Alexander Goncalves
Professor Rivas
CMPT440
2/4/2017

DFA Terminal Application

My project idea is to implement a simple command line interface using the DFA algorithm. I intend to create this project in Java with UI implementation using either the Swing or JavaFX framework. My inspiration for this project came from my familiarity with the Linux terminal, the Windows command line, and just command-based languages in general. In addition to this, while learning about how a DFA works, my textbook specifically cites such command-based implementations as a software project that uses the DFA algorithm.

My general idea is to have a program that parses a buffer, which should contain input from a user, and then attempt to make sense of it. In addition to this, I will need to create a simple instruction set of commands, for the user to actually make use of the command line. Many of these commands will be inspired by those of the Linux/Unix variety, in particular commands such “touch” and “cat” come to mind. On the other hand, I will also try to create completely original commands, an arithmetic program using an expression tree comes to mind. Some commands may take options to specify output, some may require variable inputs as well. All of this will be taken into consideration with the DFA implementation.

Taking a step back, the user’s input will be put in a buffer and then separated by whitespace, with the intent of mimic-ing how a tokenizer works. The initial state of the DFA represents not having a command, which is always the starting state. The next state should be a representation of having a valid command-name, without a correct command name there isn’t anything to run. Once there’s a correct command the possible states should branch with several options: a command name + option + vars, command name + option, or command name + vars. The states are valid depending on the command, but each one will produce results with minor variations, ultimately producing some result visually in the CLI representation.