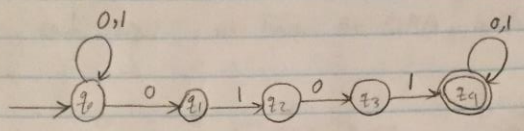
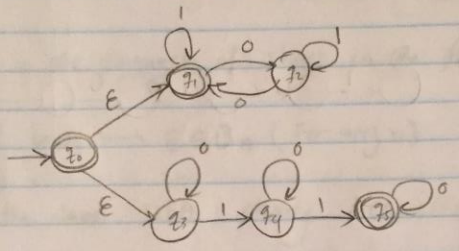
Adel Danandeh
HW#3

Exercise 7.7 (b,c,d,e,g,h)

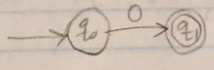
b)



c)

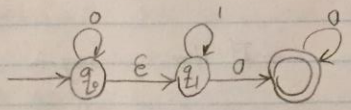


d) language {0} with two states

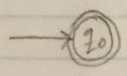


e) The language $0^*1^*0^+$ with three states

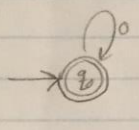
$0^*1^*0^*0$ with three states



g) The language $\{\epsilon\}$ with one state.



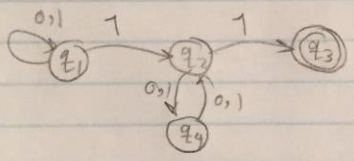
h) The language 0^* with one state.



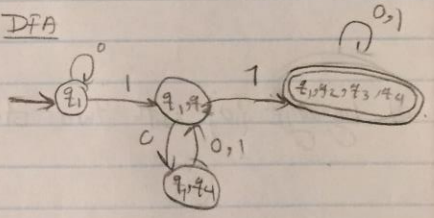
Exercise 1.13

$\Sigma = \{0,1\}$

NFA



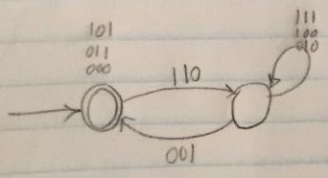
DFA



Exercise 1.32

$B = \{w \in \Sigma_3^* \mid$ the bottom row of w is the sum of the top two rows$\}$

We assume that the input is read from right to left. (Exercise 31)



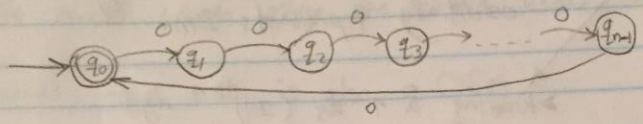
* If there is carry-in, addition is invalid.

* we need two states, representing carry-in and no carry-in.

If a language is regular the reverse is regular.

(1.36) $B_n = \{a^k \mid k \text{ is a multiple of } n\}$.

for each $n \geq 1$, we have the DFA with n states



B_n is the language of strings in the form of 0^{ni} $(i \in N)$

If $i=0 \longrightarrow \epsilon \in B_n$ (For any n)

In the above DFA clearly $L(A) = B_n$, and so B_n is regular, for any n. ▨

(1.37) $C_n = \{x \mid x \text{ is a binary number that is a multiple of } n\}$

show for $(n \geq 1)$ the language C_n is regular.

Let's say we have n states represented as the following: $0, \ldots, n-1$

Any state the machine is in represents the remainder of the number divided by n. Now, we need to figure out what happens when we read another digit.

$\underbrace{0, \ldots, 9}, \ldots, n-1$

Call it S and look at the binary representation 'b'

Read a digit as the $(j+1)^{th}$ digit

- Reading a 0 as $(j+1)^{th}$ digit

  Which this gives us the string S_0 which it essentially represents $2m$.

- Reading a 1 as $(j+1)^{th}$ digit

  This gives us the string S_1 as $2m+1$.

(1.47) $\{w | w = a_1 b_1 \cdots a_k b_k$, where $a_1 \cdots a_k \in A$ and $b_1 \cdots b_k$, each $a_i, b_i \in \Sigma^*\}$

Let's consider two DFAs as the following:

$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes A

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognizes B

Now, let's design a NFA N such that it recognizes the shuffle of A and B.

At any given time NFA N needs to keep track of all states of M_1 and M_2. N will move in M_1 or M_2 accordingly when a character is read. In order for N to accept a string both DFAs are in accept

states, If not the string gets rejected.

(we take from each M_1 and then M_2, which this is essentially the $A \times B$ which they are closed under cross multiplication. In other words, the $A \times B$ is also a DFA).