CPSC 313 — Winter 2020
Assignment 1 — Finite Automata and Regular Languages

Haohu Shen

30063099

1. **The Language of a 2-State DFA**

   Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and put $L = L(M)$. By considering the different possibilities for the set of final states F of M, prove that if Q consists of exactly two states, then one of the following hold (note that (b) and (c) are not mutually exclusive):

   (a) L is empty.
   (b) L contains the empty string.
   (c) L contains a string of length 1.

   Solution. Let Q consist of exactly 2 states q_0 and q_1 and q_0 is the start state according to the problem, we prove that one of properties (a), (b) and (c) are hold. We consider 4 cases and prove them respectively.

   Case 1: If $q_0 \notin F$ and $q_1 \notin F$, that is, M has no final states, we prove that (a) is hold by contradiction.
   Suppose (a) is not hold, that is, L is not empty. Suppose $w \in L$, since $L = L(M)$, we have $L \subseteq L(M)$, thus $w \in L(M)$, thus M accepts w, therefore, by the formal definition of the language of a DFA, w ends up in a final state, which contradicts that M has no final states. Thus, the assumption is false and L is empty, thus (a) is hold.

   Case 2: If $q_0 \in F$ and $q_1 \notin F$, that is, the only final state of M is q_0, we prove that (b) is hold by construction.
   Suppose $w = \epsilon$, when M reads w, it ends at q_0. Since q_0 is a final state, M accepts w by the formal definition of the language of a DFA, thus $w \in L(M)$, since $L(M) = L$, we have $L(M) \subseteq L$, thus $w \in L$, that is, L contains the empty string w, thus (b) is hold.

   Case 3: If $q_0 \notin F$ and $q_1 \in F$, that is, the only final state of M is q_1, we prove that (c) is hold by construction.
   Since q_1 is the only final state, $\exists w_0 \in L$ such that $\delta(q_0, w_1) = q_1$ by the formal definition of the language of a DFA. Suppose $w = w_1$, when M reads w, M moves from q_0 to q_1 while processing the symbol w_1 in w, since q_1 is a final state, M accepts w, thus $w \in L(M)$, since

$L = L(M)$, we have $L(M) \subseteq L$, thus $w \in L$, since w only contains one symbol w_1, $|w| = 1$, that is, L contains a string of length 1, thus (c) is hold.

Case 4: If $q_0 \in F$ and $q_1 \in F$, that is, all states of M are final states, we prove that (b) is hold by construction.
Suppose $w = \epsilon$ and M reads w, as we showed in case 2, (b) is also hold in this case.

Therefore, in all cases, at least one property of (a), (b) and (c) is true, thus we can conclude that the statement is true.

2. **DFA Design and Analysis**

(a) Design a DFA M with at most 4 states for the language

$$L = \{w \in \{0,1\}^* \mid \; w \text{ contains at least one 0 and } |w| \text{ is even}\}.$$

Present your DFA in the form of a state diagram and include a brief rationale for its design.

Solution. Since the DFA M must "remember" four things, there should be 4 states:

i. no 0 is encountered yet and encountered even number of 1s
ii. no 0 is encountered yet and encountered odd number of 1s
iii. a 0 is encountered and the total number of 0s and 1s encountered is odd
iv. a 0 is encountered and the total number of 0s and 1s encountered is even

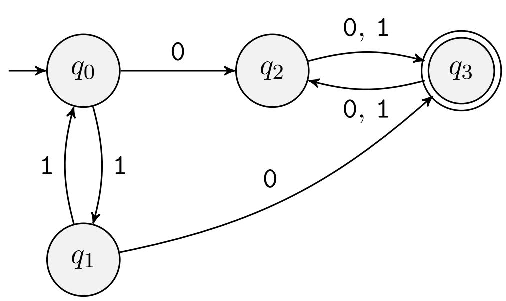
Thus our DFA M must keep track of the parity of the number of leading 1s(whether the number is even and odd or not) until a 0 is encountered. If a 0 is encountered, it must keep track of the total number of 0s and 1s encountered so far and accept if the total number of 0s and 1s encountered so far is even.
Therefore, we use 4 states to keep track of the following situations:

- state q_0 indicates no 0 is encountered yet and the number of 1s encountered so far is even;
- state q_1 indicates no 0 is encountered yet and the number of 1s encountered so far is odd;
- state q_2 indicates a 0 is encountered and the total number of 1s and 0s encountered so far is odd;
- state q_3 indicates a 0 is encountered and the total number of 1s and 0s encountered so far is even;

Now we need to identify M's start state and all accept states:

- Since at the beginning of every string, no 0 is encountered and the number of 1s encountered is zero and zero is even, q_0 is the start state.
- Since M can only accept a string if at least a 0 is encountered and the total number of 1s and 0s encountered is even, q_3 is the only one accept state of M.

Thus we have the following DFA M defined over the alphabet $\{0, 1\}$:



(b) Prove that your DFA M of part (a) accepts the language L, i.e. prove that $L(M) = L$.

Solution. In order to prove $L(M) = L$, we prove $L(M) \subseteq L$ and $L \subseteq L(M)$ respectively.

Firstly, we prove $L(M) \subseteq L$, to that end, let $w \in L(M)$, and we prove that $w \in L$. Since q_3 is the only accept state, M is in state q_3 when w is accepted and the state is only reachable from state q_1 or state q_2, we consider two cases:

Case 1: If M reaches q_3 from q_1, w must contain a 0. Moreover, in this case q_3 is only reachable from q_1 via a series of back-and-forth visits between q_0 and q_1, according to the transitions from q_0 to q_1 and q_1 to q_0, w must contain odd number of leading 1s. Thus w contains odd number of leading 1s and a 0. Thus w contains at least one 0 and $|w|$ is even. Hence $w \in L$.

Case 2: If M reaches q_3 from q_2, w must contain a 0 or a 1. Moreover, M can only reach q_2 from q_0 or q_3, thus we can split case 2 into two subcases.

Subcase 1: If M reaches q_2 from q_0, then w must contain a 0 and reach q_2 by a series of back-and-forth visits between q_0 and q_1, according to the transitions from q_0 to q_1 and q_1 to q_0, w must contain even number of leading 1s, thus w contains even number of leading 1s and a 0. According to the transition between q_2 and q_3, w contains an extra 0 or 1, which makes its length even and contains at least one 0. Hence $w \in L$.

Subcase 2: If M reaches q_2 from q_3, then w must contain a 1 or 0. Let u be the string w up to q_3, according to the transitions from q_2 to q_3 and q_3 to q_2, we have $w = u01$ or $w = u00$ or $w = u10$ or $w = u11$. Since u contains at least one 0 and $|u|$ is even, w contains at least one 0 and $|w|$ is even among all four situations. Hence $w \in L$.

Since in both cases we prove that $w \in L$, we can conclude that if $w \in L(M)$, $w \in L$, therefore $L(M) \subseteq L$.

Secondly, we prove $L \subseteq L(M)$, to that end, let $w \in L$ and we show that $w \in L(M)$. Thus w contains at least one 0 and $|w|$ is even. We consider two cases and trace through M's states as well as transitions as it processes w:

Case 1: If w contains odd number of leading 1s, since transitions from q_0 to q_2 or q_1 to q_3 accept 0 only, also transitions from q_0 to q_1 or q_1 to q_0 accept 1 only, M moves from the start state q_0 and visits between states q_0 and q_1 back and forth when processing the leading 1s of w, ending in q_1. As M processes the 0 after all leading 1s, M moves from q_1 to q_3. Let u be the string w up to q_3 and v be the remain of w followed by u, we have $w = uv$ and $|u|$ is even. Since $|w|$ is even, $|v|$ is even. Thus if $v = \epsilon$, then M ends at q_3, otherwise M moves to q_3 again via a series of back-and-forth visits between q_3 and q_2 when processing even length of symbols in v, ending at q_3 as well. Since q_3 is an accept state, $w \in L(M)$.

Case 2: If w contains even number of leading 1s, since transitions from q_0 to q_2 or q_1 to q_3 accept 0 only, also transitions from q_0 to q_1 or q_1 to q_0 accept 1 only, M moves from the start state q_0 and visits between states q_0 and q_1 back and forth when processing the leading 1s of w, ending in q_0. As M processes the 0 after all leading 1s, M moves from q_0 to q_2. Let u be the string w up to q_3 and v be the remain of w followed by u, we have $w = uv$. Since $|w|$ is even and $|u|$ is odd, $|v|$ is odd, thus M moves to q_3 again via a series of back-and-forth visits between q_2 and q_3 when processing odd length of symbols in v, ending at q_3. Since q_3 is an accept state, $w \in L(M)$.

Since in both cases we prove that $w \in L(M)$, we can conclude that if $w \in L$, $w \in L(M)$, therefore $L \subseteq L(M)$.

Thus $L(M) \subseteq L$ and $L \subseteq L(M)$ and we can conclude that $L(M) = L$.

3. **NFA Design and Analysis**

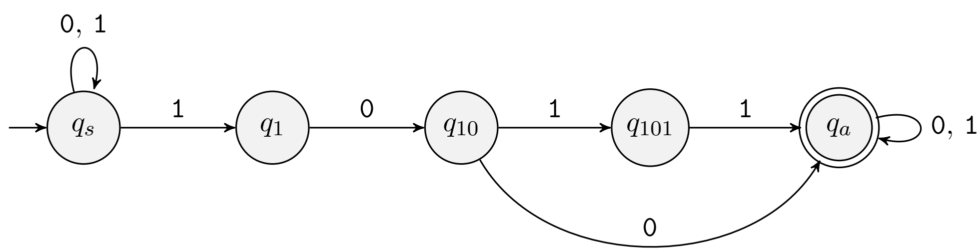
(a) Design an NFA N with at most 5 states for the language

$$L = \{w \in \{0,1\}^* \mid \; w \text{ contains the substring 1011 or the substring 100}\,\}.$$

Present your NFA in the form of a state diagram and include a brief rationale for its design.

Solution. Suppose N starts in the start state q_s and N has not encountered the substring 1011 or the substring 100 yet. Since the substring 1011 and the substring 100 have the common prefix 10, N encounters a 1 and decides non-deterministically that it is the first 1 of the common prefix at some point. If this is the case, N will move from q_s to the state q_1 which remembers this first 1, then moves to the state q_{10} that remembers the second character of the common prefix, which is 0.
After that, if N encounters a 0, it will move to q_a to remember that, thus q_a tracks that M encounters the full substring 100. Otherwise, N encounters a 1 and will move to q_{101} that remembers it. In order to make q_a to be an accept state, it must both track that M encounters full substring 100 and M encounters the full substring 1011, thus we can make M move from q_{101} to q_a to remember that an extra 1 is added. And now q_a becomes an accept state, no matter how many and what type of symbols will be processed later.

Thus we have the following NFA N defined over the alphabet $\{0, 1\}$:



(b) Prove that your NFA N of part (a) accepts the language L, i.e. prove that $L(N) = L$.

Solution. In order to prove $L(N) = L$, we prove $L \subseteq L(N)$ and $L(N) \subseteq L$ respectively. Firstly, we prove that $L \subseteq L(N)$, to that end, let $w \in L$, we show that $w \in L(N)$. Since $w \in L$, we have $w \in \{0,1\}^*$ such that w contains 1011 or 100, thus w has the form $w = x10yz$ such that $x \in \{0,1\}^*$, $z \in \{0,1\}^*$ and $y = 11$ or $y = 0$. And the following state sequence through N ends in the accept state q_a after processing w:

- Transition from q_s to q_s throughout processing x, which includes the case that $x = \epsilon$;
- Transition from q_s to q_1 on 1 following x;
- Transition from q_1 to q_{10} on 0 following 1;
- If $y = 0$, a transition will happen from q_{10} to q_a on the only symbol of y following 0, otherwise $y = 11$ and a transition will happen from q_{10} to q_{101} on the first symbol of y, which is 1, and a consequent transition will happen from q_{101} to q_a on the second symbol of y, which is also 1.
- Transition from q_a to q_a throughout processing z, which includes the case that $z = \epsilon$;

Since N is in the accept state q_a after processing w, N accepts w, thus $w \in L(N)$, therefore $L \subseteq L(N)$.

Secondly, we prove that $L(N) \subseteq L$, to that end, we prove its contrapositive statement, that is, let $w \notin L$, we prove that $w \notin L(N)$. Thus w contains neither the substring 1011 nor the substring 100.

We prove it by contradiction. Suppose N accepts w, thus after processing w, N is in the accept state q_a. Since $w \in \{0,1\}^*$, we can represent w as $w = w_1w_2...w_{n-1}w_n$ such that $1 \leq i \leq n$ and $w_i \in \{0,1\}^*$.

Let $1 \leq t \leq n$ such that w_t is the first symbol causing a transition from a state different from q_a to q_a, since all transitions out of q_a direct to itself, N is not able to move from q_a to states other than q_a once it arrives at q_a, thus all symbols $w_tw_{t+1}...w_n$ will be throughout processed from q_a to q_a. Since there are two transitions from states other than q_a to q_a, we consider two cases:

Case 1: Since q_{10} is one of states other than q_a such that there exist a transition from q_{10} to q_a, we suppose N moves from q_{10} to q_a when processing w_{t-1}, thus $w_{t-1} = 0$. Moreover,the only way to reach q_{10} is from q_1 via a 0 transition and the only way to reach q_1 is from q_s via a 1 transition, thus $w_{t-2} = 0$, $w_{t-3} = 1$, thus $w_{t-3}w_{t-2}w_{t-1} = 100$, thus w contains the substring 100, which contradicts that $w \notin L$.

Case 2: Since q_{101} is one of states other than q_a such that there exist a transition from q_{101} to q_a, we suppose N moves from q_{101} to q_a when processing w_{t-1}, thus $w_{t-1} = 1$. Moreover,the only way to reach q_{101} is from q_{10} via a 1 transition and the only way to reach q_{10} is from q_1 via a 0 transition and the only way to reach q_1 is from q_s via a 1 transition, thus $w_{t-2} = 1$, $w_{t-3} = 0$ and $w_{t-4} = 1$, thus $w_{t-4}w_{t-3}w_{t-2}w_{t-1} = 1011$, thus w contains the substring 1011, which contradicts that $w \notin L$.

So the assumption that N accepts w is false, w is rejected by N, that is, $w \notin L$.
Thus $L \subseteq L(N)$ and $L(N) \subseteq L$. Therefore, we can conclude that $L(N) = L$.