DFAs, NFAs and Languages

CS142 Assignment 1, Tanha Kate

Section 1

Problem A

The alphabets of the language in DFA M are 0 and 1, and M recognizes if the number of 0s (|0|) and the number of 1s (|1|) is |0| - |1| mod 3 = 1. In other words, L(M) = {w | |0| - |1| mod 3 = 1}.

Problem B

At first, we will construct the state transition table for all pairs of state of the DFA in Figure 1. All the accept states are highlighted yellow.

	0	1
q_0	q_5	q_1
q_1	q_0	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_6

Two states 'A' and 'B' are said to be equivalent if

$\delta(A,X) \rightarrow F$ and $\delta(B,X) \rightarrow F$ or $\delta(A,X) \nrightarrow F$ and $\delta(B,X) \nrightarrow F$ where 'X' is any input string and F is a final state in the DFA. If |X| = n, then A and B are said to be n equivalent.

Consider the case for **0 equivalence**, in which we simply write the non-final states and the final states as two different sets.

$\{q_0, q_1, q_3, q_4\}\{q_2, q_5\}$

Now consider **1 equivalence,** in which we take a pair of states from each set and check where they go on getting a particular input. If those states fall in the same set as the 0-equivalence set, they are 1-equivalent:

$\{q_0, q_3\}$
$\{q_1, q_4\}$
$\{q_2, q_5\}$

Now consider **2 equivalence:**

$\{q_0, q_3\}$
$\{q_1, q_4\}$
$\{q_2, q_5\}$

Therefore, we can merge the states in each set together and the new final state is any state which contains one of the final states of the initial DFA.

Therefore, the transition matrix for the minimized DFA is:

	0	1
q_0q_3	q_2q_5	q_1q_4
q_1q_4	q_0q_3	q_2q_5
q_2q_5	q_1q_4	q_0q_3

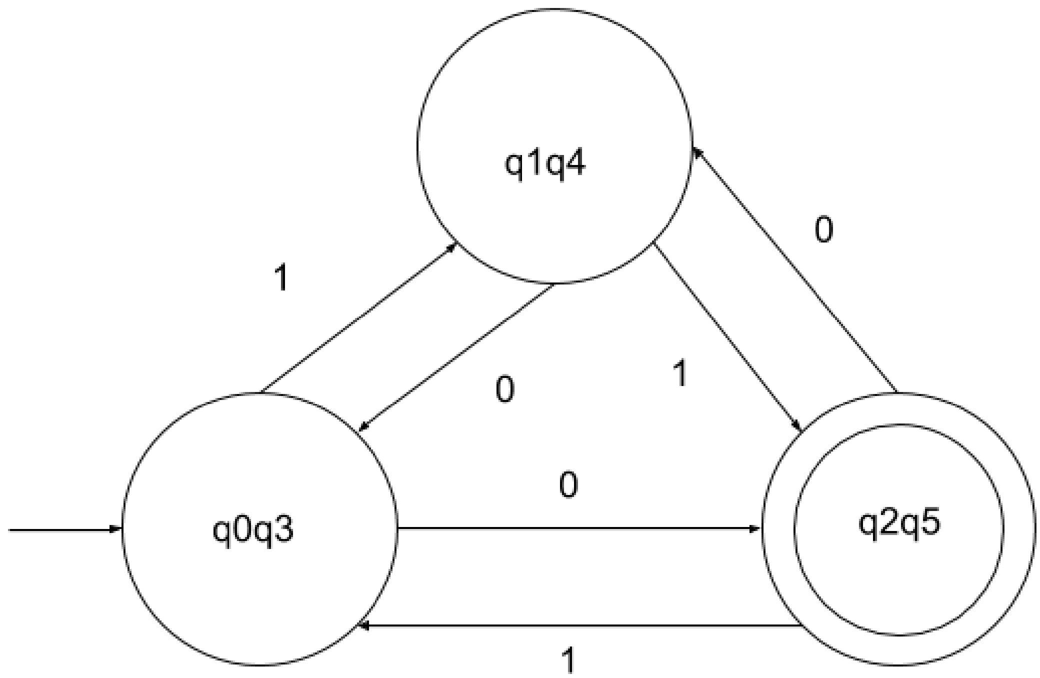

Figure 1. Minimized DFA for the Finite Automaton M.

Section 2

Question 1

Since A is regular, there is a DFA D = (Q, Σ, δ, q_0, F) such that D recognizes A. Our goal is to construct an NFA N = (Q', Σ', δ', q'_0, F') that will recognize A^R. In other words, we want to show that regular languages are closed under reversal, that is, for all w, $w \in L(D) \Leftrightarrow w^R \in L(N)$

.

Note that NFAs are easier to construct and according to Theorem 1.39 in Sipster, 2013, every NFA has an equivalent DFA.

The steps in constructing N are:

- Reverse all the arrows of D.
- Convert the initial state of D q_0 as the only accept state for N.
- Add a new initial state q'_0 that $\varepsilon - transitions$ to each state of N which were previously the accept states of D.

Formally, we can define N as follows:

1. $Q' = Q\ U\ \{q'_0\}$, we add a new initial state
2. $F' = \{q_0\}$, so that N only recognizes if our computation ends in the initial state of the original DFA.
3. Define δ' as stated below:

$$\delta'(q'_0, \varepsilon) = F \qquad if\ q = q'_0\ and\ a = \varepsilon \quad (3.1)$$
$$\delta^{-1}(z, a) = \{q' | \delta(q', a) = z\} \qquad all\ q' \in Q,\ a \in \Sigma \quad (3.2)$$
$$\delta'(q'_0, a) = \varnothing \qquad \text{for all } a \in \Sigma \quad (3.3)$$

The first line of δ' indicates $\varepsilon - transitions$ from the new initial state q'_0 to F.
The second line basically reverses all the transition arrows in D.
The last line basically defines all other possibilities from q'_0 as leading to the death state.

We hypothesize that L(N) = A^R. We prove this by showing first that w $\in L(D) \rightarrow w^R \in L(N)$.

Since $w \in L(D)$, we know that w = $w_1 w_2 ... w_n$ and there exists a series of states such that $r_0, r_1, ..., r_n$ where

a) $r_0 = q_0$
b) for all i between $0 < i \leq n,\ r_i = \delta(r_{i-1}, w_i)$.
c) $r_n \in F$, and

Therefore, N will accept w^R, which can be rewritten as $\varepsilon w_n w_{n-1} ... w_1$, with the state sequence $q'_0, r_n, r_{n-1}, ..., r_1$. As defined earlier in 3, q'_0 and $r_1 = F' = q_0$ are initial and final states for N respectively, so we only need to show that each reverse transition is valid for N.

By 3.1, the first transition satisfies $r_n \in \delta'(q'_0, \varepsilon)$ as it reduces to the established $r_n \in F$.

The remaining transitions have the form $r_{i-1} \in \delta'(r_i, w_i)$ which by 3.2 becomes $r_{i-1} \in \{q' | \delta(q', a) = r_i\}$. This follows from $\delta(r_{i-1}, w_i) = r_i$ which was established by w ∈ L(D).

Now, we will show that $w \in L(N) \rightarrow w^R \in L(D)$.

Since w $\in L(N)$, we know that w = $w_1 w_2 ... w_n$ ad there are states $r_0, r_1, ..., r_n$ such that $r_0 = q'_0,\ r_n \in \{q_0\}\ and\ r_{i+1} \in \delta'(r_i, w_{i+1})$.

Lines 3.1 and 3.3 define all transitions on q'_0, which indicates that $w_1 = \varepsilon\ and\ r_i \in F$. All other transitions are defined by line 3.2, so we know that $r_1, r_2, ..., r_n$ are in Q, the set of states of the DFA D.

r_n is $q_0\ and\ r_1 \in F$. Since $w \in L(N),\ we\ know\ that\ r_i \in \delta'(r_{i-1,} w_i)$. That is $r_i \in \{q | \delta(q, w_i) = r_{i-1}$. So, $\delta(r_i, w_i) = r_{i-1}\ as\ required.$ Therefore, D accepts $w_n w_{n-1} ... w_2$ with the state sequence $r_n, r_{n-1}, ..., r_1$.

Note: I consulted the following two sources for the proof of correctness.

(2019). [Ebook]. Retrieved from http://web.cecs.pdx.edu/~hook/cs581sp11/reverse.pdf
(2019). [Ebook]. Retrieved from http://www.public.asu.edu/~ccolbou/src/355hw2s11sol.pdf

Question 2

Problem a

Consider w containing as many single column matrices with three rows, the elements of which represent binary digits (0,1). AddBin3 computes the binary addition of binary numbers such that the bottom row of w is the sum of the two top rows. In other words, the automaton AddBin3 **recognizes** binary sums.

Problem b

According to Definition 1.16 (in Sipster, 2013)., “a language is called a regular language if some finite automaton recognizes it.”

Before we begin, we will expand the alphabet for Σ_3 and denote each alphabet using a character.

(0,0,0): A
(1,0,0): B
(0,1,0): C
(0,0,1): D
(1,1,0): E
(0,1,1): F
(1,0,1): G
(1,1,1): H

Σ_3 = {A,B,C,D,E,F,G,H}

Now, we will construct a DFA M = ($Q, \Sigma, \delta, q_0, F$) which accepts the language B.

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \Sigma_3$
δ *is described as*

	A	B	C	D	E	F	G	H
q_0	q_0	q_2	q_2	q_1	q_2	q_0	q_0	q_2
q_1	q_2	q_1	q_1	q_2	q_0	q_2	q_2	q_1
q_2	q_2	q_2	q_2	q_2	q_2	q_2	q_2	q_2

$q_0 = q_0$
F = q_0

Notice that the state q_0 is at once the initial state and the accept state of the automaton and state q_1 indicates situations where there is a carry from the next bit. State q_2 is essentially the death state when the input violates the binary addition property. Since B is recognized by the above DFA, it's a regular language.

Problem C

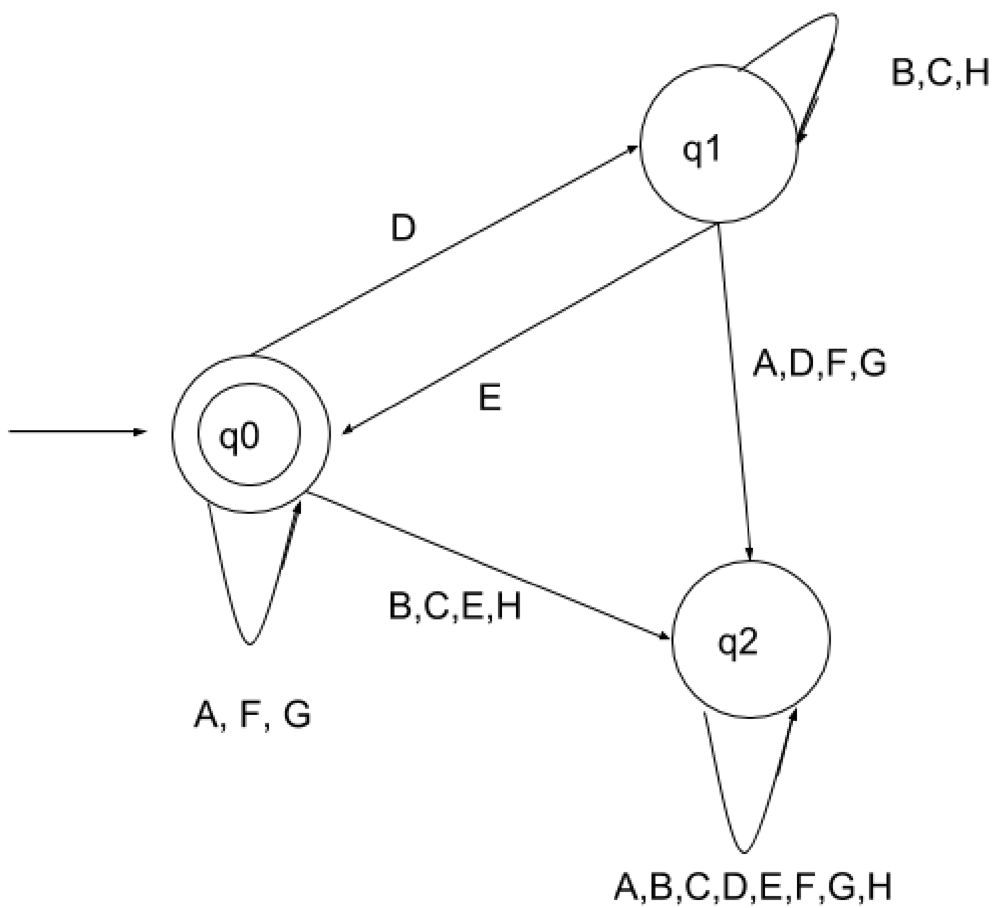

Figure 2. State diagram of DFA for AddBin3.

Notice that this is a DFA because there is exactly one exiting transition arrow for each symbol in the alphabet. There is no choice or randomness and there aren't any ε transitions.

References

Sipser, M. (2013). *Introduction to the theory of computation*. Boston, MA: Cengage Learning.