

Langages Formels

TD 1

Isa Vialard
vialard@lsv.fr

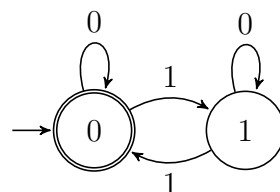
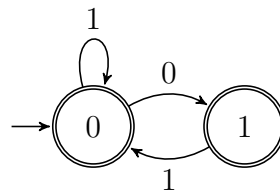
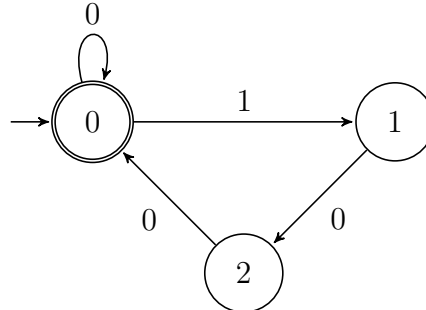
February 1, 2024

Exercise 1 : Construction d'automates

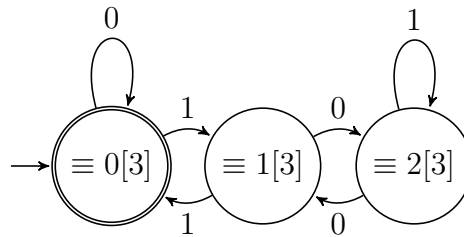
On note $\Sigma = \{0, 1\}$.

1. Donner un automate qui reconnaît les langages :

- (a) $L_1 = \{u \in \Sigma^* : \text{toute occurrence de 1 est suivie de deux occurrences de 0}\}$
- (b) $L_2 = \{u \in \Sigma^* : u \text{ ne contient pas deux occurrences successives de 0}\}$
- (c) $L_3 = \{u \in \Sigma^* : \text{le nombre d'occurrences de 1 est pair}\}$



2. Donner un automate qui reconnaît le langage des multiples de 3 en base 2 où la représentation des entiers est “big-endian” (i.e. les bits sont rangés du plus fort au plus faible).



Exercice 2 : Opérations et problèmes : questions de cours

1. Étant donné un langage reconnaissable, montrer que son complémentaire est encore reconnaissable.
2. Étant donnés deux langages reconnaissables montrer que leur union, intersection et différence sont encore des langages reconnaissables.
3. Étant donné un langage reconnaissable L , montrer que l'ensemble des suffixes des mots de L est encore reconnaissable.
4. L'appartenance du mot vide ε à un langage reconnaissable est-elle décidable ?
5. La vacuité d'un langage reconnaissable est-elle décidable ?
6. L'universalité d'un langage reconnaissable est-elle décidable ?

1. À partir d'un automate déterministe on peut construire un automate complet (il suffit de rajouter un état "puits" où vont toutes les transitions qui n'apparaissent pas sur l'automate). À partir de là, il suffit d'échanger états finals et non-finals pour obtenir un langage qui reconnaît le langage complémentaire.

2. Soit deux automates $\mathcal{A}_1 = (Q_1, i_1, F_1, \delta_1)$ et $\mathcal{A}_2 = (Q_2, i_2, F_2, \delta_2)$. Alors $(Q_1 \cup Q_2 \cup \{i\}, i, F, \delta)$, où $F = F_1 \cup F_2 \cup \{i\}$ si $i_1 \in F_1$ ou $i_2 \in F_2$, sinon $F = F_1 \cup F_2$, et $\delta = \delta_1 \cup \delta_2 \cup \{(i, a, q) | (i_1, a, q) \in \delta_1\} \cup \{(i, a, q) | (i_2, a, q) \in \delta_2\}$, reconnaît $\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$.

De même, $(Q_1 \times Q_2, (i_1, i_2), F_1 \times F_2, \delta')$ où $\delta' = \{((p_1, p_2), a, (q_1, q_2)) | (p_1, a, q_1) \in \delta_1, (p_2, a, q_2) \in \delta_2\}$, reconnaît $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.

La différence $\mathcal{L}_1 \setminus \mathcal{L}_2$ est l'intersection avec le complémentaire $\mathcal{L}_1 \cap (\mathcal{L}_2^C)$.

3. Soit $\mathcal{A} = (Q, i, F, \delta)$ un automate. Alors (Q, i, F, δ') , où $\delta' = \delta \cup \{(i, a, q) | (p, a, q) \in \delta\}$ reconnaît les suffixes de $\mathcal{L}(\mathcal{A})$ (C'est comme si on transformait tous les états en états initiaux).
4. Le mot vide est reconnu par un automate ssi son état initial est acceptant.
5. Un automate reconnaît le langage vide ssi tous ses états acceptants ne sont pas accessibles. Et l'accessibilité d'un état est décidable.

6. Un automate reconnaît Σ^* ssi son complémentaire reconnaît $\emptyset$, donc c'est décidable.

Exercise 3 : Arden Lemma

Let A, B be two languages.

1. Prove that the language $L = A^*B$ is the smallest solution to the equation:

$$X = (A \cdot X) \cup B$$

First, we prove that it is a solution:

$$\begin{aligned} (A \cdot A^*B) \cup B &= A^+B \cup A^0B \\ &= A^*B \end{aligned}$$

Let S be a solution to this equation; a first observation is that $A^0B = B \subseteq S$. By induction, for every $n \in \mathbb{N}$, $A^nB \subseteq S$. Indeed, if $A^nB \subseteq S$, then $A \cdot A^nB = A^{n+1}B \subseteq S$.

2. Prove that if $\varepsilon \notin A$, then it is the only solution.

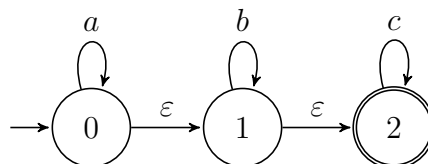
By contradiction, let S be a solution of $X = (A \cdot X) \cup B$ such that $\varepsilon \notin A$ and $A^*B \subsetneq S$. Let w be the smallest word in S which is not in A^*B .

As a consequence, $w \in (A \cdot S) \cup B$. As $w \notin B$ by hypothesis, $w = w_1 \cdot w_2$ where $w_1 \in A$ and $w_2 \in S$. By hypothesis, w_2 cannot be in A^*B so $w_2 \in S \setminus A^*B$. However, $\varepsilon \notin A$ which leads to an absurdity as w_2 is strictly shorter than w .

Exercise 4 : ε -moves

Un automate fini avec ε -transitions sur un alphabet Σ est un 5-uplet $\langle Q, I, F, \delta \rangle$ où $I, F \subseteq Q$ et $\delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times Q$. Une ε -transition (une transition étiquetée par ε) peut être empruntée sans consommer de lettres du mot d'entrée. Les notions de chemin et d'acceptation sont étendues de manière naturelle.

1. On se place sur l'alphabet $\Sigma = \{a, b, c\}$. Quel est le langage reconnu par l'automate $\mathcal{A}_0$?



Un automate $\mathcal{A}_0$ avec ε -transitions

Le langage reconnu est $a^*b^*c^*$.

Soit $\mathcal{A} = \langle Q, I, F, \delta \rangle$ un automate avec ε -transitions. Pour tout état $q \in Q$, on note $Cl(q)$ la clôture par ε -transitions de l'état q :

$$q' \in Cl(q) \Leftrightarrow \text{il existe } n \in \mathbb{N} \text{ tel que } q \xrightarrow{\varepsilon}^n q'$$

2. Quelles sont les clôtures des états de $\mathcal{A}_0$?

$$Cl(0) = \{ 0, 1, 2 \}, Cl(1) = \{ 1, 2 \}, \text{ et } Cl(2) = \{ 2 \}.$$

3. Pour tout $\mathcal{A} = \langle Q, I, F, \delta \rangle$ un automate avec ε -transitions, construire un automate sans ε -transitions $\hat{\mathcal{A}} = \langle Q, I, \hat{F}, \hat{\delta} \rangle$ qui reconnait le même langage.

On va construire un automate $\hat{\mathcal{A}} = \langle Q, I, \hat{F}, \hat{\delta} \rangle$ sans ε -transitions qui reconnait le même langage que $\mathcal{A}$. On définit $\hat{\delta}$ et $\hat{F}$ de la manière suivante. Pour $a \in \Sigma$:

$$\begin{aligned} q \in \hat{F} &\Leftrightarrow \text{ou } \begin{cases} q \in F \\ \text{il existe } q' \in Cl(q) \text{ tel que } q' \in F \end{cases} \\ (q, a, q') \in \hat{\delta} &\Leftrightarrow q' \in \bigcup_{p \in P_{q,a}} Cl(p) \\ \text{where } P_{q,a} &= \{ p : \exists r \in Cl(q), (r, a, p) \in \delta \} \end{aligned}$$

On opère une induction sur un mot w , avec pour hypothèse d'induction que s'il existe un chemin dans $\mathcal{A}$ étiqueté par w de la forme $c = q_1 \xrightarrow{a_1} \delta \dots \xrightarrow{a_{n-1}} \delta q_n$, alors il existe un chemin dans $\hat{\mathcal{A}}$ étiqueté par c de la forme $s_1 \xrightarrow{b_1} \hat{\delta} \dots \xrightarrow{b_{k-1}} \hat{\delta} s_k$ tel que $s_1 = q_1$ et $q_n \in Cl(s_k)$.

Si $w = \varepsilon : a_1 = \dots = a_{n-1} = \varepsilon$, donc $q_n \in Cl(q_1)$ et le chemin q_1 satisfait les hypothèses d'induction.

Si $w = aw' : \text{Il existe } i \leq n \text{ tel que pour tout } j < i, a_j = \varepsilon, \text{ et } a_i = a.$

On applique l'hypothèse d'induction à w' et au chemin $q_{i+1} \xrightarrow{a_{i+1}} \delta \dots \xrightarrow{a_{n-1}} \delta q_n$. On obtient un chemin $c = q_{i+1} = s_1 \xrightarrow{b_1} \delta \dots \xrightarrow{b_k} \delta s_k$ dans $\hat{\mathcal{A}}$ étiqueté par w' , commençant par q_{i+1} et terminant par un état s_k tel que $q_n \in Cl(s_k)$.

On a de plus que $q_i \in Cl(q_1)$, et $(q_i, a, q_{i+1}) \in \delta$. Ainsi, $(q_1, a, q_{i+1}) \in \hat{\delta}$. En juxtaposant le chemin $q_1 \xrightarrow{a} \delta q_{i+1}$ et c , on obtient un chemin de la bonne forme.

On conclut. Soit w un mot du langage reconnu par $\mathcal{A}$. Alors il existe un chemin $i = q_1 \xrightarrow{a_1} \delta \dots \xrightarrow{a_{n-1}} \delta q_n \in F$. Alors il existe un chemin $i = s_1 \xrightarrow{b_1} \delta \dots \xrightarrow{b_{k-1}} \delta s_k$ dans $\hat{\mathcal{A}}$ étiqueté par w' tel que $q_n \in Cl(s_k)$. Donc $s_k \in \hat{F}$ et ce chemin est acceptant.

4. Donner un automate déterministe qui reconnaît le même langage que l'automate $\mathcal{A}_0$.

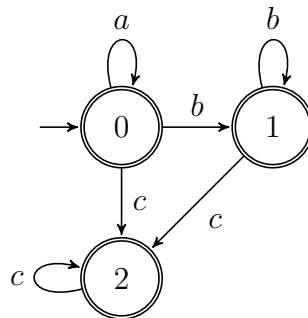


Figure 2: Automate $\mathcal{A}_0$ déterminisé

Contrôle continu 1 - À rendre pour le 01/02

Chaque semaine vous aurez un contrôle continu pour la semaine suivante, à rendre au début du TD. Vous en rendez au moins trois, et votre note sera la moyenne des trois meilleurs.

Exercice 5 : Plus long préfixe commun

Étant donné un automate reconnaissant un langage L non vide, montrer que l'on peut calculer le plus long préfixe commun à tous les mots de L .

On prend un automate émondé déterministe pour L , on peut de plus supposer qu'il a un seul état initial. À partir de cet état on avance tant qu'il existe un unique état successeur, et que l'état courant n'est pas lui-même acceptant. Quand on s'arrête on a lu ce préfixe maximal. Et si on ne s'arrête pas, c'est qu'il n'y a aucun mot dans le langage.

Exercice 6 : Star-free languages

Let $\Sigma = \{a, b\}$. The family of *star-free languages* is the smallest family including $\emptyset$, $\{\ell\}$ for every letter $\ell \in \Sigma$, and closed by union, complementation et concatenation.

- Express the following languages as star-free languages:

(a) $\{\varepsilon\}$

$$\{\varepsilon\} = (\Sigma \cdot \emptyset^c)^c$$

(b) words containing the factor bab

$$\emptyset^c \cdot bab \cdot \emptyset^c$$

- Express $L_1 = (ab)^*$ as a star-free language.

$$L_1 = \{\varepsilon\} + [(a \cdot \emptyset^c \cdot b) \cap (\emptyset^c \cdot (aa + bb) \cdot \emptyset^c)^c]$$

ou

$$L_1 = [\emptyset^c \cdot (aa + bb) \cdot \emptyset^c \cup b \cdot \emptyset^c \cup \emptyset^c \cdot a]^c$$

3. Is $L_2 = (b^* + ab)^* + (b^* + ab)^*a$ a star-free language?

$$L_2 = (\emptyset^c \cdot aa \cdot \emptyset^c)^c$$