Regular Languages

1. Draw out DFAs (or NFAs) for each of the following two languages. Do these by making smaller machines for each of the two parts of each language, and then combining them into a single machine.

   - $\{w|w$ has at least three a's and at least two b's$\}$
   - $\{w|w$ has an even number of a's and exactly one or two b's$\}$

   Both of these will involve a square grid of states. On one access we will have one condition (how many a's) and on the other access we have the other condition (how many b's). Each state represents the combination of reading some number of a's and some number of b's (e.g., we have seen 2 a's and 1 b so far). The final states are the ones that meet the requirements of acceptance (sorry for not providing a picture here).

2. For any string $w = w_1w_2, ..., w_n$, let w^R be the reverse of string w (i.e., $w^R = w_n, ..., w_2, w_1$). Prove that if a language A is regular, then the language $A^R = \{w^R|w \in A\}$ is also regular.

   Assume A is regular, this means a DFA D exists that accepts A. We can construct a DFA D' from D that accepts A^R as follows: Start with D and flip the direction of all transitions. Make the start state of D the final accepting state of D'. Make the accepting states of D normal states. Add a new dummy start state to D' and epsilon transition from this dummy start state to all of the original accepting states of D. This machine will accept A^R because it simulates D backwards, starting at all final states (using non-determinism) and working in reverse to the original start state of D.

3. Use the pumping lemma to show that the following language is not a regular language: $A_1 = \{0^n1^n2^n|n \geq 0\}$

   Assume for sake of contradiction that A_1 is regular. Thus, pumping lemma applies. Consider the string $s' = 0^p1^p2^p$ where p is the pumping length. Condition 3 of the lemma states that a substring of the first p character can be pumped. This would involve pumping only 0s, which would make the number of 0's not match the number of 1's and 2's, creating a contradiction.

4. Find and describe the error that exists in the following proof. The proof attempts to show that 0^*1^* is not regular, when in fact it is:

   Assume, for sake of contradiction, that 0^*1^* is regular. We select an element from this language that is greater than the pumping length p. We select 0^p1^p. In class, when proving that 0^n1^n was not regular, we showed that 0^p1^p cannot be pumped. Therefore, 0^*1^* is not regular.

   The issue is that 0^n1^n has to have the exact same number of 0s and 1s, but 0^*1^* does not. So 0^p1^p CAN be pumped. Pumping the 0s will make the string have more 0s than 1s which is just fine.