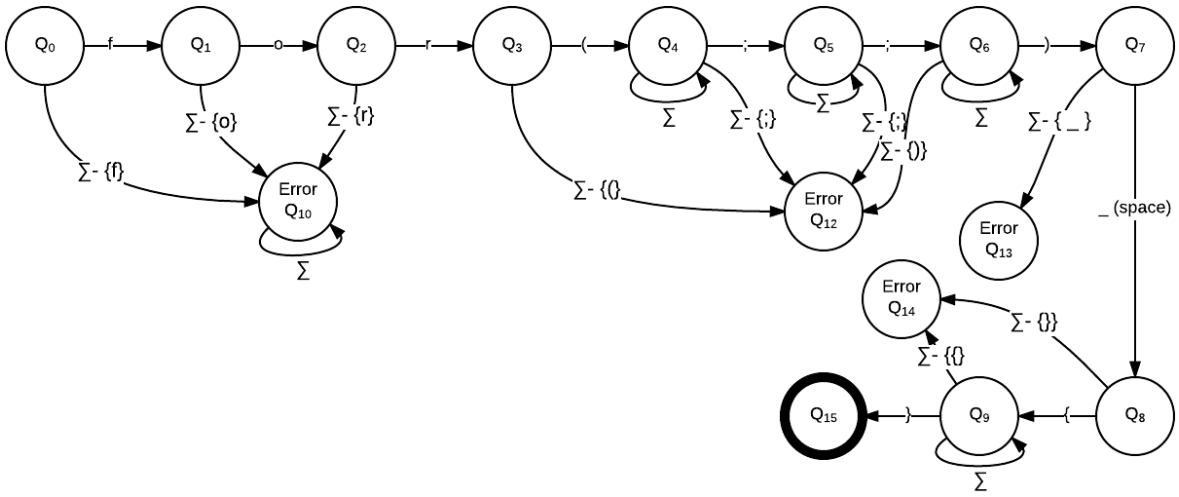
Michael Pair

Formal Languages and Computability

February 2, 2016

Professor Rivas



2. When I first started making the diagram, it was easy, adding a circle for each letter of “for,” but once I got the “int = 1;” it threw me off because I was trying to figure out how much detail I needed to go into in order to make that work. I was deciding if I needed a circle for each letter of “int” or use ∑ (an alphabet) to represent the potential to put in int. I then realized that the DFA diagram needed to work for another for loop. This other for loop was a very generic for loop that had no elements in it. Despite the missing elements, the for loop was still using proper syntax, it just didn’t have anything inside of it. This made me realize that it was just a shell of a for loop and that I could use it as a base model for my for loop DFA Diagram. Once I made the diagram, I used the inital for loop as a kind of test to see if it passed the DFA diagram and it did pass!

3. I would implement the DFA into java to validate functions called in Java. I would take the function and run each part of it through a series of test cases in order to make sure everything was working properly. Then, if everything passed the test case, I would allow the function to run, and if not, I’ll throw an error based on which part of the function did not pass the test cases. To validate a string using DFA, I would first have the DFA test for a quotation mark at the beginning. Then check all the characters to make sure each one was valid. After that it would check for a quotation mark at the end of the string. If any parts of the test failed, I would stop whatever function the string was a part of and throw an error based on which test case it failed