Languages and Machines

Chap 11: 1,5,9

1.

I will explain this rather than draw it out. My solution uses 12 states and takes n or less transitions, where n is the length of the input plus the initial blank.

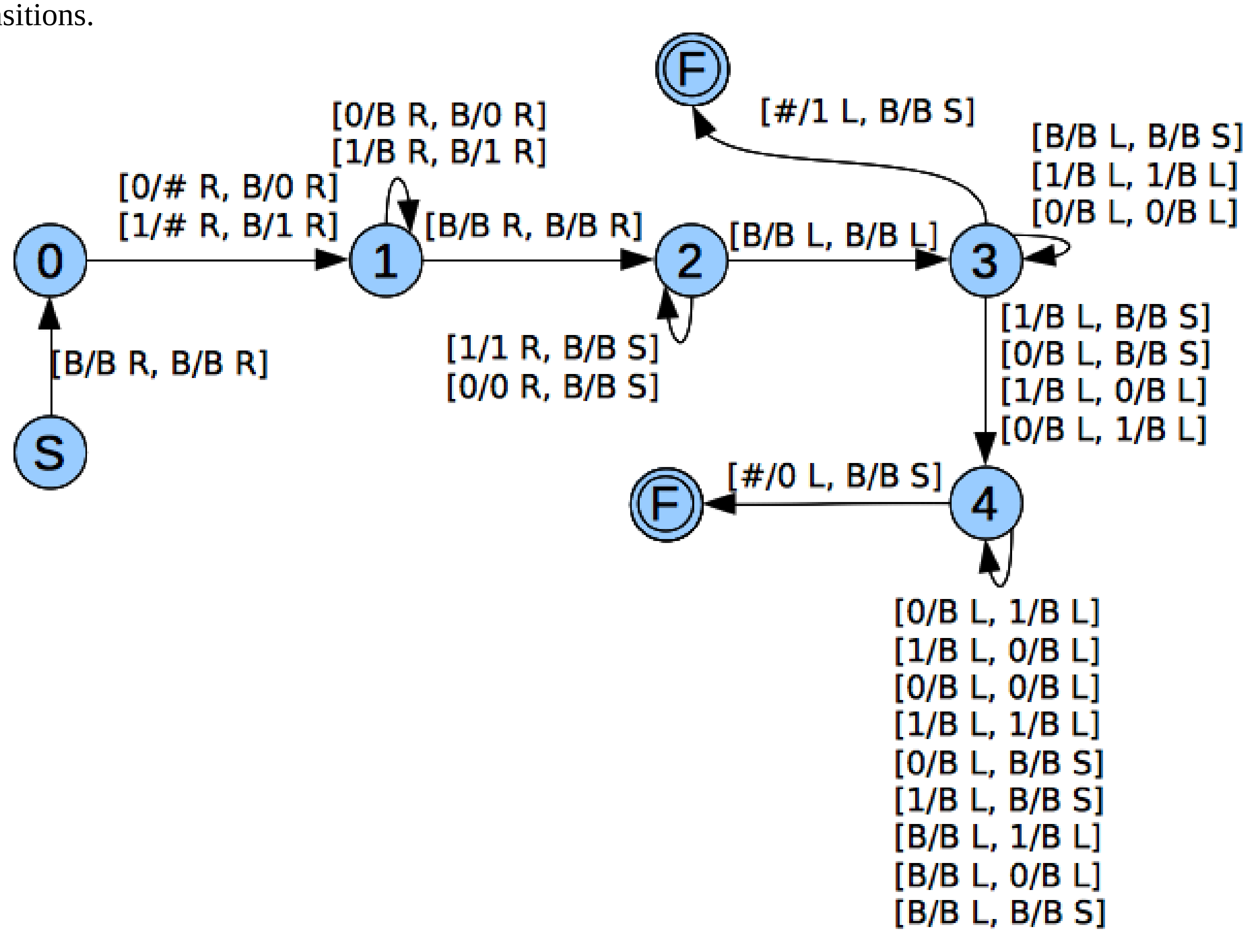
Two categories of strings are accepted: combinations of nickels and dimes and combinations of a nickel and a quarter.

We can characterize the sums of the former as 5, 10, 15, 20, and 25. The possible sums of the latter are 5, 25. We obtain our states by combining these sums as pairs and disregarding the pairs that are unreachable. Every state has deterministic transitions to the next state that corresponds to the new sum.

For example: 0,0 is the start state. 5,5 is the state reached after a nickel. 15, 5 is the state after seeing a nickel and then a dime. From now on, at any point that we see a quarter, we succeed. 20, 5 is reached after we see another nickel. And so on.

2.

Sweeps to the right, pushing the first parameter to the second tape, goes to the end of the tape, and works left, matching the second parameter to the first. This will take 4(length(u)+1) transitions assuming length(u) = length(v). I'm very curious how this could be made to take 3(length(u)+1) transitions.



5.

1. The input is checked to determine if its format is that of a representation of a directed graph followed by the encoding of two nodes. If not, M halts and rejects the string.
2. For every node n of the digraph, run the machine M' as defined in example 11.2.2 (the pathfinding machine), minus step 1, with input e(n)0e(n). If any result from these computations returns true, then halt the computation and return true.

9.

(a) {0*10}
(b)

000	
101110110111011	d(q0,B) = [q1,B,R]
00	
110101101011	d(q1,0) = [q1,0,R]
00	
110110111011011	d(q1, 1) = [q2,1,R]
00	
111010111101011	d(q2,0) = [q3, 0, R]