CS46 Homework 3

This homework is due at 9AM on Monday, February 6. Part 1 should be submitted on Automata Tutor; part 2 should be written using LaTeX and submitted using github. This is an individual homework. It’s ok to discuss approaches at a high level, but your write-up should be your own: do not share it, and do not read other people’s write-ups. Please refer to the course webpage or ask me any questions you have about this policy.

0. Before final submission, make sure to fill out the README file. ← Seriously!

Part 1 — These problems should be completed and submitted on Automata Tutor. Each problem will let you attempt several times, and give you useful feedback. You should try the practice problems first. I recommend that you first try to solve the problems on paper, then use the site to debug your solutions.

1. (Sipser 1.7d) Construct an NFA with two states for the language \( \{0\} \) over the alphabet \( \{0, 1\} \).

2. (Sipser 1.7c) Construct an NFA with six states for the language \( \{w \mid w \) contains an even number of 0s, or contains exactly two 1s\} over the alphabet \( \{0, 1\} \).

3. Let \( \Sigma = \{a, b\} \) and consider languages:
   
   \[ L_1 = \{w \mid \text{every } a \text{ in } w \text{ is immediately followed by a } b\} \]
   
   \[ L_2 = \{w \mid w \text{ contains an odd number of } bs\} \]

   Construct:
   
   (a) an NFA recognizing \( L_1 \cup L_2 \).
   
   (b) an NFA recognizing \( L_1L_2 \).

4. Let \( \Sigma = \{a, b\} \) and let \( L_3 = \{w \mid w \) contains an even number of as and an odd number of bs\}. 

   Construct an NFA recognizing \( L_3^* \).
   
   (Hint: You can use the construction of Theorem 1.49 to get a correct answer, but this NFA will be capable of being simplified. Try describing \( L_3^* \) in English first.)

5. (Sipser 1.16)

   (a) Construct a DFA equivalent to the following NFA:

   ![NFA Diagram]

   1If you want to use late days on this assignment, you will need to submit solutions to these problems via github or on paper. The site has only one deadline, which is 9AM Monday.
(b) Construct a DFA equivalent to the following NFA:

6. Write a regular expression for the language
\[ \{ w \mid w \text{ contains exactly two } a\text{s or at least two } b\text{s} \} \]
over the alphabet \( \Sigma = \{a, b\} \).

7. Let \( \Sigma = \{0, 1\} \) and \( L = \{ w \mid w \text{ contains the substring } 0ab0 \text{ or } 1ab1 \text{ where } a, b \in \Sigma \} \).
   
   (a) Construct a DFA that recognizes \( L \).
      (It is strongly recommended that you plan on paper first!)
      
      For fun: Take a screenshot of your masterpiece, email it to Lila, treasure it forever.
   
   (b) Construct an NFA that recognizes \( L \).
      Your NFA should have substantially fewer states than your DFA. (Phew!)

8. Extra credit. (Sipser 1.28a) Construct an NFA for each of the languages given by regular expressions:
   
   (i) \( a(abb)^* \cup b \)
   
   (ii) \( a^* \cup (ab)^* \)
   
   (iii) \( (a \cup b^+) a^+ b^+ \)
Part 2 — These problems should be typeset in \LaTeX{} and submitted using \texttt{github}.

9. For each of the following regular expressions over $\Sigma = \{a, b\}$, explain in English what language they describe. (Show your thought process.)

(a) $b^*a(b^*a)^*$
(b) $(a^*\emptyset b \cup ab \cup b^*\emptyset a)(b \cup \emptyset)$
(c) $\epsilon \cup a(ba)^* \cup b(ab)^*$

10. Give regular expressions for the languages accepted by NFAs $M_1$ and $M_2$. (Hint: use Lemma 1.60.)

![Figure 1: NFA $M_1$.](image1)

![Figure 2: NFA $M_2$.](image2)

11. In C++, it is possible to write a comment as a string beginning with ‘/\*’ and ending with ‘*/’. In between, it can contain any characters, including ‘/’ and ‘\*’, as long as it does not contain the substring ‘*/’ before the end of the string. For example, ‘/\*ab/ab/c/b\*\*\*ac*/’ is a valid comment, but ‘/\*c\*ab/baaaa\*\*\*a/a//bb*/’ is not.

Construct a regular expression for the language $L$ of this style of well-formed C++ comments over the alphabet $\Sigma = \{a, b, c, *, /\}$.

$L = \{w \mid w = /\*x*/ , \text{ where } x \in \Sigma^* \text{ does not contain the substring ‘*/’} \}$

You should explain your reasoning/show your work. A formal proof is not required, but you should be able to justify (at a high level) why your regular expression is correct.

12. Prove that regular languages are closed under complement.
(\textbf{Hint:} Use direct proof. For any regular language $L$, give a general construction of a DFA recognizing $\bar{L}$.)
13. **Extra credit.** Write a regular expression for the language

\[ \{w \mid w \in \{a, b\}^* \text{ contains an odd number of } bs \text{ and an even number of } as\} \]

Either start from a DFA for this language and show your work for converting the DFA to a regular expression, or justify how you came up with the regular expression and why it is correct. A formal proof is not required, but you should explain at a high level what is happening in your regular expression.

14. **Extra credit.** We have a set of rules that describe how to build regular expressions, given an alphabet \(\Sigma\) and some additional symbols, ‘\(\emptyset\)’, ‘\(\cup\)’, ‘\(\circ\)’, ‘\(*\)’, ‘\(\langle\)’, and ‘\(\rangle\)’. Consider the language \(L\) over the alphabet given by \(\Sigma\) with these added symbols, defined as:

\[ L = \{w \mid w \text{ is a regular expression over } \Sigma\} \]

Prove that \(L\) is not regular.

(Thinking dizziness warning: be careful not to think yourself in circles with this one! A cool corollary of this claim is that there is no regular expression that matches all regular expressions.)