SOLUTIONS: Homework Assignment 2

CSCI 2670 Introduction to Theory of Computing, Fall 2018

September 18, 2018

This homework assignment is about NFA, NFA to DFA conversion, operations on regular languages, and regular expressions

1. Design an NFA to recognize the following language, where $\Sigma = \{a, b, c\}$

$$L_1 = \{w : \text{ the second last symbol of } w \text{ is not `}a\text{' } \}$$

Answer:

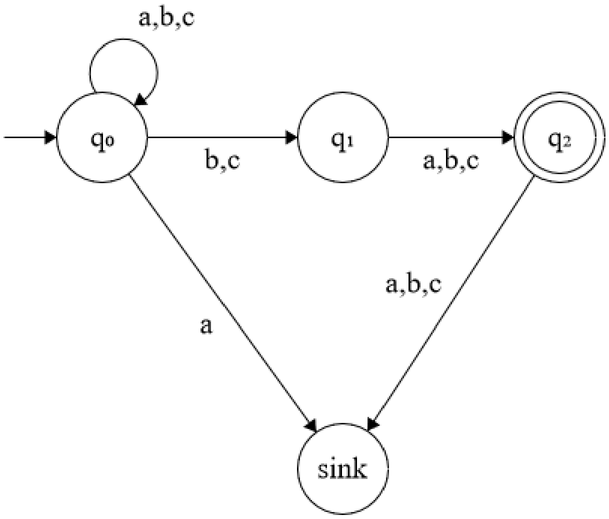

Figure 1: NFA for L_1 (credited: Connor Dooley)

2. Design an NFA to recognize the following language, where $\Sigma = \{a, b, c\}$

$$L_2 = \{w : w \text{ contains an even number of } a\text{'s or contains the pattern 'aa'}\}$$

Answer:

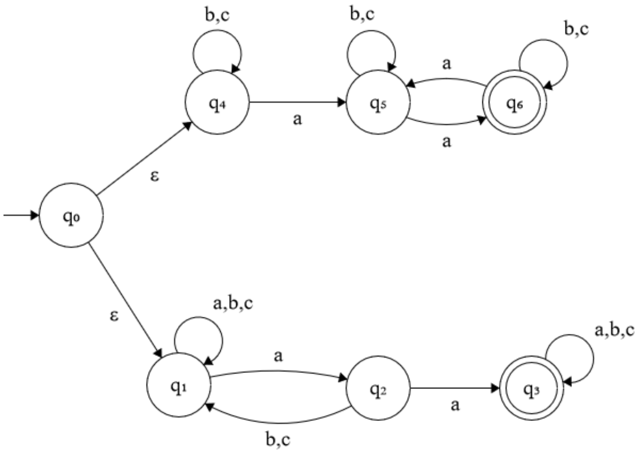

Figure 2: NFA for L_2 (credited: Connor Dooley)

3. Based on the work for Questions 1 and 2, design an NFA to recognize each of the following languages:

   (a) $L_1 \cup L_2$;

   (b) $L_1 L_2$;

   (c) L_1^*;

   Answer:

   (a)

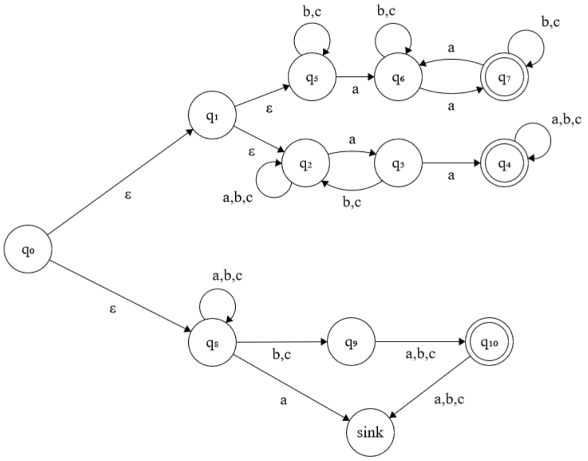

Figure 3: NFA for $L_1 \cup L_2$ (credited: Connor Dooley)

   (b)

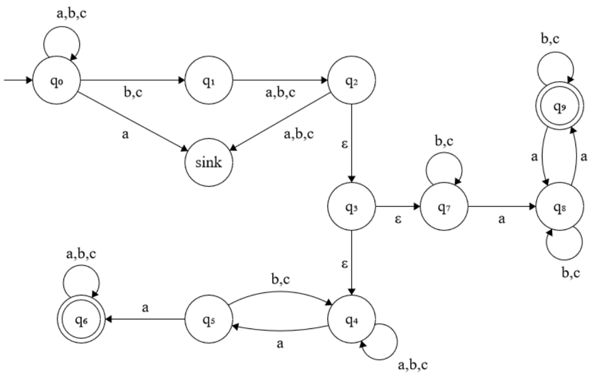

Figure 4: NFA for $L_1 L_2$ (credited: Connor Dooley)

(c)

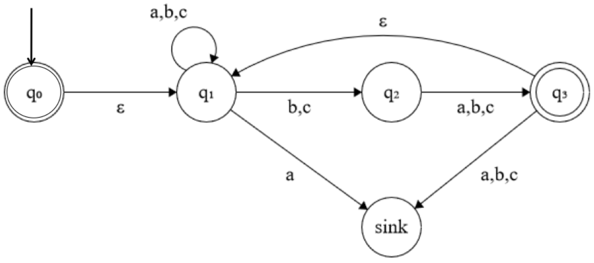

Figure 5: NFA for L_1^* (credited: Connor Dooley)

4. Consider the following NFA. Convert it to an equivalent DFA using the studied method. Note the simple conversion process is to construct the extension set E after related transitions are determined. For example, if R is a subset of states in the NFA that has transition function δ, then for symbol x, the new transition function $\Delta(R, x)$ is computed as the result of the following sequence of steps:

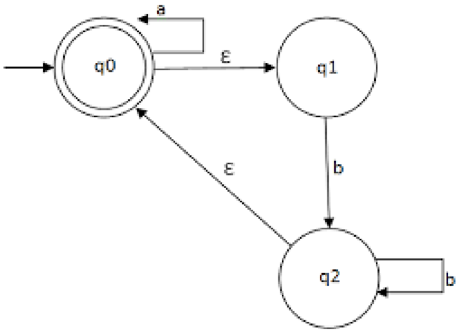

Figure 6: state diagram of an NFA

(1) compute $\delta(r, x)$ for every $r \in R$;

(2) compute extension set $E(\delta(r, x))$ for every $r \in R$, if relevant ϵ-transitions exist;

(3) take the union $\bigcup_{r \in R} E(\delta(r, x))$;

Draw the final DFA converted from the NFA.

Answer:

You would like to show all steps in computing the new transition function $\Delta(R, x)$ for every R and every x. Showing these steps serve two purposes:

(a) to reward you with partial credits even if your final answer may be wrong, and

(b) to remind you how you have come to the final answer.

(1) There are 8 possible states in the new DFA:

$$\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}$$

(2) The new transition Δ is defined as:

$\Delta(\{q_0\}, a) = \{q_0, q_1\}, \Delta(\{q_0\}, b) = \emptyset$

$\Delta(\{q_0, q_1\}, a) = \{q_0, q_1\}, \Delta(\{q_0, q_1\}, b) = \{q_0, q_1, q_2\}$

$\Delta(\{q_0, q_1, q_2\}, a) = \{q_0, q_1\}, \Delta(\{q_0, q_1, q_2\}, b) = \{q_0, q_1, q_2\}$

(3) The start state is $\{q_0, q_1\}$;

(4) The accept states are $\{q_0, q_1\}$ and $\{q_0, q_1, q_2\}$.

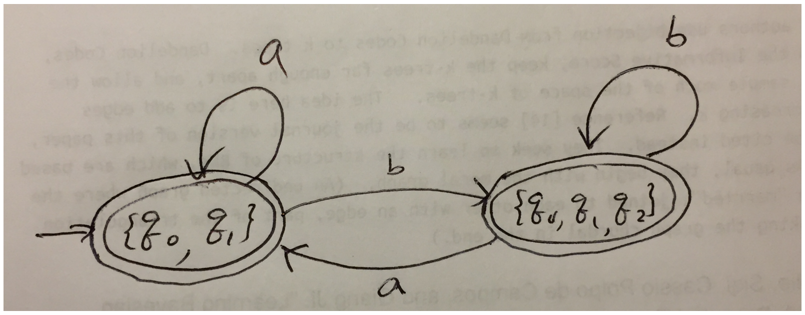

Figure 7: Converted DFA

5. For each of the following regular expressions, give two positive and two negative members for the language it generates:

   (a) $a(ba)^*b$;

   (b) $(\epsilon \cup b)a$;

   Answers:

   (a)

   positive members: ab, $abababab$

   negative members: aba, ϵ

   (b)

   positive members: a, ba

   negative members: ab, bb

6. Design an NFA for each of language given in Question 5.

   Answers

   (a)

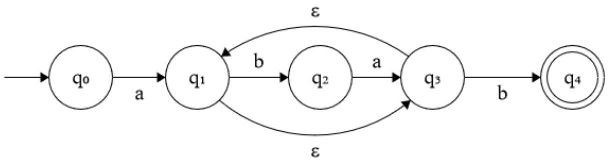

Figure 8: NFA for $a(ba)^*b$ (credited: Connor Dooley)

   (b)

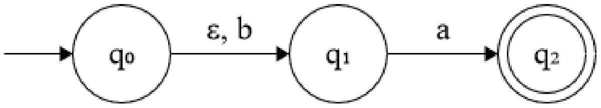

Figure 9: NFA for $(\epsilon \cup b)a$ (credited: Connor Dooley)

7. Give regular expressions for the following languages, where $\Sigma = \{0, 1\}$

   (a) $\{w : w$ contains exactly two 0’s $\}$

   (b) $\{w : w$ contains at least two 0’s and at most one 1 $\}$

   Hints: There are only a few ways that “exactly two 0’s” can be arranged in a string. “At least two 0’s” is the same as ”exactly two 0’s or more than two 0’s”.

   answers:

   (a) $1^*01^*01^*$

   (b) $000^* \cup 1000^* \cup 0100^* \cup 000^*10^*$

8. In certain programming languages, comments appear between delimiters such as /# and #/. Let C be the language of all valid delimited comment strings. Such a string in C must begin with /# and end with #/ but have no intervening #/. For simplicity, assume the alphabet $\Sigma = \{a, b, /, \#\}$.

   (a) Give an NFA that recognizes language C.

   (b) Give a regular expression that generates C.

   Answers:

   (a)

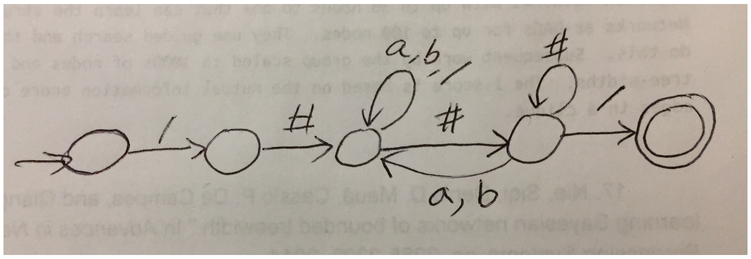

Figure 10: NFA that recognizes language C

   (b) $/\#(a \cup b \cup / \cup (\#^*(a \cup b)))^*\#/$

NOTE: All homework answers need to be word-processed or typed. Hand-writing only applies to figure or table drawings. A hard copy of answers should be received in classroom or in the instructor's office by 5:00pm on the due date. Policy on late homework answers is given in the syllabus. *Email submission will not be accepted unless a such a request has been approved.*