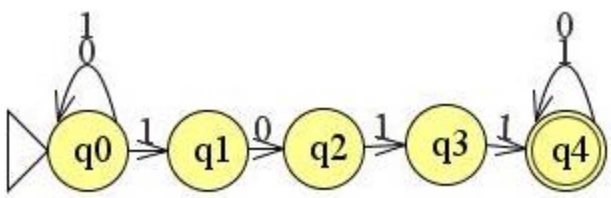
Lucy Wilcox

Assignment for Day 6

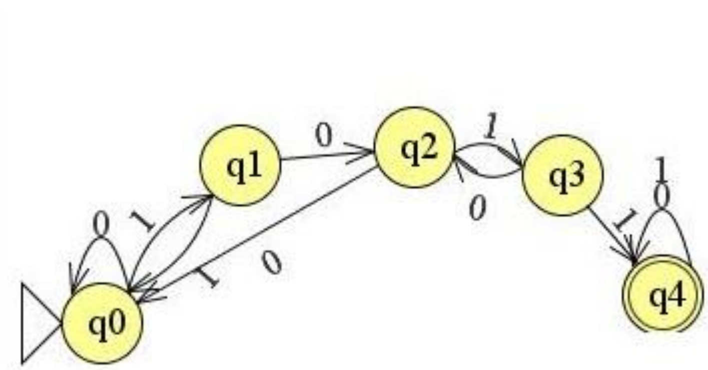
I used Sipser's book to assist me with the homework and the JFLAP program.

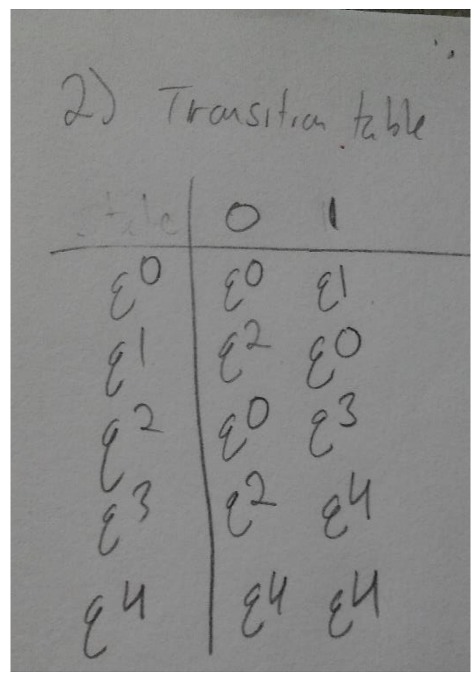
**Part 1**

**Problem 1**




**Problem 2**



2) Transition table

| State | 0 | 1 |
| --- | --- | --- |
| q0 | q0 | q1 |
| q1 | q2 | q0 |
| q2 | q0 | q3 |
| q3 | q2 | q4 |
| q4 | q4 | q4 |

**Problem 3**

I do not think that a large alphabet is more powerful than a short one.

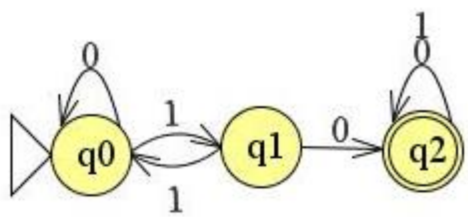
An alphabet L = {a, b, c, d} contains no more information than an alphabet N = {0, 1}.

From N, we can form the alphabet N* = {empty, 0, 1, 00, 01, 10, 11, 000, … } each element of N*. Regular languages are closed under the star operation so elements in N* are part of N. N* also contains infinite entries, therefore it contains enough information that it can compute at least the same amount as the four entries L which has. They could for example be represented as {00, 01, 10, 11}. This concept is used in the binary alphabet which encodes letters as numbers.

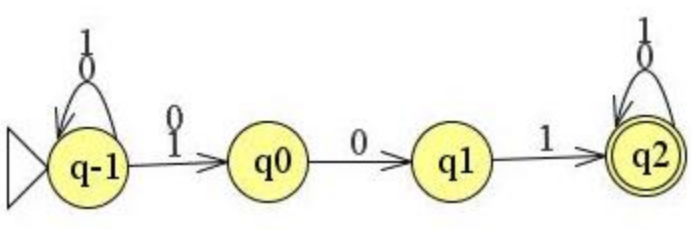
**Problem 4**

Here's my attempt:

This is A which looks for "10". Any DFA looking a string can be represented as a forward chain of transitions which are the values we are looking for, in this case q0->q1 if 1, q1->q2 if 0 , so 1 then 0. It will also have other arrows going to previous states, q's with a smaller index them themselves. And on the final state, has 0 and 1 returning to itself.




A NFA can be made which looks for the reverse by adding a new starting node which loops 0, 1 back to itself, and to the q0 of the DFA, A. Remove all arrows which go to q's with a smaller index than themselves, and perform a not operation on all other transition values (the 0s and 1s).




**Part 2**

**Problem 5**

Done

**Problem 6**

0*10*