Reflection Assignment #1

A1: Groupings of Equivalence {a1dfa1.jff, a2dfa2.jff, a1dfa4.jff} {a1dfa3.jff} {a1dfa5.jff}

When I began constructing the DFAs I was honestly overwhelmed and felt like it was a lot. However after a couple tries of constructing the first DFA I soon realized it wasn’t as bad as I thought. When I got to the second DFA I realized quickly that it was related the first DFA I made. The DFA ended up being the exact same as the first DFA. The third DFA was a little bit trickier and took more thought. For DFAs that get more complicated I really had to make sure a 1 and 0 were leaving each state. By this point I think I had the hang of creating DFAs and making the final one came together rather quickly.

A2: What I observed when converting the NFA to the DFA is there can be a lot of useless information in the NFA. While the NFA seems more complex, the actual DFA only needed two states to complete. I also realized in this DFA there is no trap state like there was in A1.

A3: When I was minimizing the DFA the first thing I noticed was the useless states on the right hand side. So just like the problem above there was some useless information and once I was able to get everything else straight, the DFA creation process went quickly. The biggest problem I have is I forget to make sure each state has two outputs coming out of each state.

A4a: (xyx, xxy, y, xy)

A4b: (yxy, xyy, yy, xx)

A4c: I noticed that the accepted strings were eight xy, two x’s followed by a y or y by itself. When there were two xs in succession without anything after it was rejected as well as two y’s in a row. So you couldn’t have more than two x’s or y’s in a row.

A5: I actually got a lot of help from classmates in order to finish this one. I had an idea of what to do but I had trouble figuring it all out.