Syntax and semantics - Regular Languages

1 Learning Objectives

1. Describe an automaton from a state diagram
2. Design a DFA that recognizes a language with certain specifications
3. Construct a DFA that recognizes the complement, the union or the intersection of the languages recognized by given DFAs.
4. Construct a NFA that recognizes a given language
5. Given a NFA, construct an equivalent DFA
6. Describe regular expressions
7. Construct a NFA that recognizes a given regular expression
8. Describe, using regular expressions, the language recognized by a DFA
9. Use Pumping Lemma to prove that a given language is not regular

2 Readings

Sipser's book:
Part I – Automata and Languages, section 1. Regular Languages

3 Homework - Exercises

Exercise 1. Let $L_1 = \{aa, bb, bbb\}$ and $L_2 = \{abba, aab, bb\}$. Specify the following languages:

1. $L_1 \circ L_2 = \boxed{}$
2. $L_1 \cup L_2 = \boxed{}$
3. $L_1 \cap L_2 = \boxed{}$
4. $L_1 \setminus L2 = \boxed{}$
5. Provide at least three strings belonging to L_2^*:
   $\boxed{}$

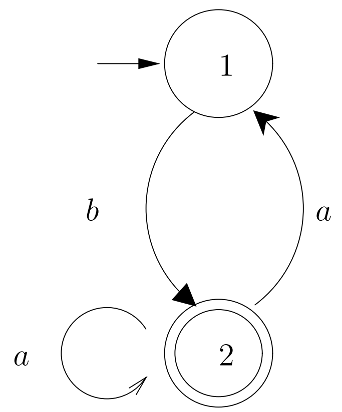
Exercise 2. Construct finite automata that recognize the following languages. Draw them and specify their components.

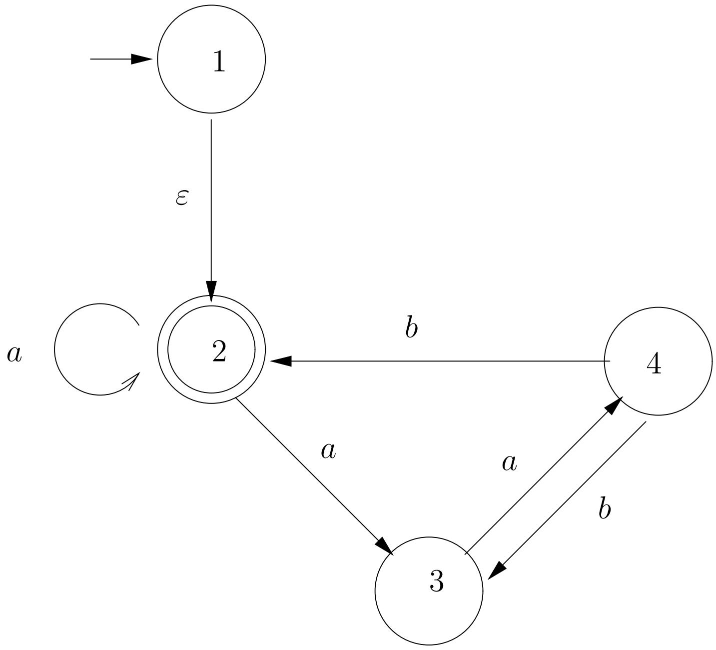
1. $L_1 = \{w \in \{a, b\}^* \mid w \text{ contains the substring } aa\}$
2. $L_2 = \{w \in \{a, b\}^* \mid w \text{ contains at least 2 occurrences of } b\}$.
3. $L = L_1 \cup L_2$.

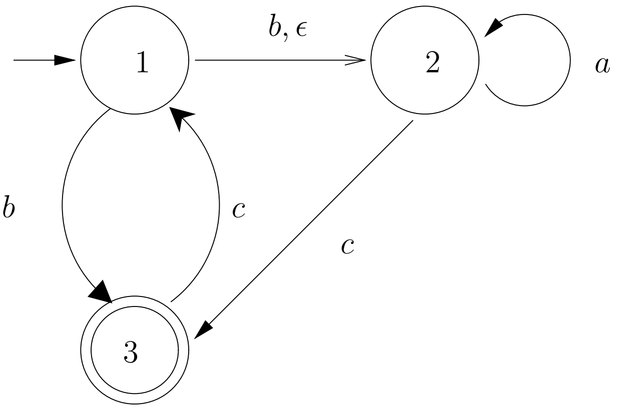
Exercise 3. Construct the nondeterministic automata that recognize the languages

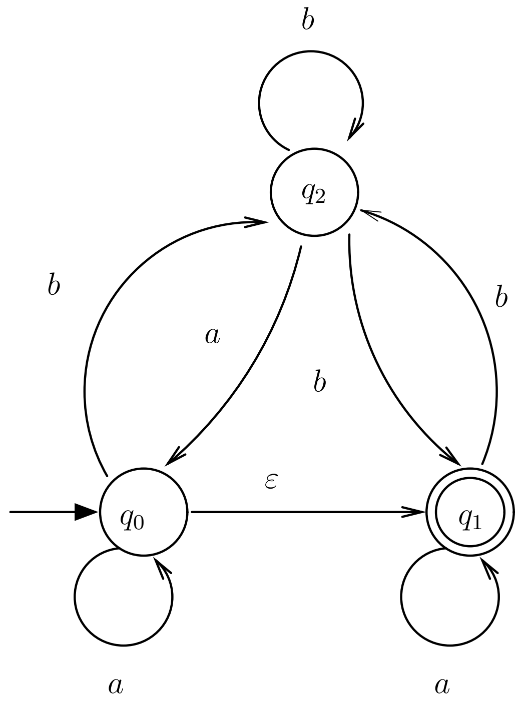
$L_1 = \{w \in \{a, b\}^* \mid w \text{ contains the substring } abab\}$,
$L_2 = \{w \in \{0, 1\}^* \mid w \text{ contains an even number of occurrences of 0 or exactly two of 1}\}$.

Exercise 4. Covert the nondeterministic automaton below to a deterministic finite one using the method explained today.




1
ε
a
2
b
4
a
a
b
3


1
b, ϵ
2
a
b
c
c
3



Exercise 5. Construct an NFA that recognizes the language

$$L_3 = \{w \in \{a,b\}^* \mid w \text{ contains the substring } bab\}\,.$$

It must be a proper NFA, not a DFA.

Exercise 6. Here are three regular expressions over the alphabet $\{\mathtt{a},\mathtt{b}\}$. Specify for each of them two different strings that belong to the language denoted by the regular expression, and two strings that *do not* belong to the language.

$$(\mathtt{b} \cup \mathtt{aa})(\mathtt{aab} \cup \varepsilon)$$

In: ☐ Not in: ☐

$$((\mathtt{a} \cup \mathtt{bb})^*\mathtt{a}) \cup (\emptyset^* \cup (\mathtt{bba}^* \cup \mathtt{aa})^*\mathtt{baba})$$

In: ☐ Not in: ☐

$$(\mathtt{baaba} \cup \emptyset)(\mathtt{bb} \cup (\mathtt{ba} \cup \mathtt{ab})^*)$$

In: ☐ Not in: ☐

Exercise 7. Give to each of languages below a regular expression that denotes it. *Use the textbook's notation for regular expressions.*

- The language of the strings that start with `0` and end with `22` or `010`.

- The language of the strings that either has length `2` or contains an odd number of `1`-occurrences.

- The language of the strings that do *not* contain occurrences of the substring `100`.

Exercise 8. Consider the following regular expressions. Using the method described today, convert them to equivalent NFAs. *Use the textbook's method, i.e., no ad hoc solutions and "smart" shortcuts.*

$$(\mathsf{ab} \cup \emptyset)(\varepsilon \cup (\mathsf{ba} \cup \mathsf{ab})^*)\ .$$

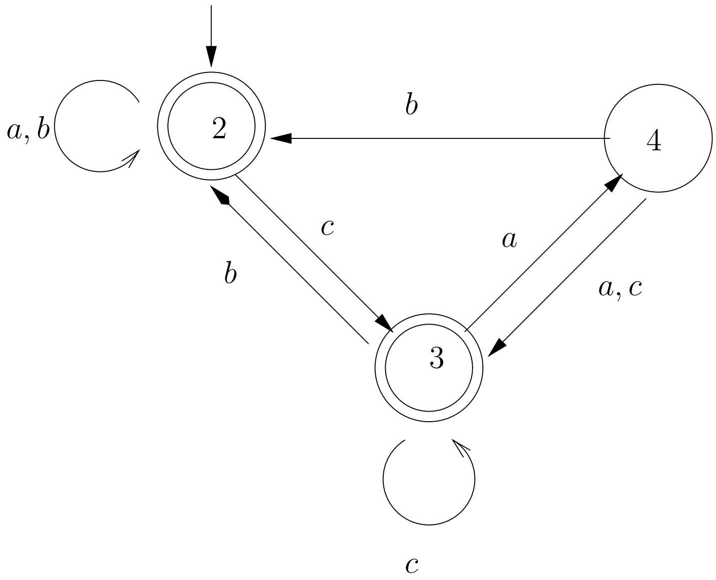
$$(\mathsf{bc} \cup \mathsf{aaa})^*\ .$$

$$(\mathsf{ac})^*(\mathsf{bb} \cup \mathsf{a})$$

$$((\mathsf{ab})^* \cup (\mathsf{ba} \cup \mathsf{a}))^*$$

$$(\mathsf{ab} \cup \mathsf{c})^*\mathsf{ac}$$

Exercise 9. Consider the following DFA:



Using the algorithm explained today, define a regular expression equivalent to the DFA. *You must only use the textbook's method, i.e., no ad hoc solutions and "smart" shortcuts.*

Exercise 10. Consider the following non-regular languages:

$$L_1 = \{a^i b^j c^k \mid k = j + i\}$$

$$L_2 = \{w \in \{a, b\}^* \mid w \text{ have fewer instances of } a \text{ than } b\}.$$

$$L_3 = \{a^i b^j c^k \mid k = j + i\}.$$

1. For each of them, give two examples of words which belong to them and two examples of words that do not belong to them.
2. Prove that they are not regular languages.