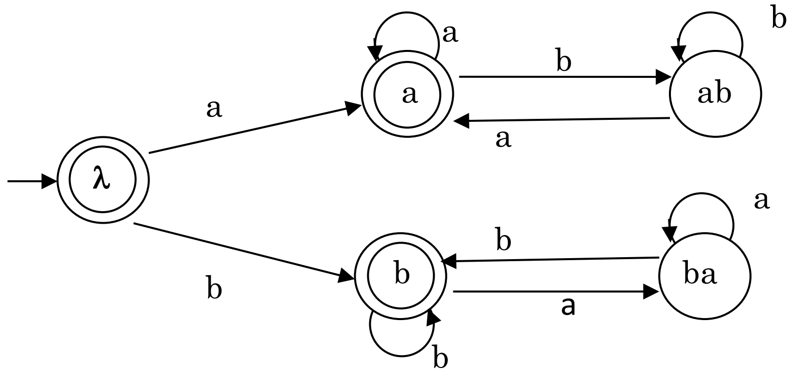
COT 4210 Fall 2014 **Sample Problems Key**

1. Draw a DFA to recognize the set of strings over **{a,b}***that contain the same number of occurrences of the substring **ab** as of the substring **ba**.



2. Present the transition diagram or table for a DFA that accepts the regular set denoted by the expression **(0+1)* (010 + 11) (0 + 1)***

	0	**1**
<λ>	<0>	<1>
<0>	<0>	<01>
<01>	<010+11>	<010+11>
<1>	<0>	<010+11>
<010+11>	<010+11>	<010+11>

3. Consider the following assertion:

Let **R** be a regular language, then any set **S**, such that $\mathbf{S \cup R = S}$, is also regular.

State whether you believe this statement to be True or False by circling your answer.

~~**TRUE**~~ **FALSE**

If you believe that this assertion is True, present a convincing argument (not formal proof) to back up your conjecture. If you believe that it is False, present a counterexample using known regular and non-regular languages, **R** and **S**, respectively.

Let **R** = ϕ and $\mathbf{S = a^nb^n}$; $\mathbf{S \cup R = a^nb^n = S}$, but S is not regular.

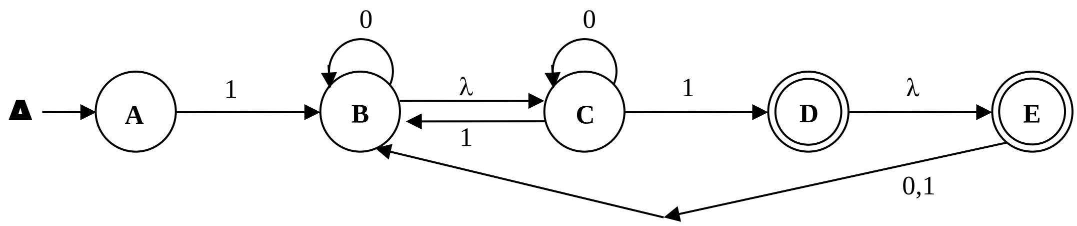
4. Assume that $\mathbf{L_1}$ and $\mathbf{L_1 \cap L_2}$ are both regular languages. Is $\mathbf{L_2}$ necessarily a regular language? If so, prove this, otherwise show that $\mathbf{L_2}$ could either be regular or non-regular.

$\mathbf{L_2}$ could either be regular or non-regular

$\mathbf{L_2}$ regular: $\mathbf{L_1} = \phi$; $\mathbf{L_2} = \phi$; $\mathbf{L_1 \cap L_2} = \phi$

$\mathbf{L_2}$ non-regular: $\mathbf{L_1} = \phi$; $\mathbf{L_2} = \mathbf{a^nb^n}$; $\mathbf{L_1 \cap L_2} = \phi$

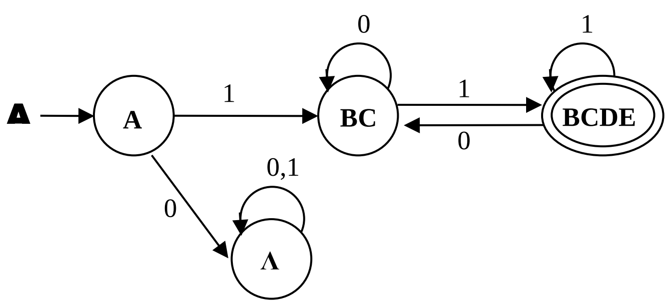
5. Let **L** be defined as the language accepted by the finite state automaton **A**:



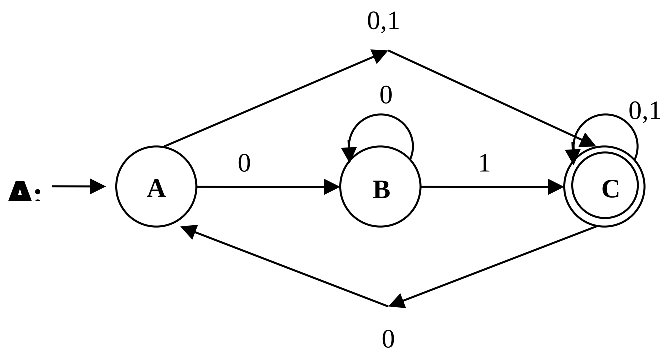
a.) Fill in the following table, showing the λ-closures for each of **A**'s states.

State	**A**	**B**	**C**	**D**	**E**
λ-closure	**{ A }**	**{ B , C }**	**{ C }**	**{ D, E }**	**{ E }**

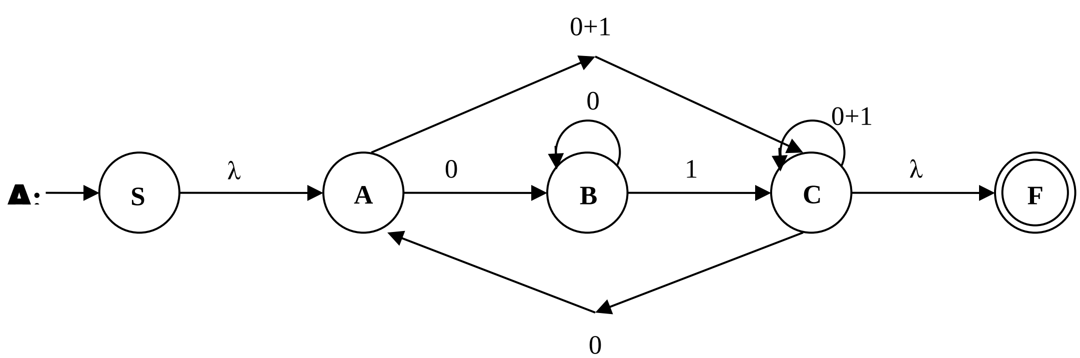
b.) Convert **A** to an equivalent deterministic finite state automaton. Use states like **AC** to denote the subset of states **{A,C}**. Be careful -- λ-closures are important.

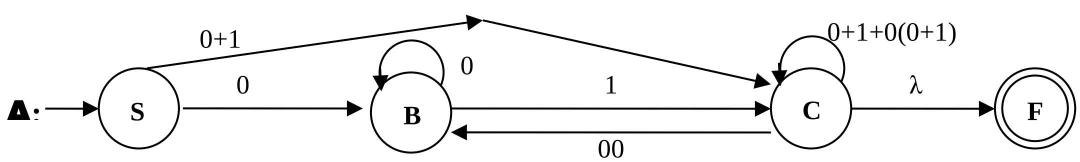


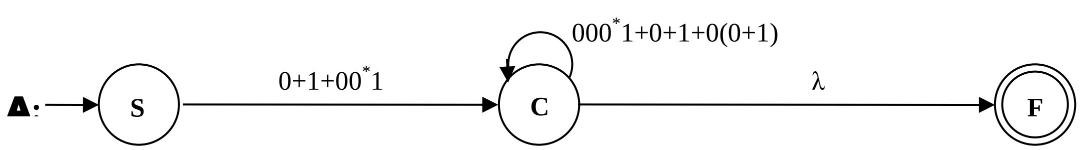
6. Let **L** be defined as the language accepted by the finite state automaton **A**:

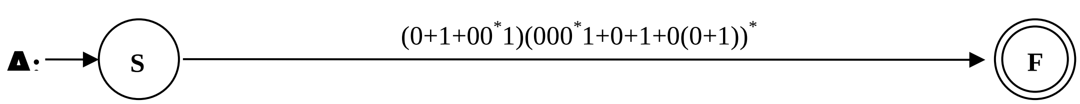


Change **A** to a GNFA (although I don't see the need for all those ϕ transitions). Now, using the technique of replacing transition letters by regular expressions and then ripping (collapsing) states, develop the regular expression associated with **A** that represents the set (language) **L**.



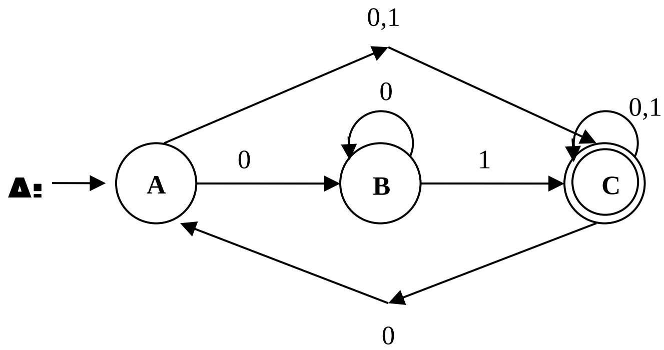






Note: $(0+1+00^*1)(000^*1+0+1+0(0+1))^* = (0+1+0^+1)(0+1)^* = (0+1)(0+1)^* + (0^+1)(0+1)^* = (0+1)^+$

7. Let **L** be defined as the language accepted by the finite state automaton **A**:



Using the technique of regular equations, develop the regular expression associated with **A** that represents the set (language) **L**.

A = λ + C0

B = A0 + B0

C = A(0+1) + B1 + C(0+1)

B = 0 + C00 + B0 = (0 + C00) 0*

C = (0+1) + C0(0+1) + (0 + C00)0*1 + C(0+1))

$= (0+1) + 0^{+}1 + C(00 + 01 + 0 + 1 + 00^{+}1)$

$= (0+1) + 0^{+}1 + C(00 + 0 + 1 + 0^{+}1)$

$= (0+1+0^{+}1)(00 + 0 + 1 + 0^{+}1)^*$

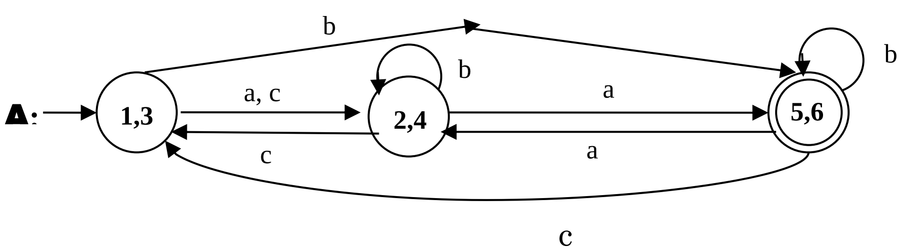
$= (0+1+0^{+}1)(0 + 1)^* = (0+1)^{+}$

8. Given a finite state automaton denoted by the transition table shown below, and assuming that **5** and **6** are final states, fill in the equivalent states matrix I have provided. Use this to create an equivalent, minimal state automaton. State **1** is the start state.

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

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

Don't forget to construct and write down your new, equivalent automaton!!



FIX

9. Use the Pumping Lemma to show that the following languages are not regular:

a.) **L = { $a^n b^m c^t$ | n > m** or **n > t,** and **n, m, t** ≥ **0 }**

Let **N** be given by the P.L.
Choose **w = $a^{N+1}b^Nc^N$** Clearly **w ∈ L**
Let **w = xyz**, **|xy| ≤ N**, **|y|>0** be given by the P.L.
By P.L., **xy^iz ∈ L**, for all **i ≥ 0**
Let **i=0**, then P.L. requires that **xz ∈ L**
However, since **|xy| ≤ N**, then **y** is a non-empty string consisting of only **a**'s.
Thus, we have that **$a^{N+1-|y|} b^N c^N$ ∈ L.**
But, this is not so, since **N+1-|y| ≤ N** and hence the number of **a**'s is not greater than either of the number of **b**'s or **c**'s. This proves that L cannot be regular.

b.) **L = { $a^n b^m$ | n ≤ m,** and **n, m** ≥ **0 }**

Let **N** be given by the P.L.
Choose **w = a^Nb^N** Clearly **w ∈ L**
Let **w = xyz**, **|xy| ≤ N**, **|y|>0** be given by the P.L.
By P.L., **xy^iz ∈ L**, for all **i ≥ 0**
Let **i=2**, then P.L. requires that **xy^2z ∈ L**
However, since **|xy| ≤ N**, then **y** is a non-empty string consisting of only **a**'s.
Thus, we have that **$a^{N+|y|} b^N$ ∈ L.**
But, this is not so, since **N+|y| > N** and hence the number of **a**'s is greater than the number of **b**'s. This proves that L cannot be regular.

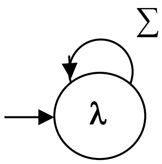
c. Let **NonPrime** = { **a^q** | **q is not a prime** }
This language is not easily shown non-regular using the Pumping Lemma. However, we already saw in class how to prove that **Prime** = { **a^p** | **p is a prime** } is non-regular using the Pumping Lemma. Explain how you could use this latter fact to show **NonPrime** is non-regular.

As regular languages are closed under complement, a demonstration that **NonPrime** is regular could be used to show **Prime** is also regular, but that is already known to be false, so **NonPrime** must itself be non-regular.

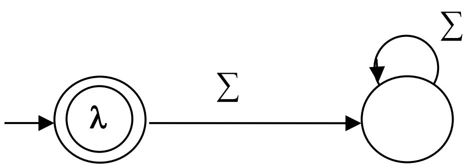
10. DFAs accept the class of regular languages. Regular expressions denote the class of regular sets. The equivalence of these is seen by a proof that every regular set is a regular language and vice versa. The first part of this, that every regular set is a regular language, can be done by first showing that the basis regular sets (**Ø , { λ } , { a | a ∈ ∑ }**) are each accepted by a DFA.

i.) Demonstrate a DFA for each of the basis regular sets.

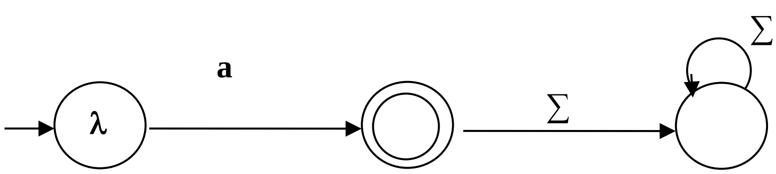
Ø



{ λ }



{ a }



Let $\mathbf{L_1}$ be generated by the DFA $\mathbf{A_1 = (Q_1 , \sum , \delta_1 , q_1 , F_1)}$ and $\mathbf{L_2}$ be generated by the DFA $\mathbf{A_2 = (Q_2 , \sum , \delta_2 , q_2 , F_2)}$.

ii.) Present a construction from $\mathbf{A_1}$, $\mathbf{A_2}$ that produces a DFA $\mathbf{A_3}$ for $\mathbf{L_1 \cup L_2}$.
Hint: Cross product.

$\mathbf{A_3 = (Q_1 \times Q_2 , \sum , \delta_3 , < q_1,q_2 >, F_3)}$.

$\mathbf{\delta_3 (<q,r>, a) = <\delta_1 (q, a) , \delta_2 (r, a)}$ > where q ∈ $\mathbf{Q_1}$; **r** ∈ $\mathbf{Q_2}$; and **a** ∈ **Σ**

$\mathbf{F_3 = F_1 \times Q_2 \cup Q_1 \times F_2}$

Clearly δ_3*(< $\mathbf{q_1,q_2}$ >, **w**) = < δ_1*($\mathbf{q_1}$, **w**), δ_2*($\mathbf{q_2}$, **w**) >

And thus **w** ∈ $\mathbf{L_3}$, just in case **w** ∈ $\mathbf{L_1}$ or **w** ∈ $\mathbf{L_2}$

iii.) What remains to be done to show that every regular set is a regular language? Don't do the proof, just state what two steps still needs to be done.

Present a construction from $\mathbf{A_1}$ and $\mathbf{A_2}$ that produces a DFA for $\mathbf{L_1 \, L_2}$. This is generally not done directly but by first showing NFAs are equivalent to DFAs and then producing a NFA for $\mathbf{L_1 \, L_2}$.

Present a construction from $\mathbf{A_1}$ that produces a DFA for $\mathbf{L_1}$*. This is generally not done directly but by first showing NFAs are equivalent to DFAs and then producing a NFA for $\mathbf{L_1}$*.