(1) (a) Design an NFA $N$ with only 5 arcs such that $L(N)$ equals the language of strings that end in $aba$. Count a single arc with a choice of two options as two arcs.

(b) Then show the conversion of $N$ into an equivalent DFA, following the method in class. Compare the number of arcs and states that you get between the two. (6 + 12 = 18 pts.)

Solution: The creation of the NFA was rather simple. Notice the DFA has the same number of states as the NFA but the DFA has more arcs. This makes sense in general; the NFA to DFA conversion uses the powerset of the NFA states thus for $k$ NFA states you might have $2^k$ DFA states and $2^{k+1}$ arcs for a 2-letter alphabet.
(2) Convert the following NFA $N$ with $\epsilon$-transitions into an equivalent DFA. Now talk about the increase in arcs and states. (18 pts.)

Solution: This time we included an epsilon ($\epsilon$) arrow so our start state was the epsilon closure of state 1 (e.g. where can you get by following along zero or more epsilon arrows/arcs): (1,2). Notice that the epsilon closure for the transition only applies when you LAND on the state, not when you start from it. So the transition in the DFA if you were in JUST state 1 on input $b$ would be the dead state. On the other hand, being in JUST state 4 on input $b$ let’s you travel to state 1 where you can determine the epsilon closure of the state you just landed in. As a mathematical example let $\delta'$ be the DFA transition, $\delta$ be the NFA transition, $E$ be the epsilon closure of a set of states,$R$ be a set of states from the NFA (each state in the DFA is a set of states from the NFA) and $Q$ be the (set of) states of the NFA:

$$E(R) = \{ q | q \text{ can be reached from } R \text{ by traveling along 0 or more } \epsilon \text{ arrows} \}$$

$$\delta'(R, a) = \{ q \in Q | q \in E(\delta(r, a)) \text{ for some } r \in R \}$$

$$\delta'(\{4\}, b) = \{ q \in Q | q \in E(\delta(r, b)) \text{ for some } r \in \{4\} \}$$

$$E(\delta(\{4\}, b)) = E(\{1\}) = \{1, 2\}$$

Per the definition of $\delta'$ we have that $\delta'(\{4\}, b)$ is the set of states reachable along 0 or more epsilon arcs once transition in the NFA occurred. In this case we only had 1 state, 4, so that set of states is just $\{1, 2\}$.

In terms of the increase in states notice that this time the DFA does have more states than the NFA. Notice that the number of transitions also increased.
(3) Find regular expressions for the languages of the following two FAs. Show the use of the algorithm from class, and then use "sight-reading" to check your work. (Note that arcs going to a dead state are not shown, which is fine if you consider these to be NFAs. 18 pts. For 54 total on the set)

**Solution:** We do the "sight reading" technique as the explanation and diagrams are a bit more intuitive. Here the first conversion is somewhat trivial: you may have any number of $b$'s in state 2, but you need an $a$ to get there from state 1 and an $a$ to leave. You could have also had a single $b$ to get to state 3. Putting it all together: $(ab^*a \cup b)(a(ab^*a \cup b))^*$.

The second conversion is a bit tricky when you do the third and fourth state collapse. Starting from state 1 you need at least a $b$ followed by an $a$ to get to the accept state. To stay in the accept state you need at least a $b$ followed by an $a$. Putting it all together: $(ab)^*(ba)^*$