# CSC 311—Theory of Computation

## DFAs, NFAs, regular expressions, a regular grammar

Tutorial 1 Second Semester 2017

Assume that $\Sigma = \{a, b\}$ when it is not otherwise given.

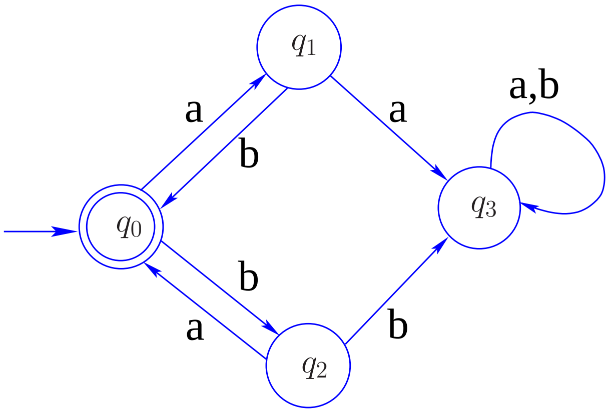
1. Give a DFA for which $L(M) = \Sigma^* aba \Sigma^*$.

2. Give a DFA for which $L(M) = \{w | w \text{ starts and ends with the same symbol}\}$.

3. Do Exercise 1.4 from Sipser (p. 83). See Sipser for solutions of **b** and **d**.

4. Do Exercise 1.5 from Sipser (p. 84). See Sipser's solutions for **a** and **b**.

5. Consider the following DFA $M$.



   (a) Why is it a DFA? Is it also an NFA? Why?

   (b) Describe the language recognized by the DFA.

   (c) Write a regular grammar that recognizes $L(M)$.

   (d) Write a state table for this DFA.

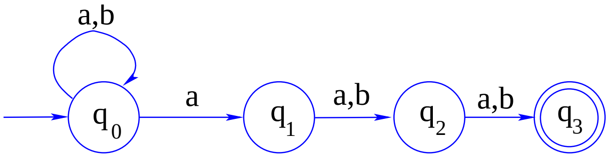
   (e) Write a regular expression for the language $L(M)$.

   (f) Write some python code to represent the DFA for the language $L(M)$ and then give a `while` loop in python to determine if a string represented `s` is accepted by the machine or is rejected. Do this exercise without resorting to the python's `re` package.

   (g) Use the `re` package in python to recognize expressions that the DFA $M$ recognizes. Test your code with a few representative regular expressions.

6. Do Exercises 1.8, 1.9, and 1.10 from Sipser (p. 85).

7. Consider the following NFA, $N$.



(a) Give a formal description $(Q, \Sigma, \delta, q_0, F)$ of $N$

(b) Construct an equivalent DFA, $M$.

(c) Minimize the DFA, $M$.

8. Do Exercise 1.16 from Sipser (p. 86).

9. Do Exercise 1.17 from Sipser (p. 86).

10. Do Exercise 1.21 from Sipser (p. 86).

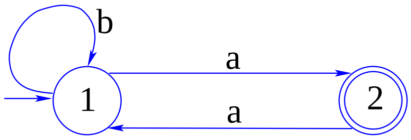
11. Do Exercise 1.24 from Sipser (p. 87).

12. Do Problem 1.31 (Sipser p. 88).

13. Do Problem 1.40a (Sipser p. 89).

14. Do Problem 1.50 (Sipser p. 90).

15.



(a) Is it a DFA? Is it an NFA? Why?

(b) If it is not a DFA turn it into the DFA M and describe the language recognized by the DFA.

(c) Write a regular grammar that recognizes $L(M)$.

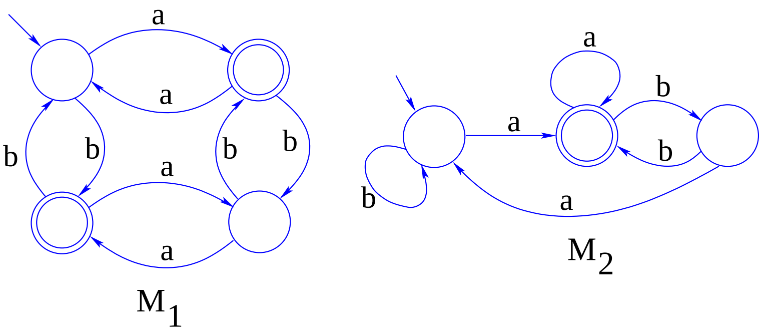
(d) Write a state table for this DFA.

(e) Write a regular expression for the language $L(M)$.

(f) Minimize the DFA for M.

(g) Write some C-code to represent the DFA for the language $L(M)$ and then give a `while` loop in C to determine if a string represented as `char* s;` is accepted by the machine or is rejected.

16. $M_1$ and $M_2$ recognize languages $L_1$ and $L_2$, respectively.



(a) Draw an NFA that will recognize $L_1^* L_2$.

(b) Draw an NFA that will recognize $L_1 \cup L_2^*$.

(c) Do not draw the final NFA, but describe how you will go about constructing the NFA that will recognize $L_1 \cap L_2$.

17. Draw a DFA equivalent to the regular expression

$0 + 10^* + 01^*0$

18. Find regular expressions corresponding to each of the following subsets of $\{0, 1\}^*$.

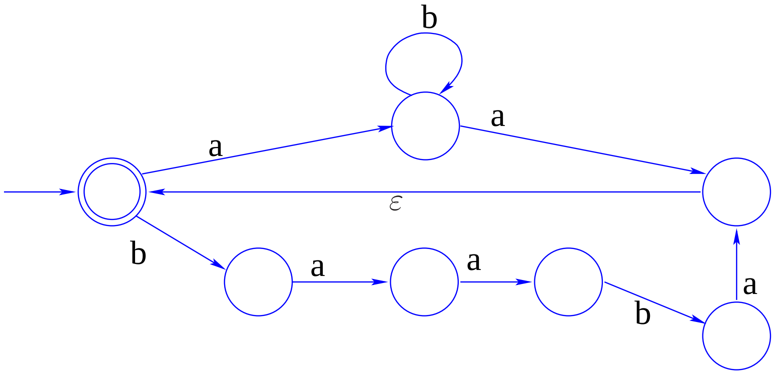
(a) The language of all strings that do not end with 10.

(b) The language of all strings in which the number of 0's is even.

(c) The language of all strings in $0^*$ in which the number of 0's is divisible by 6.

(d) The language of all strings in $\{0, 1\}^*$ in which the number of 0's is divisible by 6.

(a) Find a regular expression corresponding to the following NFA:



19. (a) 1.6(b) in Sipser

(b) 1.7(b) in Sipser

(c) 1.8(b) in Sipser

20. 1.14(b) in Sipser

21. 1.16(b) in Sipser.