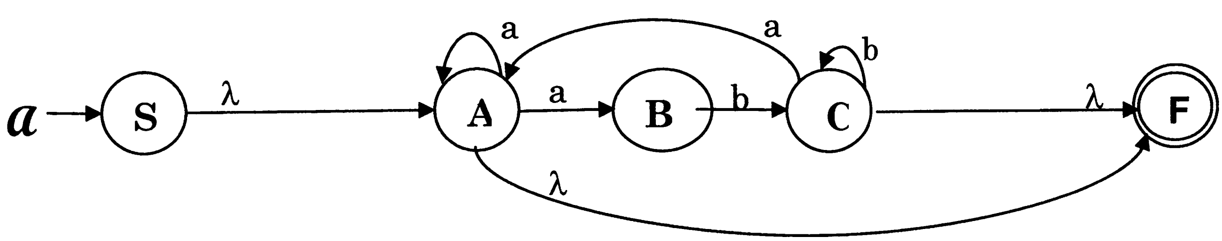
How is $R^3_{1,2}$ calculated from the set of $R^2_{i,j}$'s above? Give this abstractly in terms of the $R^k_{i,j}$'s

$$R^3_{12} = R^2_{12} + R^2_{13}(R^2_{33})^* R^2_{32}$$

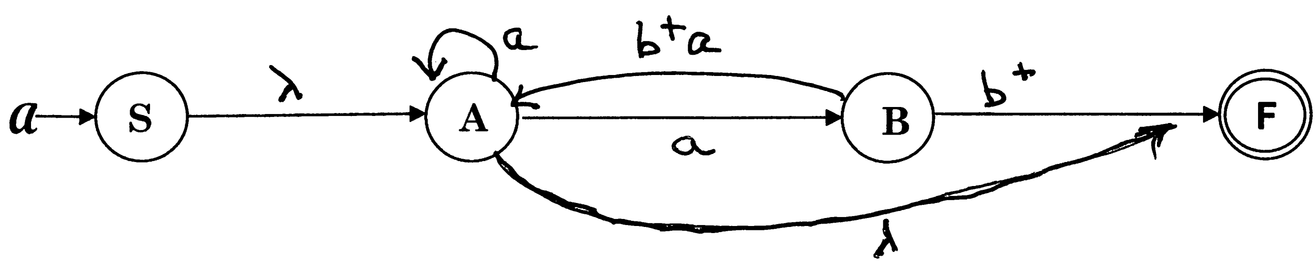
What expression does $R^3_{1,2}$ evaluate to, given that you have all the component values?

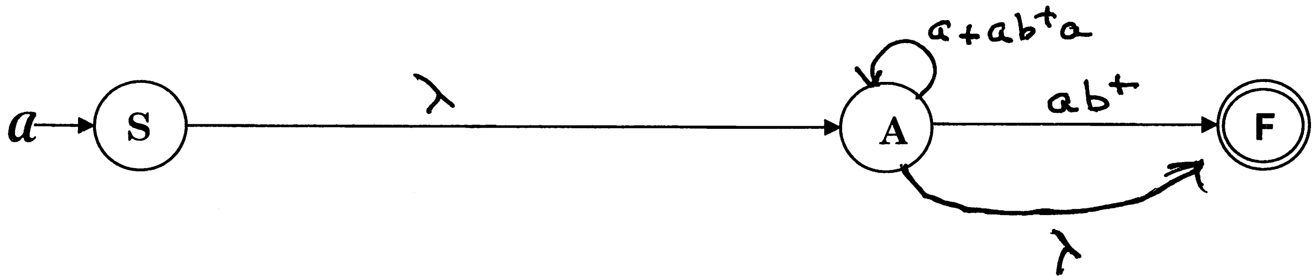
$$b^*a^+ + b^*a^+b(ba^*b + a)^* ba^*$$

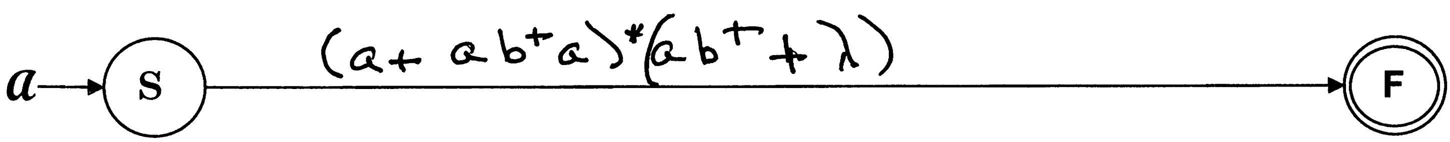
7 **5.** Let **L** be defined as the language accepted by the NFA $\mathcal{A}$:



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 a GNFA associated with removing states **C**, **B** and then **A**, in that order. You must use this approach of collapsing one state at a time, showing the resulting transitions with non-empty regular expressions







$L = (a+ab^+a)^*(ab^+ + \lambda)$

4 **6.** The class of regular expressions over some alphabet **Σ** is defined inductively from a set of primitive regular expressions and closure operations as follows:

ϕ is a regular expression
λ is a regular expression
a is a regular expression, whenever **a** $\in \Sigma$
If **R** and **S** are regular expressions, then so are

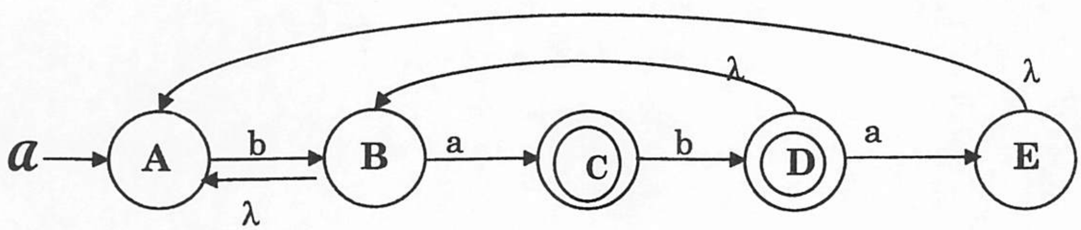
$\mathbf{R + S}$
$\mathbf{R \cdot S}$
$\mathbf{R^*}$

Consider the regular expression $\mathbf{R = (10 + 101)^+}$
Note: The first + is an "or" and the second is related to Kleene *
Show a right linear grammar that generates **R**.

$S \to 1T \mid 1U$
$T \to 0S \mid 0$
$U \to 0V$
$V \to 1S \mid 11$

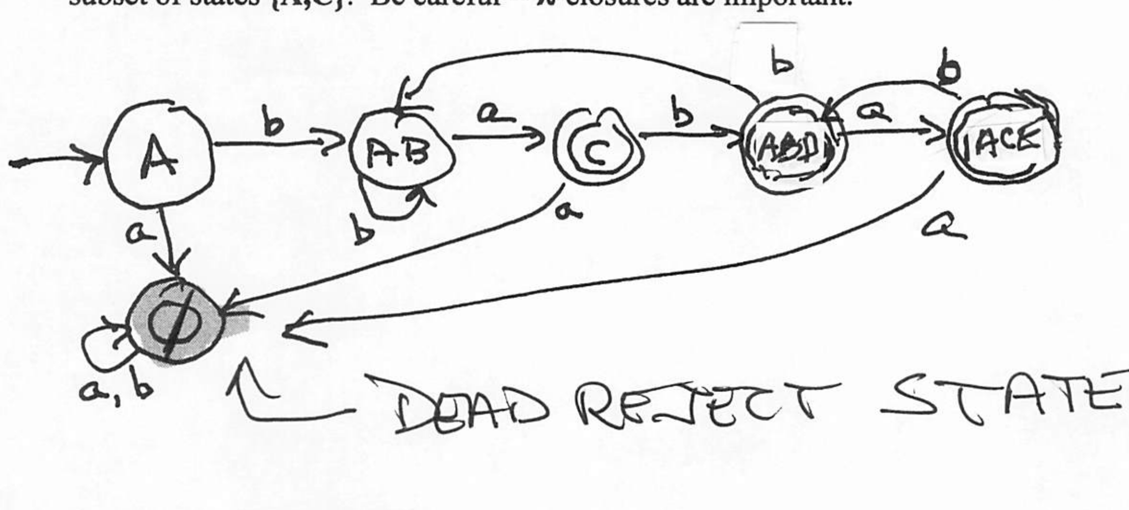
7. Let **L** be defined as the language accepted by the finite state automaton $\mathcal{A}$:

3 **a.)** Fill in the following table, showing the λ-closures for each of $\mathcal{A}$'s states. Note: **F={C, D, E}**.



State	**A**	**B**	**C**	**D**	**E**
λ-closure	A	AB	C	ABD	AE

5 **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.



6 **8.** Do <u>ONE</u> of (a) or (b), each of which accomplishes the same thing, that is, shows that
L = { $a^i b^j c^k$ | k > max(i,j), i,j > 0 } is not regular

a.) Apply the Pumping Lemma to show **L** is **NOT** regular. Be sure to differentiate the steps (contributions) to the process provided by the Pumping Lemma and those provided by you. Be sure to be clear about the contradiction. I'll even start the process for you. Please be careful to note constraints on the Pumping Lemma as I have when noting that **N>0**.

You: L is Regular
PL: Provides N>0 associated with L

YOU: $a^N b c^{N+1} \in L$

PL: $a^N b c^{N+1} = xyz$, $|xy| \le N$, $|y| > 0$, $\forall i \ge 0\ xy^iz \in L$

YOU: $i = 2$

$a^{N+|y|} b c^{N+1}$ IS IN L ACCORDING TO PL

BUT $N+|y| \ge N+1$ SINCE $|y|>0$

AND THEN $a^{N+|y|} b c^{N+1} \notin L$

$\times$ SO L IS <u>NOT</u> REGULAR

OR

b.) Analyze the 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:

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

You don't have to present all equivalence classes; just demonstrate a pattern that gives rise to an infinite number of classes, along with evidence that these classes are distinct from one another.

CONSIDER ALL CLASSES $[a^i b]_{R_L}$ $\forall i > 0$

CLEARLY $a^i b c^{i+1} \in L$ WHEN $i > 0$

BUT $a^j b c^{i+1} \notin L$ WHEN $j > i$

SO $[a^i b]_{R_L} \ne [a^j b]_{R_L}$ WHEN $i \ne j$ $i,j > 0$

THUS THE INDEX OF R_L IS INFINITE

AND SO, BY M-N, L IS <u>NOT</u> REGULAR

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

	a	b	c
>1	5	3	2
2	5	2	2
3	2	4	6
4	5	4	2
5	4	5	5
6	5	3	1

2	X				
3	X	2,5 2,4 2,6 x			
4	X	✓	2,5 2,6 x		
5	X	4,5	2,4 4,5 5,6 x	2,5	
6	X	2,3 1,2 x	2,5 3,4 1,6 x	3,4 1,2 x	4,5 3,5 1,5 x
	1	**2**	**3**	**4**	**5**

Don't forget to construct and write down your new, equivalent DFA!! Be sure to clearly mark your start state and your final state(s). In your minimum state DFA, label merged states with the states that comprise the merge. Thus, if **1, 2** and **3** are indistinguishable, label the merged state as **123**.

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