Licence Sciences et Technologies
Université Joseph Fourier

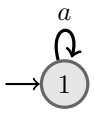
Mid-term Exam
INF 232: Languages and Automata
L2, 2013/2014

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.

Answer of exercise 1

1. True. A finite language contains a finite number of words. Each of these words can be recognized by an automaton. To obtain an automaton for the language, one just has to compute an automaton for the union of the languages recognized by the previous automata.
2. False. For instance the language of words that contain only a's is a finite-state language but not a finite language.
3. False. Consider the alphabet $\{a\}$. The following automaton is complete and deterministic but does not recognize the universal language (it actually recognizes the empty language).



4. False. Consider the automaton provided as a counter-example of the previous question.
5. True. The difference between two sets A and B is noted $A \setminus B$ and is defined as $A \cap \overline{B}$. Moreover, the regular languages are closed under intersection and complementation.
6. False. $L \cap L = L$. Thus, if L is not a finite-state language, $L \cap L$ will not be either.

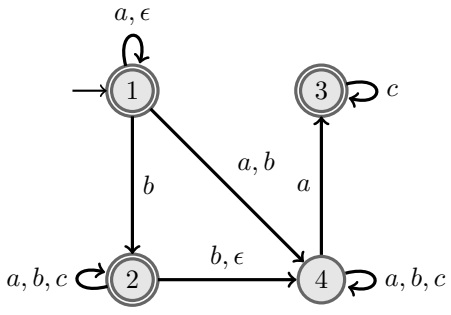


Figure 1: Automaton for Exercise 2

Answer of exercise 2

1. The automaton, represented by its transition table, is below:

	1*	14*	24*	134*	234*	34*	4
a	14	134	234	14	234	34	34
b	24	24	234	24	234	4	4
c		4	24	34	234	34	4

Remark. Alternatively, one could have noticed that the transition $(1, \epsilon, 1)$ is not needed. Moreover, since any word is accepted after state 2, we can suppress the transition $(2, b, 4)$ and $(2, \epsilon, 4)$. Similarly, the transition $(1, b, 4)$ can be suppressed. With these justifications, one could start with the simplified automaton.

2. Let us rename the states.

	1*	2*	3*	4*	5*	6*	7
a	2	4	5	2	5	6	6
b	3	3	5	3	5	7	7
c		7	3	6	5	6	7

Let us apply the minimization algorithm.

$\equiv_0$	$\equiv_1$	$\equiv_2$	$\equiv_3$
1	1	1	1
2	2	2	2
3	3	4	4
4	4	3	3
5	5	5	5
6	6	6	6
7	7	7	7

There is one equivalence class with two states, so the automaton is not minimal. We can merge states 3 and 5. We obtain:

	1*	2*	3*	4*	6*	7
a	2	4	3	2	6	6
b	3	3	3	3	7	7
c		7	3	6	6	7

3. To find the regular expression, one has to solve the equation system obtained from the nondeterministic automaton. Below, we use Arden's lemma on an equation $X = AX + B$ and check that $\epsilon \notin A$.

$$\begin{cases} X_1 &= aX_1 + bX_2 + (a+b)X_4 + \epsilon \\ X_2 &= (a+b+c)X_2 + (b+\epsilon)X_4 + \epsilon \\ X_3 &= (a+b+c)X_3 + \epsilon \\ X_4 &= aX_3 + (a+b+c)X_4 \end{cases}$$

$$\begin{cases} X_1 &= aX_1 + bX_2 + (a+b)X_4 + \epsilon \\ X_2 &= (a+b+c)X_2 + (b+\epsilon)X_4 + \epsilon \\ X_3 &= (a+b+c)^* \\ X_4 &= (a+b+c)X_4 + a(a+b+c)^* \end{cases}$$

$$\begin{cases} X_1 &= aX_1 + bX_2 + (a+b)X_4 + \epsilon \\ X_2 &= (a+b+c)X_2 + (b+\epsilon)X_4 + \epsilon \\ X_3 &= (a+b+c)^* \\ X_4 &= (a+b+c)^* a(a+b+c)^* \end{cases}$$

$$\begin{cases} X_1 &= aX_1 + bX_2 + (a+b)X_4 + \epsilon \\ X_2 &= (a+b+c)X_2 + (b+\epsilon)(a+b+c)^* a(a+b+c \\ &= (a+b+c)X_2 + (a+b+c)^* a(a+b+c)^* + \epsilon \\ X_3 &= (a+b+c)^* \\ X_4 &= (a+b+c)^* a(a+b+c)^* \end{cases}$$

$$\begin{cases} X_1 &= aX_1 + bX_2 + (a+b)X_4 + \epsilon \\ X_2 &= (a+b+c)^*((a+b+c)^* a(a+b+c)^* + \epsilon) \\ &= (a+b+c)^* \\ X_3 &= (a+b+c)^* \\ X_4 &= (a+b+c)^* a(a+b+c)^* \end{cases}$$

Consequently

$$\begin{aligned} X_1 &= aX_1 + b(a+b+c)^* + (a+b)(a+b+c)^* a(a+b+c)^* + \epsilon \\ &= aX_1 + (b + (a+b)(a+b+c)^* a)(a+b+c)^* + \epsilon. \end{aligned}$$

Using again Arden's lemma, as $\epsilon \notin a$, we get:

$$X_1 = a^*\big((b + (a+b)(a+b+c)^*a)(a+b+c)^* + \epsilon\big).$$

Answer of exercise 3

1. We give two examples:
   - Consider $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ and $L_2 = \{a^i b^j \mid i, j \in \mathbb{N} \wedge i \geq j\}$. We have $L_1 \cap L_2 = L_1$.
   - More generally, take L a non-regular language, we have $L \cap L = L$.
2. We give two examples:
   - Consider $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ and $L_2 = \{a^i b^j \mid i, j \in \mathbb{N} \wedge i > j\}$. We have $L_1 \cap L_2 = \emptyset$.
   - More generally, take L a non-regular language, $\overline{L}$ is a non-regular language (closure by complementation of regular and non-regular languages). We have $L \cap \overline{L} = \emptyset$ is a regular language.

Answer of exercise 4

1. The algorithm is implemented by function check_completeness(). The algorithms assumes an automaton $(Q, q_{init}, \Sigma, \delta, F)$. (The solution is not purposed to be efficient but simple).

```
1 complete := true ;
2 for each q in Q do
3 |   for each σ in Σ do
4 |   |   complete := complete ∧ |{(q, σ, q') ∈ δ | q' ∈ Q}| ≥ 1;
5 return complete
```

Algorithm 1: function check_completeness()

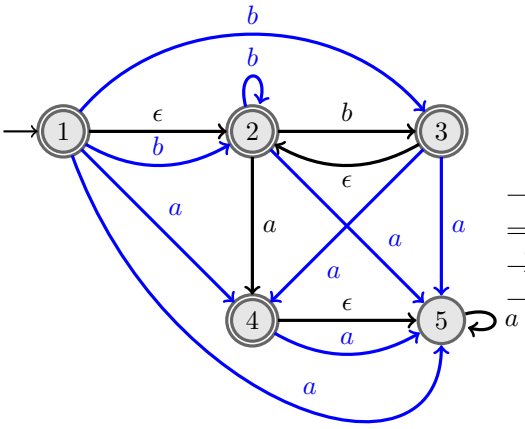
2. Several solutions are possible. Let $\mathcal{A} = (Q, q_{init}, \Sigma, \delta, F)$ be the input automaton. One solution is to consider δ the transition relation of $\mathcal{A}$ and consider $\delta' = \{(q, \sigma', q') \in \delta \mid \sigma' \in \Sigma'\}$. Let $\mathcal{A}' = (Q, q_{init}, \Sigma', \delta', F)$ be the restriction of $\mathcal{A}$ when considering Σ'. Moreover, one can build easily an automaton for Σ'^*. To check whether $\Sigma'^* \subseteq \mathcal{L}(\mathcal{A}')$, one can check whether $\Sigma'^* \cap \overline{\mathcal{L}(\mathcal{A}')} = \emptyset$.

Answer of exercise 5

1. We have immediately $1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4$. Let us remark that $(a^*b^*)^* = (aa^* + bb^*)^* = \epsilon + a(a+b)^* + b(a+b)^* = \{a, b\}^*$, thus $1 \to 4, 2 \to 4, 3 \to 4, 1 \to 2, 3 \to 2$, and $4 \to 2$. Moreover:
   - $1 \not\to 3$ because $\epsilon \in a^*b^*)^*$ but $\epsilon \notin (\epsilon + a + b)^*(a+b)$,
   - $3 \to 1$,
   - $1 \not\to 3, 2 \not\to 3$ and $4 \not\to 3$ because of ϵ.

Answer of exercise 6

1. The automaton where ϵ-transitions are removed is below on the left (one should ignore ϵ-transitions). Then the determinized version is below on the right.



	1*	45*	23*	5
a	45	5	45	5
b	23		23	

2. Let us rename the states of the automaton.

	1*	2*	3*	4
a	2	4	2	4
b	3		3	

$\equiv_0$	$\equiv_1$	$\equiv_2$
1	1	1
2	3	3
3	2	2
4	4	4

	1*	2*	4
a	2	4	4
b	1		

The automaton is not minimal because there is one equivalence class with two states. States 1 and 3 can be merged.

Answer of exercise 7

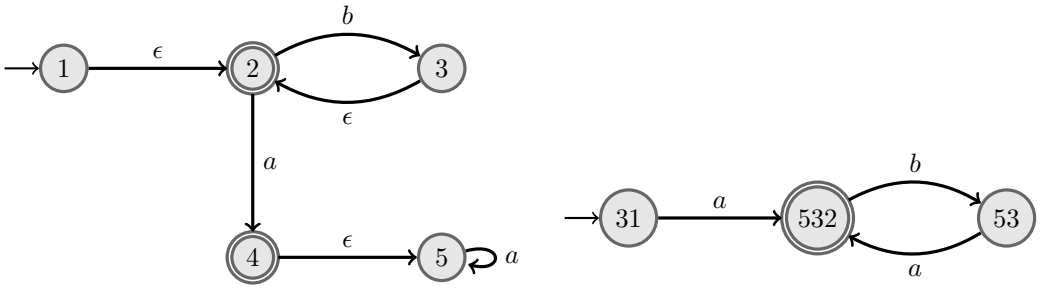
- $R_{11}^0 = \epsilon$
- $R_{12}^0 = a$
- $R_{13}^0 = \emptyset$
- $R_{21}^0 = \emptyset$
- $R_{22}^0 = \epsilon$
- $R_{23}^0 = b$
- $R_{31}^0 = \emptyset$
- $R_{32}^0 = a$
- $R_{33}^0 = \epsilon$

- $R_{11}^1 = \epsilon$
- $R_{12}^1 = a$
- $R_{13}^1 = \emptyset$
- $R_{21}^1 = \emptyset$
- $R_{22}^1 = \epsilon$
- $R_{23}^1 = b$
- $R_{31}^1 = \emptyset$
- $R_{32}^1 = a$
- $R_{33}^1 = \epsilon$

- $R_{11}^2 = \epsilon$
- $R_{12}^2 = a$
- $R_{13}^2 = ab$
- $R_{21}^2 = \emptyset$
- $R_{22}^2 = \epsilon$
- $R_{23}^2 = b$
- $R_{31}^2 = \emptyset$
- $R_{32}^2 = a$
- $R_{33}^2 = ab$

Finally the regular expression of the automaton is: $R_{12}^3 = R_{12}^2 + R_{13}^2(R_{33}^2)^* R_{32}^2 = a + a \cdot b \cdot (ab)^* \cdot a$.

Remark. States 1 and 2 are equivalent, and thus the automaton can be simplified beforehand.



(a) Automaton for exercise 6

(b) Automaton for exercise 7

Figure 2: Automaton for exercises 6 and 7