

Samuel Leonard

CS 461

Project Assignment 1

Convert NFA to DFA

I found this project to be a little more difficult than anticipated. The problem required an input coming from standard in representing the transition table of an NFA being parsed and stored, then processed and put into a DFA transition table. There were three algorithms that needed to be created along with the actual reading in and parsing of the input. The first was the e-closure function, then the subset constructor and a simple move from these states on this input function.

I initially chose to write this in python because of the ease with which strings can be sliced up into fields making it easier to parse the input. However, due to my very limited knowledge of python this became very time consuming and I began writing the program in c++. I struggled with parsing the input like I wanted and eventually went back to python.

In order to solve this problem, I took the write out for the program and broke it up into sections. I could see that the first order of business was to read in the input and create some sets and a data structure to hold the actual NFA states and their edges. The sets were for the initial state, total states, etc. To be sure this was working, I printed all the states and their edges, and the sets for the NFA state information. Then with each algorithm, I carefully constructed it line by line, testing each block created. For example, when building the e-closure segment, if something went on the stack, I printed the stack, if something was popped, I printed that object or value. I would check each edge and then I printed the e-closure result before building the subset constructor.

As I built the subset constructor, I also tested line by line. I kept a print statement to watch as the dStates filled. I watched the return of the e-closure within the function. Everything was going smoothly until I tried to deal with marking the states in the dStates when they had completed going through their inputs. Something just wasn't right, I had an endless loop. I created a watch for when each state was coming in and when the states were getting marked and the list was growing faster than it could be marked until it was exponential. I found that there were empty subsets being created, so I dealt with those and the smaller NFA inputs ran close to as expected, but the third would not complete. I could not find the error and decided to write this up and turn everything in. I built a timeout feature to handle big input files and I only got to build the corresponding DFA. The program works for small input files but due to my lack of knowledge with python, large files are running big $O(N^2)$.

I hope to see a result so I can find out where I went wrong. The assignment however was enjoyable and frustrating all at the same time.