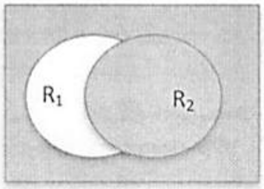
COT 4210 **Fall 2016** **Midterm#1** **Name:** KEY

Total Points Available 65 **Your Raw Score** _______ **Grade:**____________

4 **1.** Present the transition diagram for a **DFA** that accepts the set of binary strings that represent numbers that have a remainder of either 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 010010 (18) are not. Note: Leading zeroes are allowed.

5 **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 **DFAs**.
Show that $\mathbf{L = \sim(R_1 - R_2) = \{ w \mid w \text{ is in the complement of } (R_1 - R_2) \} = \sim(R_1 \cap \sim R_2) = \sim R_1 \cup R_2}$
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 do not need to present a formal proof that it works, but you must clearly define $\mathbf{Q_3, \delta_3, q_3}$, and $\mathbf{F_3}$.
I do not require you to justify your choice of final set, just to get it right.

$Q_3 = Q_1 \times Q_2$

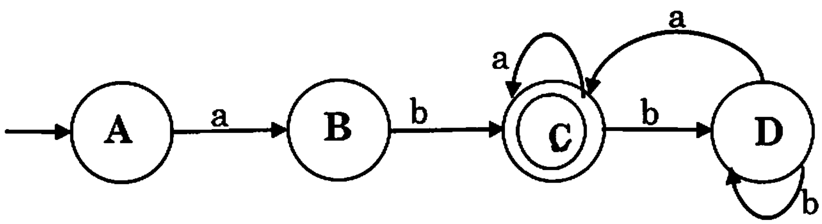
$q_3 = \langle q_1, q_2 \rangle$

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

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

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

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



7 **a.)** Present the regular equations associated with each of $\mathcal{A}$'s states, solving for the regular expression associated with the language recognized by $\mathcal{A}$.

$A = \lambda \quad B = Aa = a$

$C = Bb + Da + Ca = ab + Da + Ca$

$D = Cb + Db = Cb^+$

$C = ab + Cb^+a + Ca$

$= ab(b^+a + a)^* = ab\,(b^*a)^*$

5 **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 $\mathcal{A}$, where $i \in \{1..4\}, j \in \{1..4\}$, and $k \in \{0..4\}$.

The following are values of

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

What are the values of the following?

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

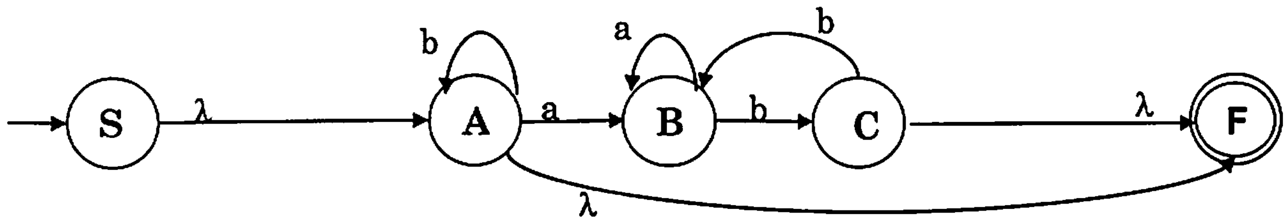
How is $R^4_{3,3}$ calculated from the set of $R^3_{i,j}$'s above? Give this abstractly in terms of the $R^3_{i,j}$'s

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

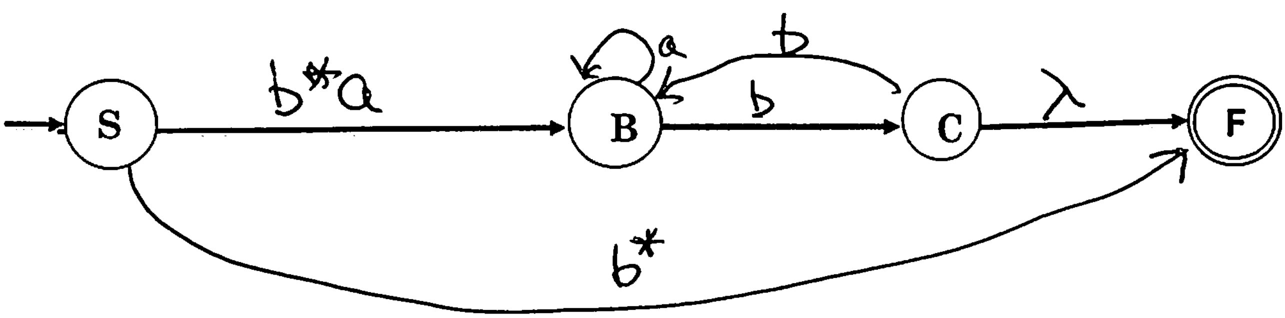
What expression does $R^4_{3,3}$ evaluate to, given that you have all the component values?

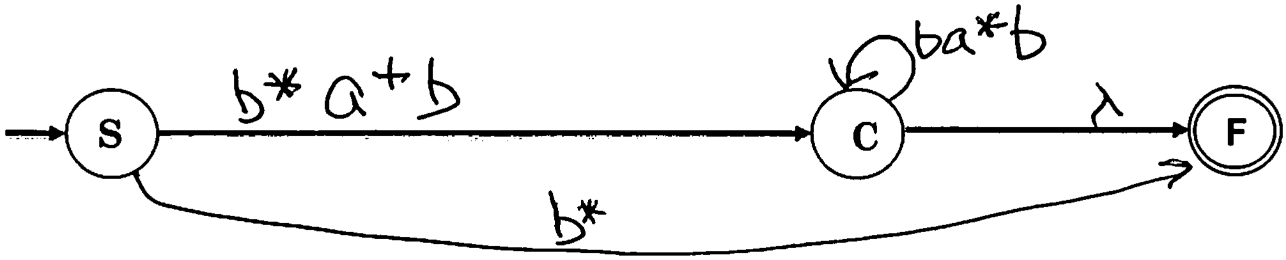
$R^4_{33} = a^* + b\,(a^*b)^* a^+$

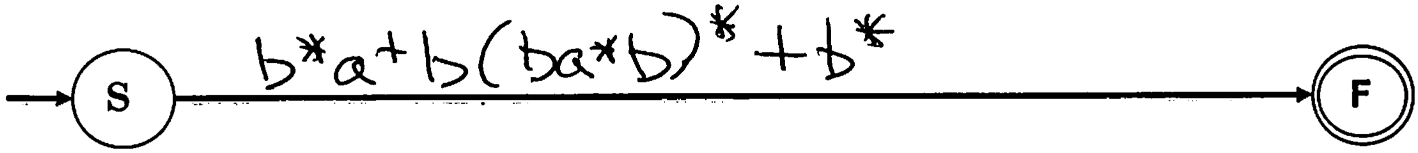
6 **4.** Let L be defined as the language accepted by the NFA $\mathcal{a}$:



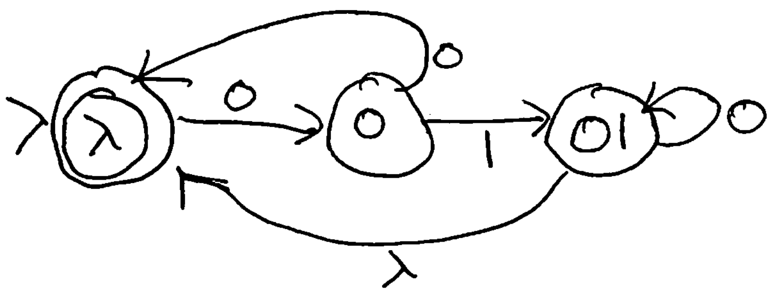
Using the technique of replacing transition letters by regular expressions and then ripping states from a GNFA to create new expressions, develop the regular expression associated with the automaton $\mathcal{a}$ that generates L. I have included the states of GNFA's associated with removing states A, 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.







3 **5.** Consider the regular expression **R = (0 0 + 0 1 0*)***
The only **+** used here stands for union. Show a **NFA** (do by transition diagram) that accepts **R**.



4 **6.** **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 $x_i \in \Sigma$

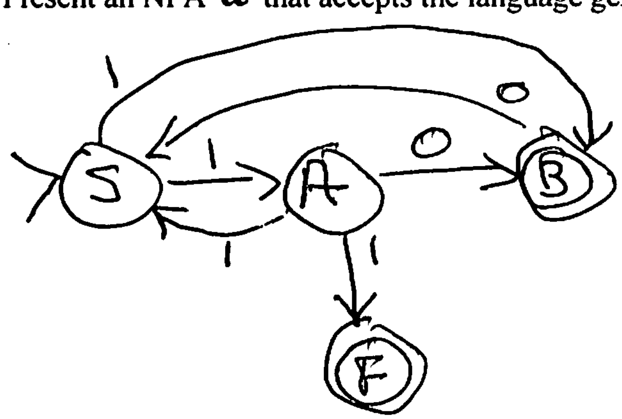
Show that any class of languages (in particular the Regular Languages) that is closed under substitution, concatenation and intersection is also closed under **OddLetters**. A constructive solution is all I ask; no proof required. I'll help. Consider using substitutions
f(a) = {a, a'}; g(a) = a'; h(a)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

$$\mathrm{OddLetters}(L) = h\left(f(L) \cap (\Sigma \cdot g(\Sigma))^* (\Sigma \cup \Sigma \cdot g(\Sigma))\right)$$

4 **7.** 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 **8.** Apply the Pumping Lemma to show the following 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.

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

PL: Gives you a value of N>0.

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

PL: $xyz = a^N b^{N+1}$, $|xy| \le N$, $|y| > 0$

$\ni \forall i$, $xy^iz \in L$, $i \geq 0$

ME: $i = 2$

$a^{N+|y|} b^{N+1} \in L$ BY P.L.

BUT $N+|y| \geq N+1$, where $|y| > 0$

So $a^{N+|y|} b^{N+1} \notin L$

↯

So L IS NOT REGULAR

4 **9.** 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:

$\mathbf{x\ R_L\ y}$ if and only if [$\forall \mathbf{z} \in \{\mathbf{a,b}\}^*$, $\mathbf{xz} \in \mathbf{L}$ 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 one another.

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

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

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

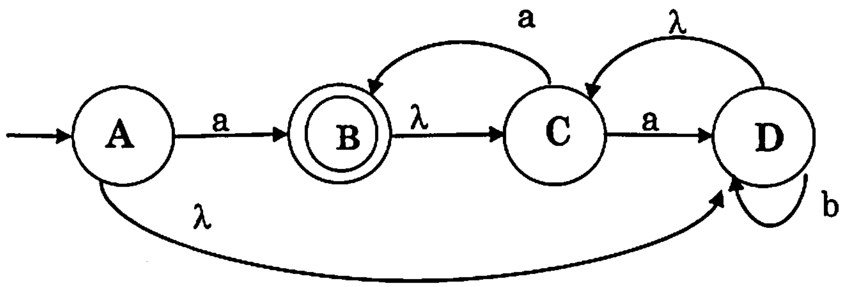
So $[a^i] \neq [a^j]$ $j > i$

AND HENCE EACH SUCH CLASS DIFFERS

AND THERE ARE AN INFINITE NUMBER OF SUCH CLASSES

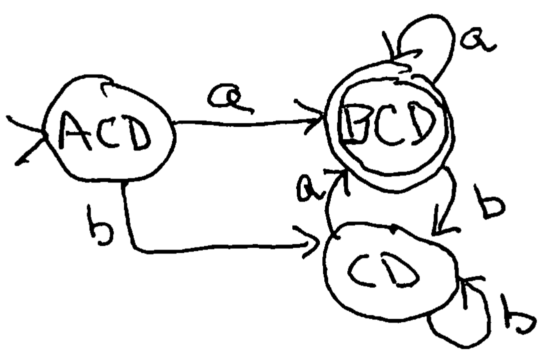
10. Let L be defined as the language accepted by the finite state automaton $\mathcal{A}$:

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

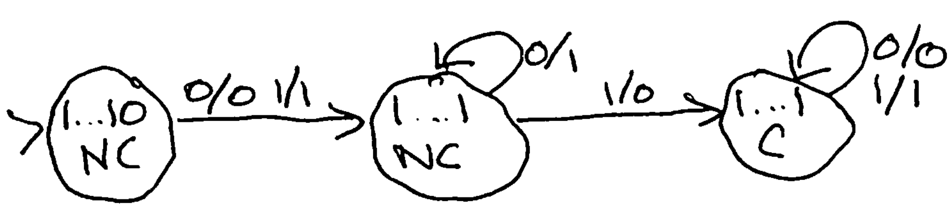


State	**A**	**B**	**C**	**D**
λ-closure	A, C, D	B, C	C	C, D

4 **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.



3 **11.** Present a Mealy Model finite state machine that reads an input **x** ∈ **{0, 1}*** and produces the binary number that represents the result of adding the twos complement representation of decimal **-2**, that is adding binary **1...1110** to **x** (this assumes all numbers are in two's complement notation, including results). Assume that **x** is read starting with its least significant digit.
Examples: **00010 → 00000; 11001 → 11010; 00001 → 11111**



7 **12.** 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
3	2	3	3
4	1	3	5
5	1	3	6
6	1	3	6

	1	2	3	4	5
2	1,4				
3	X	X			
4	3,6	3,6	X		
5	1,4 3,6 5,6X	3,6 5,6X	X	5,6X	
6	X	X	1,2	X	X

Don't forget to construct and write down your new, equivalent automaton!! 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** and **3** are indistinguishable, label the merged state as **13**.

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