1

Encoding a DFA

The DFA is encoded in a JSON file this was particularly done to increase the amount of structures that can be represented including array and the object key/value pairs. Further reading the JSON file is automated through the python JSON library. Test values can also be inserted into the JSON array to parse through the DFA. The JSON object format allows to describe the DFA transition function with much ease compared to other formats, for example describing it using other languages would make it difficult to read the file, involving more lines of code. Using the JSON object the key/value pairs abstract the process and are within the rules of the program.

2

Reading the DFA

The DFA is read through the python json library. They are read to their respective notation representing the DFA object. For example using the key, "states", all the states are read into the states array. Once the file containing the representation of the DFA is read in tests are made to compare the strings to the alphabet and parse the characters of the string through the transition (delta) function.

3

Processing a String

The string is processed through two main functions, the first function compares the string by parsing the characters to be compared against the alphabet, any discrepancies lead to the function returning a false '0' value which at that point the program will cease execution. The second function parses the characters in the string using the transition function. The state changes as the string is parsed through the transistion function. Once the final state is obtained it is compared to the list of final states according to the DFA. The designed DFA files are consist of, a DFA that recognizes all binary numbers that are multiples of 5 and have test cases that include empty strings, and other alphabets. The other DFA consists of all strings of consisting of an even number of zeros. Further more strings are checked to evaluate the DFA and the program itself. The comments in the program are consistent with the layout of the program. The functions described can be combined

form continued on next page…

into one function however the division of the program provides consistency. The design decisions made were alphabets were limited to binary and that every state has a transition based on the alphabet, a complete DFA. This may not always be the case and may be adjusted for in the JSON file. These considerations were not taken into effect when testing however this does not limit the program, and should not break.