CS101 Homework Assignment 3

Tadius Frank

September 24, 2021

1. **Academic Honesty Declaration**

   In the process of finishing this homework:

   (a) I had conversations about the contents and solutions of this assignment with the following people: Melat Feseha

   (b) I consulted the following resources, such as books, articles, webpages:
   - https://wch.github.io/latexsheet/
   - https://en.wikibooks.org/wiki/LaTeX/Basics

   (c) I did not look at the answers of any other students.

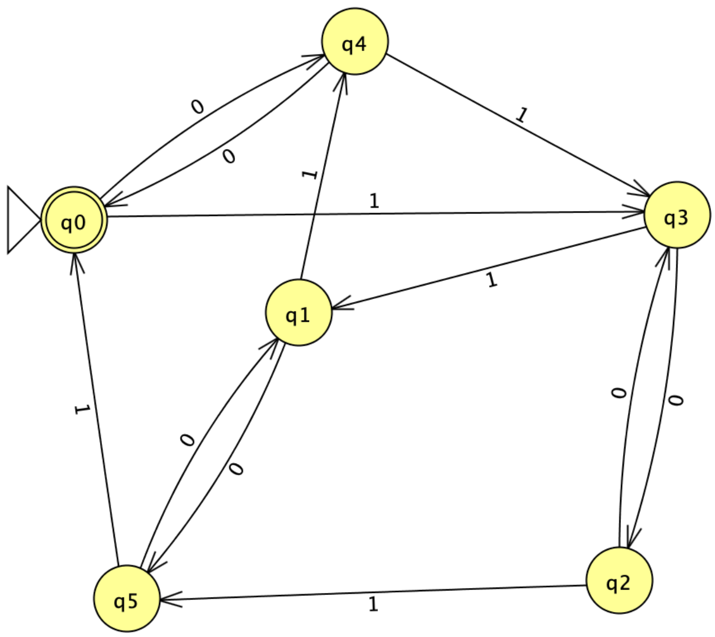
   (d) I did not provide my answers to other students.

2. **Writing Component**

(a) **Deterministic FSM**

Build a deterministic finite state machine that accepts the set of all strings from $\{0,1\}$ such that the number of 0's in x is even and the number of 1's in x is a multiple of three. Provide some intuitive justification that your DFSM accepts exactly that language. A proof by induction of correctness is NOT necessary.

Solution:



Explanation: We know that the Machine M, represented in the graph above, accepts a finite input alphabet comprised of strings from the set $\{0, 1\}$, but the strings must be comprised of an even number of 0's and the number of 1's need to be odd. To construct this machine, we must consider the diffrent possible distribution of 0's (whether they are odd or even) and of 1's for the multiples of 3 (whether we are on the zeroth, second, or third multiple).

(q0):accepts the string if it is an empty string (ϵ), and allows the string to be processed if it has even 0's and on the 0th multiple.

(q1): allows the string to be processed if it has odd 0's and on the 2nd multiple.

(q2): allows the string to be processed if it has even 0's and on the 1st multiple.

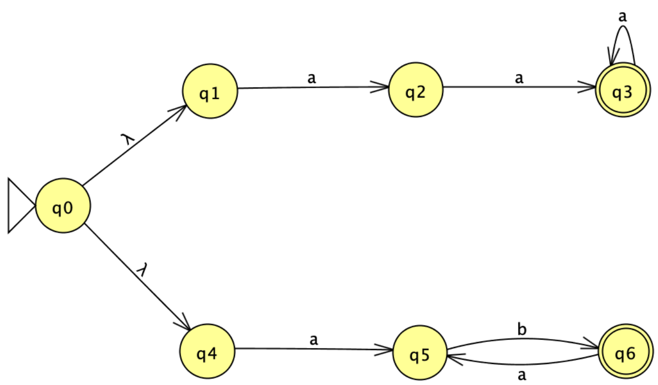
(q3): allows the string to be processed if it has odd 0's and on the 1st multiple.

(q4): allows the string to be processed if it has odd 0's and on the 0th multiple.

(q5):allows the string to be processed if it has even 0's and on the 2nd multiple.

(b) **Non-Deterministic FSM**
Draw a state diagram for an NDFSM that accepts the language given by regular expression $(ab)^* \cup (aa^*)$. Give an intuitive explanation of why your machine accepts that language.

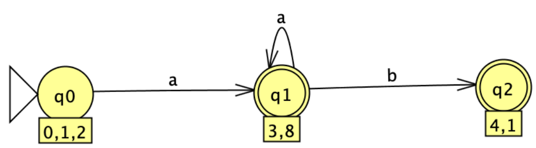
Solution:



Explanation: The machine represented by the diagram is the $(ab)^* \cup (aa^*)$ where two seperate machines were combined utilizing epsilon (here lambda) moves from the starting state. One machine represents $(ab)^*$ and another (aa^*). As a result, if a string input is part of the set of strings from the finite alphabet $(ab)^*$ or (aa^*) the NDFSM will find the correct pathway to process that string input.

3. **Non-Deterministic to Deterministic**
   FSM Convert the NDFSM in Figure 3 to a DFSM. As a first step, please show the epsilon closure of each state in the original machine. For each of the states in the new deterministic machine, list the states that it corresponds to in the original machine.

   Solution:



4. **Closure**

Let min be a function on languages such that: $min(L) = \{w \epsilon L \mid \text{no proper prefix of } w \text{ is in } L\}$.Thus if w in $min(L)$ then w in L, but there is no prefix of w that is in L.

If $L = L(a^*b+)$, what is $min(L)$? Explain your answer.

Solution: $min(L) = (a^*b)$

Explanation: After listing out the possible input strings that are present in the set of strings comprised of alphabet (a^*b+), there's a clear pattern that arises in that if a string is comprised of more than one b creates a prefix that is present in L which is not accepted by min. However, the number of a can be zero or more since it doesn't create prefixes. As a result, the distribution of a's which is zero or more can be represented by (a^*) and the distribution of b's which is exactly one b would be (b) which demonstrates that input strings accepted by $min(L)$ must comprised with the regular expression (a^*b).

5. **Closure**
   Let min be a function on languages such that: $min(L) = \{w \; \epsilon \; L \mid \text{no proper prefix of } w \text{ is in } L\}$. Thus, if w in $min(L)$ then w in L, but there is no prefix of w that is in L.

   Prove that If L is regular, $min(L)$ is also regular.

   Solution: Given language L is regular, there's a DFSM M that accpets L we must show that the language $min(L)$, where there is no prefix of a string w that is in L, is regular. As a result, we must show that there's a DFSM M' that accepts $min(L)$.

   - $(L(M') \subset min(L))$ Let the language $min(L)$, where there is no prefix of a string w that is in L be given. A DFSM M' that accepts $min(L)$ would not accept a string w with prefixes in L because the string w would reach the accepting state of M' and have leftover characters (which would make it go through the accept state more than once) which indicates there's a prefix accepted by Language L. Because the strings accepted by M' have no prefixes in L, then all strings accepted by M' are in $min(L)$.
   - $(L(M') \subset min(L))$ Let a string w in min(L) be given. We know that the string w has no prefixes that are in L; as a result, w is accepted by M' since it would arrive at the accepting state with no leftover charcaters (it would only go through the accept state once).

   Therefore, since all string inputs accepted by M' must be in $min(L)$ and that strings inputs in $min(L)$ are accepted by M', these must be equivalent. This equivalence justifies that $min(L)$ is a regular language because there is some DFSM which is M' that accepts this language.

for closure problem, but it wasn't needed.