Reminder about guidelines and some advices/remarks

- Duration: 2 hours (08:00am → 10:00am).
- No exit before 30 minutes. No entry after 30 minutes.
- Any document of the course or the tutorial is allowed. Other documents are not authorized.
- Any electronic device is forbidden (calculator, phone, tablet, etc.).
- **Care of your submission will be taken into account (-1 point if there is a lack of care).**
- Exercises are independent.
- The grading scale is indicative.

Exercise 1 (True or False - 3 points)

Answer by True or False to the following questions. Justify *carefully* and concisely your answers (without proof). If a proposition is false, provide a counter-example.

1. Any finite language is a finite-state language.
2. Any finite-state language is a finite language.
3. A complete and deterministic automaton recognizes the universal language.
4. If an automaton has a loop, the language it recognizes is infinite
5. The difference between two regular languages is a regular language.
6. Let L be a language, then $L \cap L$ is a finite-state language.

Exercise 2 (Determinization, minimization, regular expression - 3 points)

Let $\Sigma = \{a, b, c\}$. Let us consider the automaton in Figure 1.

1. Determinize the automaton
2. Minimize the obtained automaton.
3. Compute a regular expression for this automaton using the method associating equations to states.

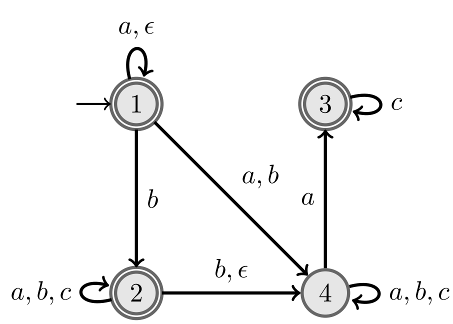


Figure 1: Automaton for Exercise 2

Exercise 3 (Intersection of Languages - 3 points)

We consider an alphabet Σ. The intersection of two non-regular languages L_1 and L_2 can be a regular language or not. Justify without proof your answers to the following questions.

1. Give an example of pair of non-regular languages L_1 and L_2 such that $L_1 \cap L_2$ is not regular.
2. Give an example of pair of non-regular languages L_1 and L_2 such that $L_1 \cap L_2$ is regular.

Exercise 4 (Algorithms - 4 points)

1. Give an algorithm that checks whether an input automaton is complete.
2. Give an algorithm that takes as input an automaton defined over some alphabet Σ and an alphabet $\Sigma' \subseteq \Sigma$ and determines whether the input automaton recognizes the universal language over Σ'.

Exercise 5 (Some relations between regular expressions - 5 points)

Let $\Sigma = \{a, b\}$. For a regular expression e, we note $L(e)$ the language associated to e. We consider two relations, $\rightarrow$ and $\nrightarrow$, between two regular expressions defined by:

$$e_1 \rightarrow e_2 \text{ if and only if } L(e_1) \subseteq L(e_2)$$

$$e_1 \nrightarrow e_2 \text{ if and only if } L(e_1) \not\subseteq L(e_2)$$

1. Establish all relations $(\rightarrow, \nrightarrow)$ between the following regular expressions, by justifying relations $\nrightarrow$ with counter-examples:
   1) $(a^*b^*)^*$
   2) $(aa^* + bb^*)^*$
   3) $(\epsilon + a + b)^*(a + b)$
   4) $\epsilon + a(a + b)^* + b(a + b)^*$

Exercise 6 (Determinization, minimization, regular expression - 3 points)

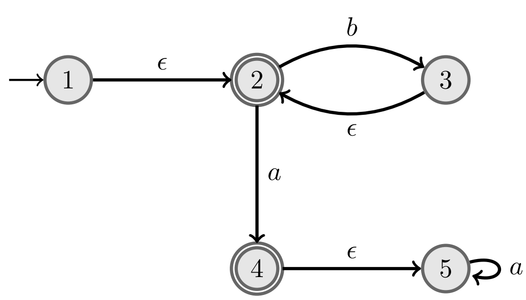
Let $\Sigma = \{a, b, c\}$. Let us consider the automaton in Figure 2a.

1. Suppress ϵ-transitions and determinize the automaton (either directly or in two steps).
2. Minimize the obtained automaton.

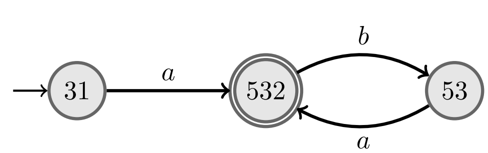
Exercise 7 (Determinization, minimization, regular expression - 2 points)

Let $\Sigma = \{a, b\}$. Let us consider the automaton in Figure 2b.

1. Compute a regular expression by using the method associating equations to paths.



(a) Automaton for exercise 6



(b) Automaton for exercise 7

Figure 2: Automaton for exercises 6 and 7