Cursus: M1, computer-science
Code UE: JEIN8602
Solutions to the subject: Formal languages theory
Date: 26 August 2014
Duration: 3H
Documents: authorized
Lectures by: Mr Géraud Sénizergues

Exercice 1 [/4]

$$e := ((abb)^*a(cb)^*) \cup (ab)^*(ba)^*$$

Let us apply Glushkov's method.
The locally testable language associated to e is:

$$L' := ((a_1b_1b_2)^*a_2(c_1b_3)^*) \cup (a_3b_4)^*(b_5a_4)^*$$

We compute

$$Ini(L') = \{a_1, a_2, a_3, b_5\}, \quad Fin(L') = \{a_2, b_3, b_4, a_4\},$$

$$Dig(L') = \{a_1b_1, b_1b_2, b_2a_1, b_2a_2, a_2c_1, c_1b_3, b_3c_1, a_3b_4, b_4a_3, b_4b_5, b_5a_4, a_4b_5\}$$

This gives the finite automaton represented on figure 1, where the set of states is

$$\{0, a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, b_5, c_1\}$$

0 is the unique initial state and a_2, b_3, b_4, a_4 are the final states.

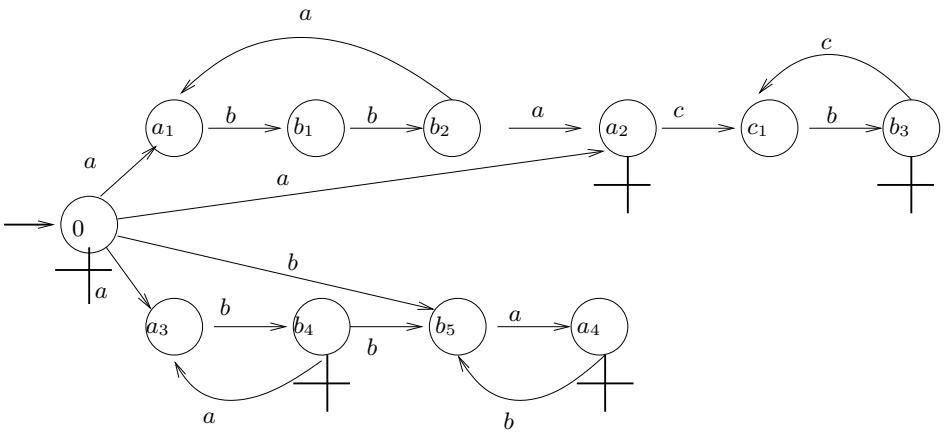

Figure 1: finite automaton for L_e

Exercice 2 [/4] We transform $\mathcal{A}$ into a normalized extended f.a. $\mathcal{A}_1$ where i (resp. t)

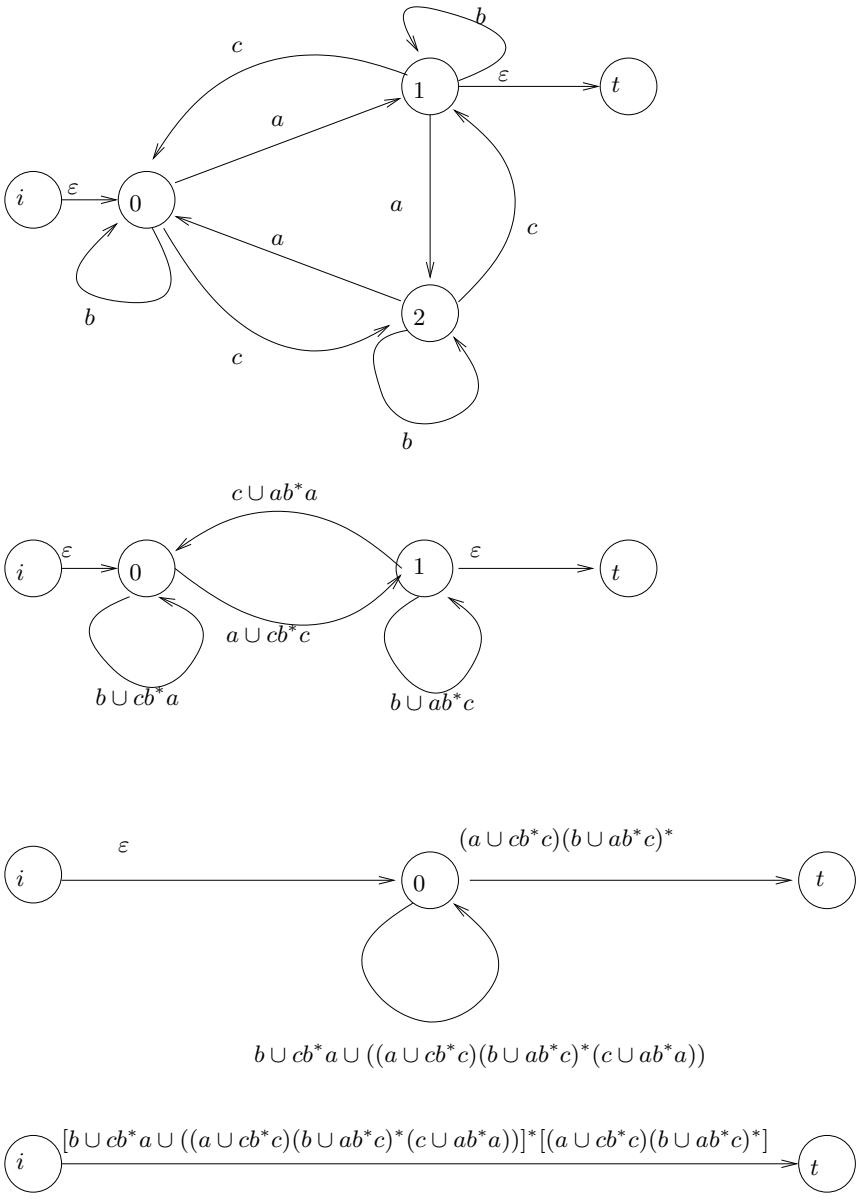

Figure 2: sequence of extended automata, ex. 2

is the initial (resp. terminal) state. We then eliminate successively states in the ordering: $2, 1, 0$. We obtain the sequence of extended automata shown on figure 2. It follows that:

$$[b \cup cb^*a \cup ((a \cup cb^*c)(b \cup ab^*c)^*(c \cup ab^*a))]^*[(a \cup cb^*c)(b \cup ab^*c)^*]$$

is a regular expression for $L_{\mathcal{A}}$.

Exercice 3 [/5] The language L_e is recognized by the following finite automaton $\mathcal{A}$: Making $\mathcal{A}$ complete and taking its complement, we obtain a finite automaton $\mathcal{B}$ recognizing

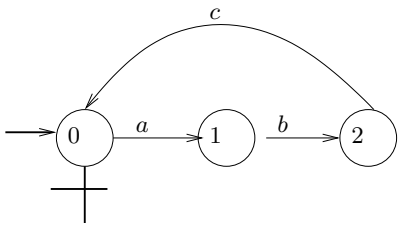

Figure 3: a non-det automaton for L_e

$\{a, b, c\}^* - L_e$. We transform $\mathcal{B}$ into a normalized extended f.a. $\mathcal{B}_1$ where i (resp. t) is the

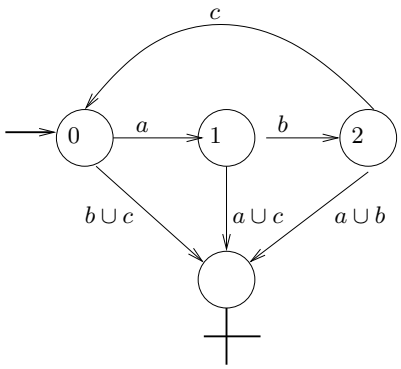

Figure 4: a det automaton for $\{a, b, c\}^* - L_e$

initial (resp. terminal) state. We then eliminate successively states in the ordering: $1, 2, 3, 0$. We obtain the sequence of extended automata shown on figure 5. It follows that:

$$(abc)^*(b \cup c \cup aa \cup ac \cup aba \cup abb)(a \cup b \cup c)^*$$

is a regular expression for $\{a, b, c\}^* - L_e$.

Exercice 4 [/4] We consider the context-free grammar $G := (A, N, R)$ where $A = \{a, b\}$, $N = \{S, T_1, T_2\}$ and R consists of the following rules:

r1: $S \to T_1$
r2: $S \to T_2$
r3: $T_1 \to aT_1$
r4: $T_1 \to aT_1b$
r5: $T_1 \to a$
r6: $T_2 \to T_2b$
r7: $T_2 \to aT_2b$
r8: $T_2 \to b$

The start symbol of G is S.
1- $\mathrm{L}(G, T_1) = \{a^p b^q \mid p > q \geq 0\}$
2- $\mathrm{L}(G, T_2) = \{a^p b^q \mid q > p \geq 0\}$

3- $\mathrm{L}(G,S) = \{a^p b^q \mid p \geq 0, q \geq 0, p \neq q\}$
4- The word $aaab$ has two different leftmost derivations:

$$S \to T_1 \to aT_1 b \to aaT_1 b \to aaab$$

$$S \to T_1 \to aT_1 \to aaT_1 b \to aaab$$

This shows that G is ambiguous.
5- The ambiguity is due to the fact that rules $r3, r4$ are commuting and, as well $r6, r7$ are commuting. Hence a word w is generated by G with *exactly one* derivation-tree iff w is generated by a derivation tree T such that:
- either T uses $r4$ but not $r3$
- or T uses $r7$ but not $r6$
- or T uses $r3$ but not $r4$
- or T uses $r6$ but not $r7$

The two first types of derivation-trees give rise to the set of words:
$\{a^p b^q \mid p \geq 1, q \geq 1, |p-q| = 1\}$
The third type of derivation-trees gives rise to the set of words: $\{a^p \mid p \geq 2\}$
The fourth type of derivation-trees gives rise to the set of words: $\{b^q \mid q \geq 2\}$.
Consequently:

$$L' = \{a^p b^q \mid p \geq 1, q \geq 1, |p-q| = 1\} \cup \{a^p \mid p \geq 2\} \cup \{b^q \mid q \geq 2\}$$

6- A non-ambigyous c.f. grammar could be, for example: $G' := (A, N', R')$ where $A = \{a, b\}$, $N' = \{S, T_1, T_2, E\}$ and R' consists of the following rules:
r'1: $S \to T_1$
r'2: $S \to T_2$
r'3: $T_1 \to aT_1$
r'4: $T_1 \to aE$
r'5: $T_1 \to a$
r'6: $T_2 \to T_2 b$
r'7: $T_2 \to Eb$
r'8: $T_2 \to b$
r'9: $E \to aEb$
r'10: $E \to ab$

Exercice 5 [/4] We consider the context-free grammar $G := (A, N, R)$ where $A = \{a, b, c\}$, $N = \{S_1, S_2, S_3, S_4, S_5\}$ and R consists of the following 12 rules:

$$\begin{array}{ccc} S_1 \to aS_1S_1 & S_1 \to bS_3S_1 & S_1 \to S_2c \\ S_2 \to S_2S_1 & S_2 \to aS_3 & S_2 \to S_1S_2S_1 \\ S_3 \to a & S_3 \to S_1S_3 & S_4 \to cS_4 \\ S_4 \to aS_4S_5 & S_5 \to aS_5 & S_5 \to aS_4a \end{array}$$

The start symbol of G is S_1.
1- We compute the subset of *productive* non-terminals of G by the fixpoint technique explained

in the lecture:

$$V_1 = \{S_3\}, V_2 = \{S_3, S_2\}.V_3 = \{S_3, S_2, S_1\}, V_4 = V_3.$$

Hence the set of productive non-terminals is $\{S_1, S_2, S_3\}$.
2- We compute the subset of *useful* non-terminals of G by the fixpoint technique explained in the lecture:

$$N_1 = \{S_1\}, N_2 = \{S_1, S_2, S_3\}, N_3 = N_2$$

Hence the set of useful non-terminals is $\{S_1, S_2, S_3\}$.
3- We can thus transform the grammar G into an equivalent grammar G' where every non-terminal is productive and useful, just by restricting both the non-terminal alphabet and the rules to the subset N_2:
$G' := (A, N', R')$ where $A = \{a, b, c\}$, $N' = \{S_1, S_2, S_3\}$ and R' consists of the following 8 rules:

$$\begin{array}{ccc} S_1 \to aS_1S_1 & S_1 \to bS_3S_1 & S_1 \to S_2c \\ S_2 \to S_2S_1 & S_2 \to aS_3 & S_2 \to S_1S_2S_1 \\ S_3 \to a & S_3 \to S_1S_3 & \end{array}$$

4- The language $\mathrm{L}(G, S_1)$ is not empty, since the non-terminal S_1 is productive.
5- We observe that:

$$S_1 \to aS_1S_1 \to^* aS_1aac \text{ and } S_1 \to^* aac$$

hence $\{a^n(aac)^{n+1}\} \subseteq \mathrm{L}(G, S_1)$, which shows that the language $\mathrm{L}(G, S_1)$ is *infinite*.
Exercice 6 [/4] We consider the two following context-free grammars $G_1 := (A, N_1, R_1), G_2 := (A, N_2, R_2)$ where $A = \{a, b, c\}$, $N_1 = \{S, T\}, N_2 = \{U\}$,R_1 consists of the rules:
r1: $S \to aSbT$
r2: $S \to cT$
r3: $T \to aTTb$
r4: $T \to c$
and R_2 consists of the rules:
r4: $U \to UUb$
r5: $U \to a$
1- The following context-free grammar generates the language $\mathrm{L}(G_1, S) \cdot \mathrm{L}(G_2, U)$:
$G_\cdot := (A, N, R_\cdot)$ where $A = \{a, b, c\}$, $N := N_1 \cup N_2 \cup \{\sigma\}$,$R_\cdot := R_1 \cup R_2 \cup \{\sigma \to ST\}$,
2- The following context-free grammar generates the language $\mathrm{L}(G_1, S) \cup \mathrm{L}(G_2, U)$:
$G_\cup := (A, N, R_\cup)$ where $A = \{a, b, c\}$, $N := N_1 \cup N_2 \cup \{\sigma\}$,$R_\cup := R_1 \cup R_2 \cup \{\sigma \to S, \sigma \to T\}$,
3- The following context-free grammar generates the language $\mathrm{L}(G_1, S)^*$:
$G_* := (A, N_*, R_*)$ where $A = \{a, b, c\}$, $N_* := N_1 \cup \{\sigma\}$,$R_* := R_1 \cup \{\sigma \to \sigma S, \sigma \to \varepsilon\}$,
4- The following context-free grammar H_1 generates $\varphi(\mathrm{L}(G_1, S))$: $H_1 := (B, N_1, R_{1,\varphi})$ where $B = \{x, y\}$, $R_{1,\varphi}$ consists of the rules:

$$S \to xySyxT, \;\; S \to yT, \;\; T \to xyTTyx, \;\; T \to y$$

The following context-free grammar H_2 generates $\varphi(\mathrm{L}(G_2, U))$: $H_2 := (B, N_2, R_{2,\varphi})$ where $B = \{x, y\}$, $R_{2,\varphi}$ consists of the rules:
$U \to UUyx, \;\; U \to xy.$

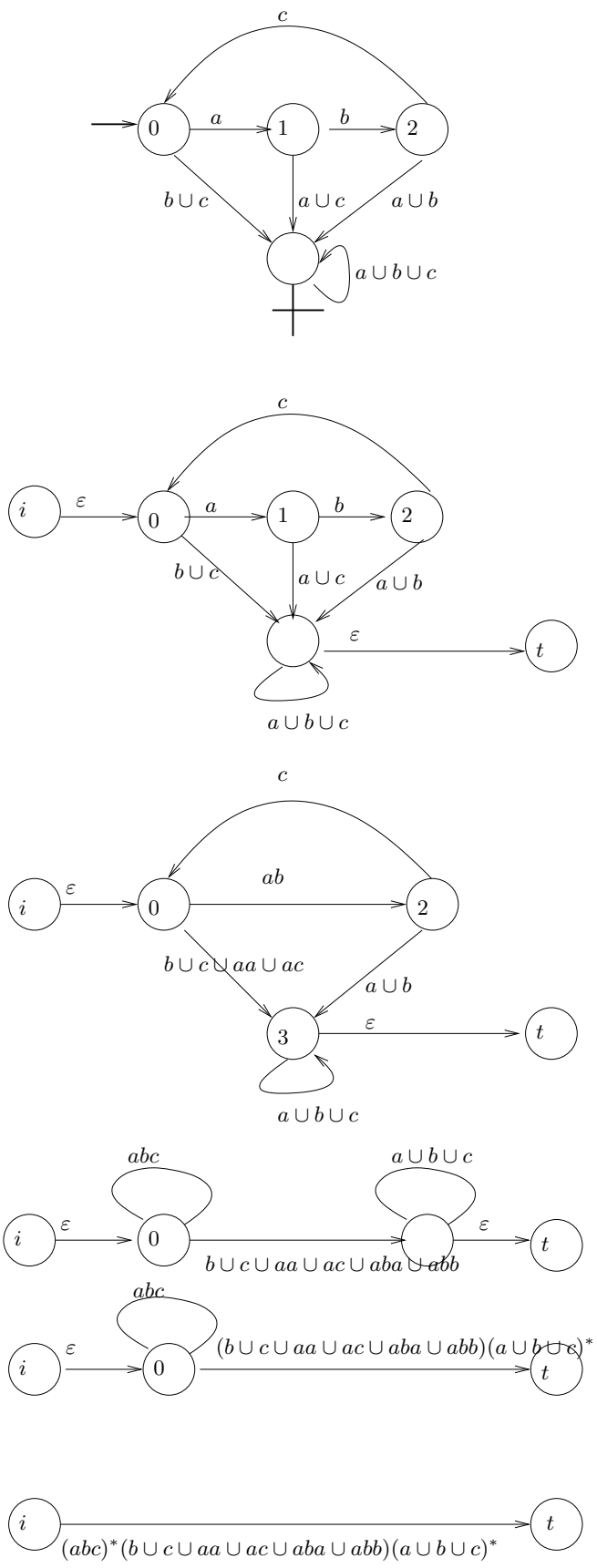

Figure 5: sequence of extended automata, ex. 3