Swetha Jayapathy

Student ID : 934041047

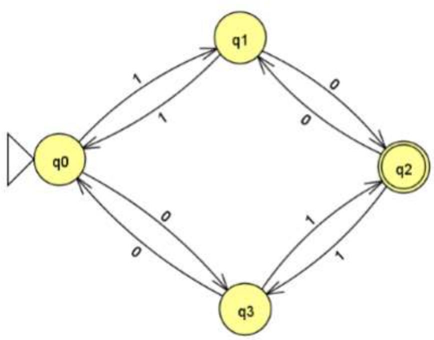
Instructor : Julianne Schutfort

CS321: Introduction to Theory of Computation

January 14th, 2020

CS 321 – Homework 1

1) (3 pts) For the DFA M below, give its formal definition as a quintuple and verbally describe the language, L(M), accepted by M.



M = (Q, ∑, δ, q0, F) where

Q = {q0, q1, q2, q3}

∑ = {0, 1}

F = {q2}

Transisiton function δ :

δ (q0,0) = q3

δ (q0,1) = q1

δ (q1,0) = q2

δ (q1,1) = q0

δ (q2,0) = q1

δ (q2,1) = q3

δ (q3,0) = q0

δ (q3,1) = q2

Transition Table :

δ	0	1
q0	q3	q1
q1	q2	q0
q2	q1	q3
q3	q0	q2

L(M) = {w ϵ {0, 1} | (10)^n+(01)^n+(01^n)+(10^n)+(1^n 0)+ (0^n 1)}

Where n is a odd number.

Description – L(M) consists of strings of 0s and 1s. Strings that drive M to a Final state include

- When we start from the initial state, q0 can take either 1 or 0. Upon taking a 1, it can take odd number of 0's to reach the final state.
- Upon taking a 0, it can take odd number of 1's to reach the final state.
- It can also take odd number of 1's and a 0 to reach a final state. Similarly, it can also take odd number of 0's and take a 1 to go to the final state.
- It can take either 10 or 01 raised to odd powers in order to reach the final state.

2) (5 pts) Let L = {w ϵ {0, 1}* such that w is a binary representation of an odd integer}. Prove that L is a regular language.

Number	**Binary Representation**
1	00001
3	00011
5	00101
7	00111
9	01001
11	01011
13	01101
15	01111
17	10001

As per the above table, all the binary representations for odd numbers end in 1, hence we can construct a DFA with strings 0s and 1s such that it goes to a final state on seeing a 1.

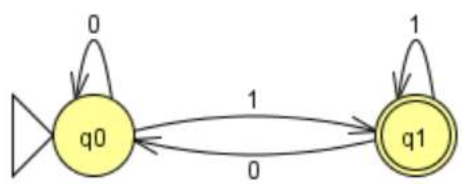
The following DFA accepts the specified Language L

M = (Q, ∑, δ, q0, F) where

Q = {q0, q1}

∑ = {0, 1}

F = {q1}



The transition function δ can be represented as below :

δ (q0,0) = q0

δ (q0,1) = q1

δ (q1,0) = q0

δ (q1,1) = q1

δ	0	1
q0	q0	q1
q1	q0	q1

If a language is accepted by DFA then it is a Regular Language. Here DFA M with L (M) proves that the L is regular.

3) (5 pts) Suppose that a bank only permits passwords that are strings from the alphabet ∑= {a, b, c, d, 1, 2, 3, 4} that follow the rules:

- The length is at least five characters
- It begins with a letter {a, b, c, d}
- It ends with two digits {1, 2, 3, 4}

The set of legal passwords forms a regular language L. Construct a NFA or DFA for L.

Solution :

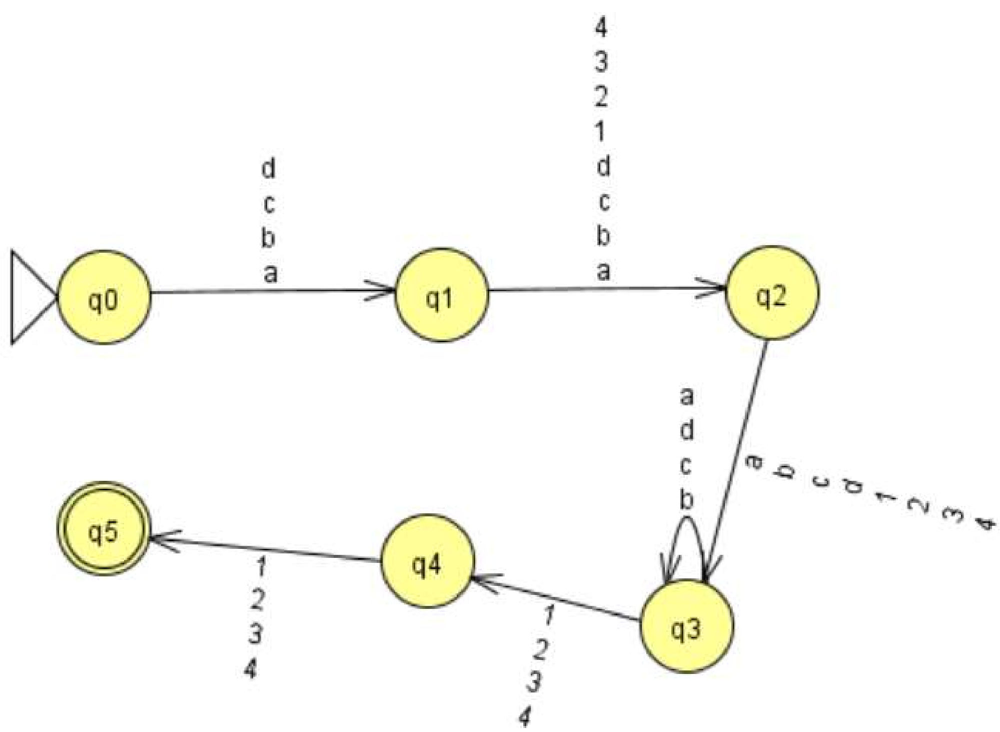
Let L be a Regular language over alphabet ∑. Since it is Regular, there exists a NFA

M = (Q, ∑, δ, q0, F) which recognizes L where

Q = {q0, q1, q2,q3,q4,q5}

∑ = {a, b,c,d,1,2,3,4}

F = {q5}



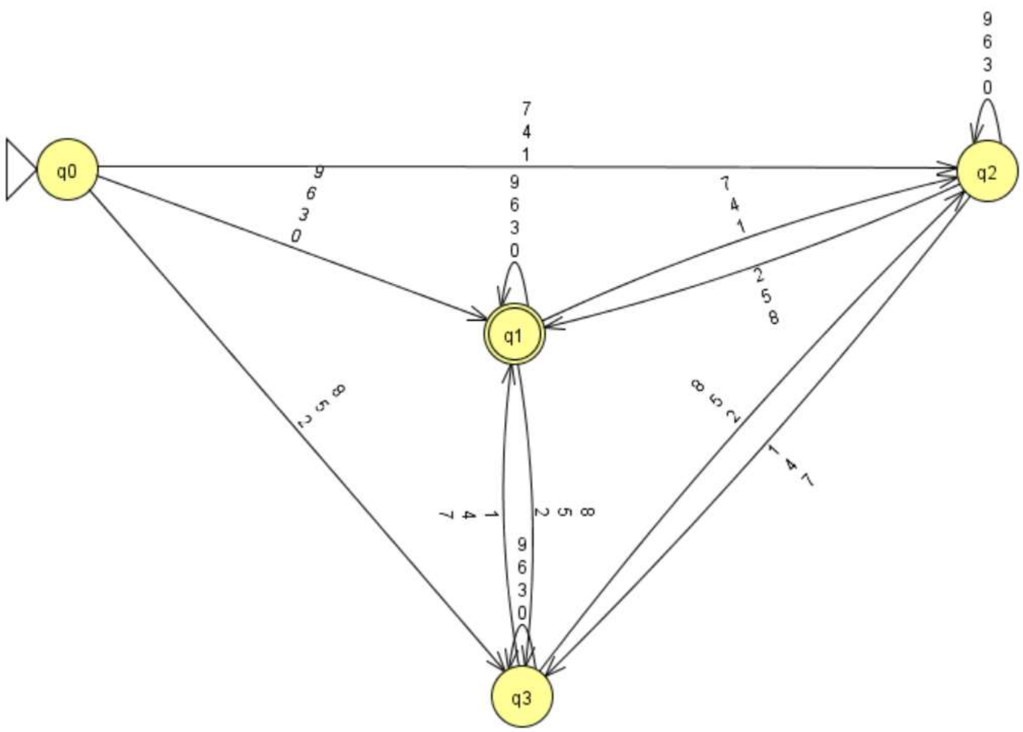
Transition Function is

δ	a	b	c	d	1	2	3	4
q0	q1	q1	q1	q1	ϕ	ϕ	ϕ	ϕ
q1	q2	q2	q2	q2	q2	q2	q2	q2
q2	q3	q3	q3	q3	q3	q3	q3	q3
q3	q3	q3	q3	q3	q4	q4	q4	q4
q4	ϕ	ϕ	ϕ	ϕ	q5	q5	q5	q5
q5	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

4) (7 pts) A number is divisible by 3 if the sum of its digits is divisible by 3.

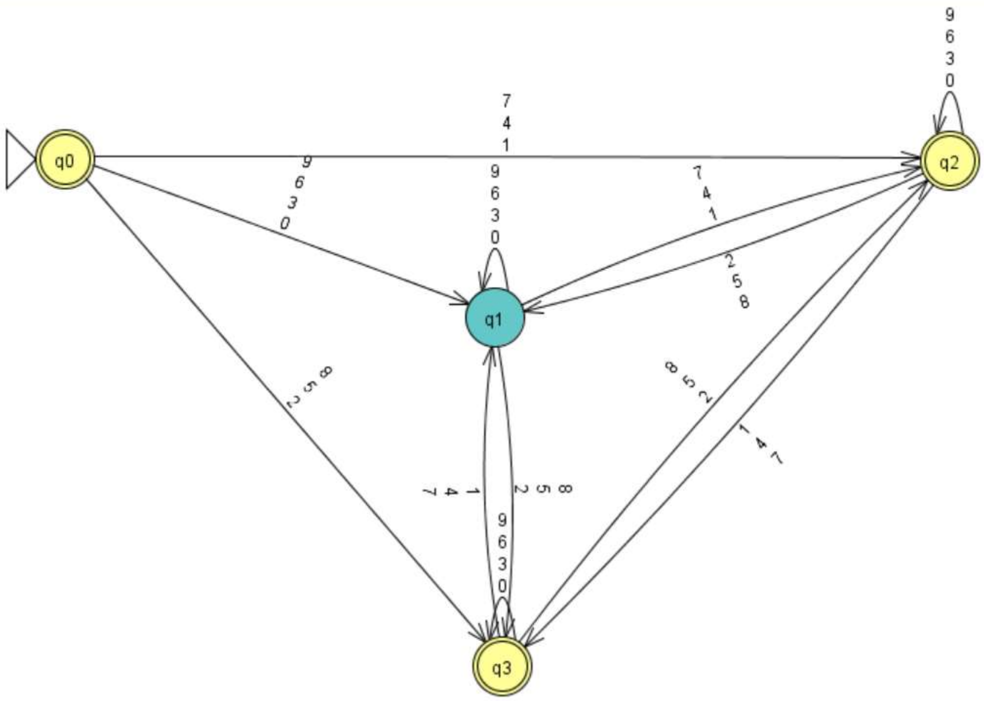
a) Construct a DFA M that accepts a base-10 number if it is divisible by 3. That is

L(M) = { w ϵ {0, 1,2, 3, 4, 5, 6, 7, 8, 9}* : w mod 3 = 0 }



b) Construct a DFA $\overline{M}$ that accepts numbers that are not divisible by 3. That is

L($\overline{M}$) = { w ϵ {0, 1,2, 3,4 , 5, 6, 7, 8, 9}* : w mod 3 not equals 0 }



5) (5 pts) Prove that the class of regular languages is closed under complementation. That is if L is a regular language then $\overline{L}$ is also a regular language. Hint: Use the DFA M that recognizes L to construct a DFA $\overline{M}$ that recognizes $\overline{L}$.

Proof :

Let L be a Regular language over alphabet ∑. Since it is Regular, there exists a DFA

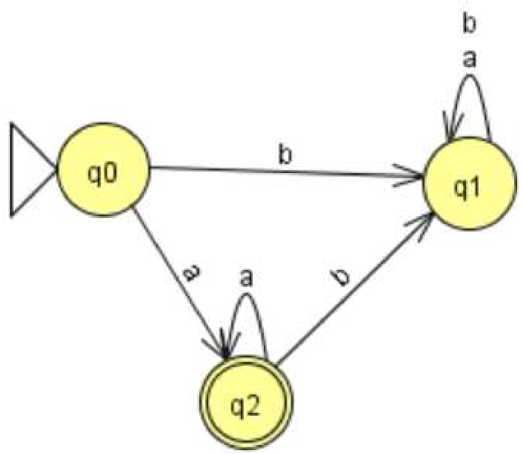
M = (Q, ∑, δ, q0, F) which recognizes L where

Q = {q0, q1, q2}

∑ = {a, b}

F = {q2}

The DFA accepts Language L(M) = {a,aa,aaa,…} that is a+



Now let us take a DFA which is complement to the above mentioned one, let L' be its complement

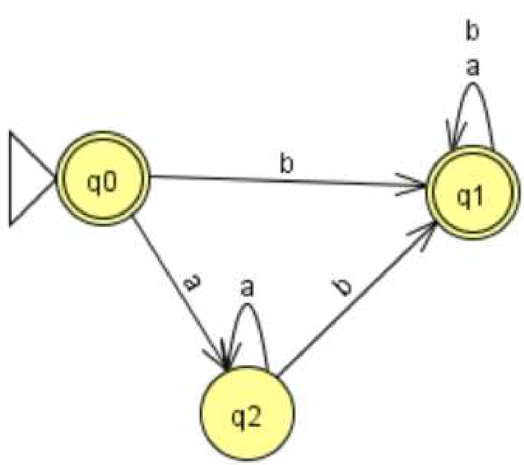
M = (Q, ∑, δ, q0, F') where

Q = {q0, q1, q2}

∑ = {a, b}

F' = {q2}

L'(M) = {λ,b,ab,bb,ba,…}



We can observe that the set of states, the initial state and transition function of M and M' are same. The final states for both L(M) and L'(M) are different. From this we can say that w ϵ L'(M) as L'(M) also inputs the same input alphabet as L(M).

L(M) = { w ϵ ∑* : δ*(q0,W) ϵ F} where W is the walk to the final state

Whereas,

L'(M) = { w ϵ ∑* : δ*(q0,W) does not belong to F}

It can be seen that L'(M) is a Finite Language and also a DFA, therefore L'(M) is Regular.