CSC 333 Lab 6

Samuel Jentsch

Before beginning problem 32, I practiced some with the FSM simulator. It took some time to figure out how the program worked (how to build the files for the simulator) and even how to run the program (java class files are provided that have to be executed from the command line). After some practice, I began on problem 32.

Before implementing the FSM in problem 32 in the simulator, I worked out which regular expressions the machine can recognize. I found it can recognize expressions in the form:

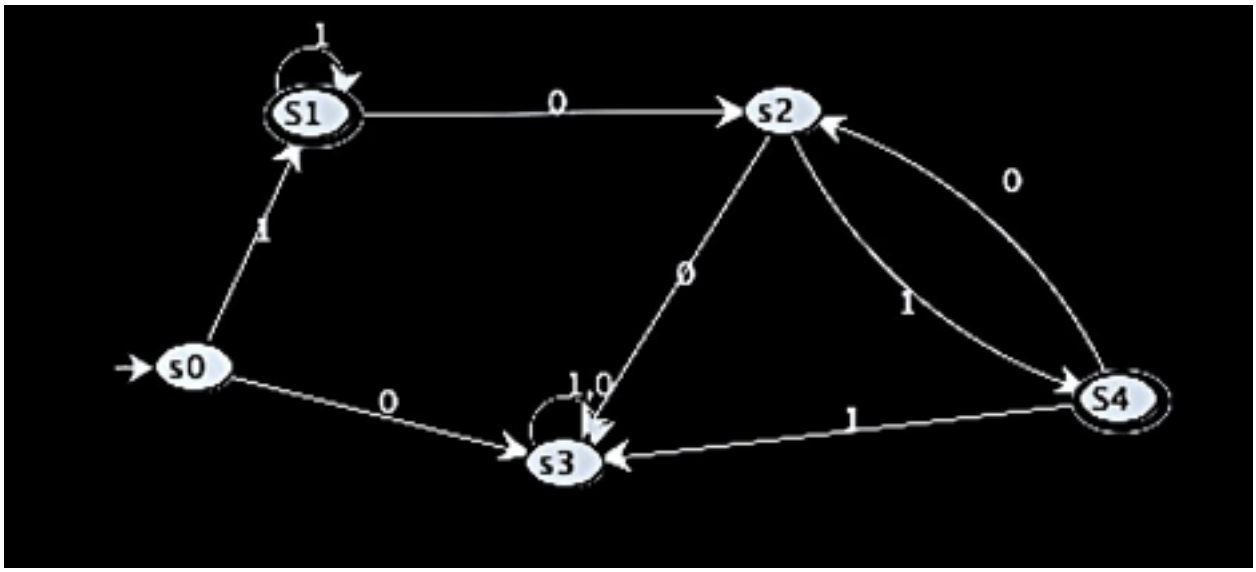
$$11^*\vee11^*01(01)^*$$

This is based on the final states in the machine and the paths that must be taken to reach the final states.

Next, I found the state table representation of the machine in problem 32 to make it simpler to implement in the FSM simulator:

	Next State		
Present State	0	1	Output
S0	S3	S1	0
S1	S2	S1	1 (Final State)
S2	S3	S4	0
S3	S3	S3	0
S4	S2	S3	1 (Final State)

After finishing the state table, I implemented the FSM in the simulator:

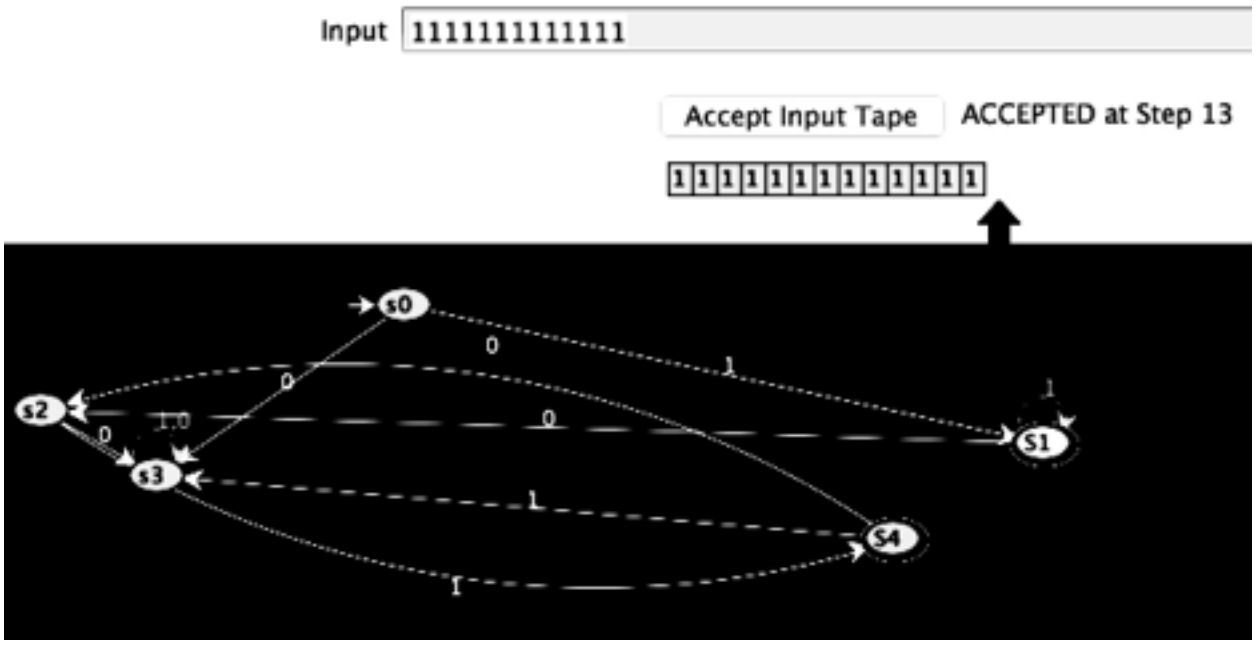


Here is the listing for the machine above:

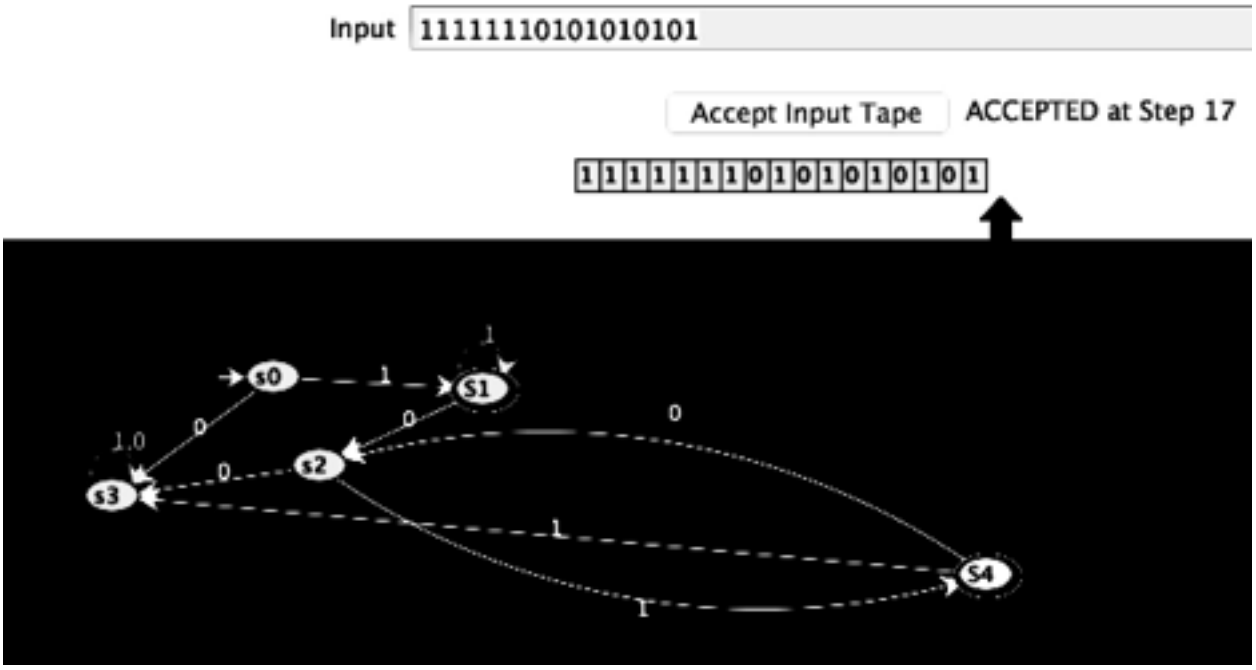
```
DFA //Type. There MUST be a space before "//" on all the lines
Sam Jentsch- 333 FSM- Project 6  //Title
0 1       // input alphabet, note such comments are permitted at the end of the line
s0 s1 s2 s3 s4 // Machine states
s0        // the initial state
s1 s4 // final states
s0  0  s3  // transitions: input state, input symbol, output state
s0  1  s1
s1  1  s1
s1  0  s2
s2  0  s3
s2  1  s4
s3  1  s3
s3  0  s3
s4  0  s2
s4  1  s3
end       //required
```

Sample Executions:

Testing regex 11* with string 111111111111:



Testing regex 11*01(01)* with string 111111110101010101:



Testing unrecognized string format with string 011111111110101: