Gregory Fox

Professor Rivas

CMPT 440

1 February 2016

Create a DFA

1. The assignment was originally difficult to understand because the DFA diagram does not represent the actual purpose of the function, but the way that it is written. This means that the DFA diagram needs to correctly represent the syntax of the program and not the output. Once I moved past that level of understanding, showing that any character can be represented was difficult in the DMA diagram. I ended up going with a literal attempt to show most of the characters that would be used like "a-z" or "0-9". It was also important to show that if there were any illegal characters entered at certain steps in the function that an error would occur. This assignment gave me a better understanding of how to represent Java functions in a Lucid charts. Some of the problems I had with Lucid charts were the arrows used to represent loopbacks are difficult to use, and showing large amounts of text as a transition can make the chart unorganized very quickly. Lucid charts is generally pretty intuitive to use, also exporting the file as a pdf was easy because lucid charts already has a feature to do that.

2. To implement this in Java I would use a series of If and Else statements to constantly check the characters be entered. This would end up being very difficult because the program would constantly be returning errors if any character was entered except the singular character that is allowed by the program. Passing If statements in Java is represented in the DFA diagram as the chosen paths to each step. This would require several functions in Java that would constantly return a result of whether there was an error or the correct character was entered and the next if statement will then start. There would need to be a part of the function that would allow the user to re-enter a character and then also give them some type of information as to what character the program was looking for. The DMA diagram below represents any character by showing a majority of the common symbols that would be entered by a user interacting with the program. Each of the transition arrows shows the character that is needed to progress to the next stage without getting an error. The qError bubble is used to show that the program terminates if an incorrect character is entered.

