Informatique – MPI

Romain Bricout

10 octobre 2023

Introduction

Ce document réunit l'ensemble de mes cours d'Informatique de MPI, ainsi que les exercices les accompagnant. Le professeur était M. Carcenac. J'ai adapté certaines formulations me paraissant floues ou ne me plaisant pas mais le contenu pur des cours est strictement équivalent.

Les éléments des tables des matières initiale et présentes au début de chaque chapitre sont cliquables (amenant directement à la partie cliquée). C'est également le cas des références à des éléments antérieurs de la forme, par exemple, « Démonstration 5.22 ».

Table des matières

Première partie

Cours

Chapitre 1

Langages réguliers

Sommaire

Introduction

Cf. diaporama.

Dans ce cours, on nommera « langages réguliers » et « langages rationnels » les mêmes objets.

1.1 Langage formel

1.1.1 Alphabet, mot

Définition 1.1
Un alphabet Σ est un ensemble fini de symboles.

Un mot (sur Σ) est une suite finie de symboles (de Σ).

Exemple 1.2
$m = m_1 m_2 \dots m_n$ est un mot de n symboles.

On note $m = \varepsilon$ le mot vide.

Notation 1.3
L'ensemble de tous les mots (sur Σ) est noté Σ^* (étoile de Kleene).

Notation 1.4
On note $|m|$ la longueur du mot m.

Proposition 1.5
On a $|m_1 m_2 \dots m_n| = n$ *et* $|\varepsilon| = 0$.

Notation 1.6
On note $|m|_x$ le nombre d'occurrences de $x \in \Sigma$ dans m.

Définition 1.7 (Concaténation)
Soient $u, v \in \Sigma^*$.

On note $u.v$ ou uv la concaténation de u et v.

On pose

$$\varepsilon\varepsilon = \varepsilon \qquad \varepsilon u = u\varepsilon = u.$$

Proposition 1.8
Soient $u, v \in \Sigma^*$ *et* $x \in \Sigma$.

On a

$$|uv| = |u| + |v| \qquad |uv|_x = |u|_x + |v|_x .$$

Remarque 1.9 (HP, lien avec les structures algébriques)
La concaténation est une loi de composition interne sur Σ^*.

Elle est non-commutative : on n'a pas $\forall u, v \in \Sigma^*,\ uv = vu$.

Elle est associative : on a $\forall u, v, w \in \Sigma^*,\ u\,(vw) = (uv)\,w$.

Elle admet ε comme élément neutre.

On obtient alors que $(\Sigma^*, .)$ est un monoïde.

Définition 1.10 (Puissance)
Soient $u \in \Sigma^*$ et $n \in \mathbb{N}^*$.

On définit $u^n = \underbrace{uu\dots u}_{n \text{ facteurs}}$ et on pose $u^0 = \varepsilon$.

Remarque 1.11
Pour $x, y, z \in \Sigma^*$, on a $(xyz)^2 = xyzxyz \neq x^2y^2z^2$.

Définition 1.12
Soit $m \in \Sigma^*$.

- $u \in \Sigma^*$ est un préfixe de m ssi il existe $v \in \Sigma^*$ tel que $m = uv$.
  $u \in \Sigma^*$ est un préfixe propre de m ssi il existe $v \in \Sigma^* \setminus \{\varepsilon\}$ tel que $m = uv$.

- $u \in \Sigma^*$ est un suffixe de m ssi il existe $v \in \Sigma^*$ tel que $m = vu$.
  $u \in \Sigma^*$ est un suffixe propre de m ssi il existe $v \in \Sigma^* \setminus \{\varepsilon\}$ tel que $m = vu$.

- $u \in \Sigma^*$ est un facteur de m ssi il existe $v, w \in \Sigma^*$ tel que $m = vuw$.
  $u \in \Sigma^*$ est un facteur propre de m ssi il existe $v, w \in \Sigma^* \setminus \{\varepsilon\}$ tel que $m = vuw$.

- $u \in \Sigma^*$ est un sous-mot de m si c'est une suite extraite de m (*cf.* Exercice 1.7 pour une définition formelle).

Exemple 1.13
Cf. exercices.

Remarque 1.14
ε est un préfixe, un suffixe, un facteur et un sous-mot de tout mot $m \in \Sigma^*$.

Remarque 1.15
Soient $u, m \in \Sigma^*$.

On a les implications suivantes :

$$u \text{ est un préfixe de } m \implies u \text{ est un facteur de } m$$

$$u \text{ est un suffixe de } m \implies u \text{ est un facteur de } m$$

$$u \text{ est un facteur de } m \implies u \text{ est un sous-mot de } m$$

Notation 1.16
Si $u \in \Sigma^*$ est un préfixe de $m \in \Sigma^*$, on note $u \leqslant_p m$.

1.1.2 Langage

Définition 1.17
Un langage L (sur Σ) est un ensemble de mots, *i.e.* une partie de Σ^*.

Exemple 1.18
On pose $\Sigma = \{a, b\}$. On peut alors définir les langages sur Σ suivants :

- $L_1 = \{ab, aab, aaab\}$
- $L_2 = \{a^n \mid n \in \mathbb{N}^*\}$
- $L_3 = \varnothing$
- $L_4 = \{\varepsilon\}$
- $L_5 = \{\text{mots qui contiennent un nombre pair de } a\}$
  $= \{m \in \Sigma^* \mid |m|_a \equiv 0\,[2]\}$
  $\neq \{(aa)^n \mid n \in \mathbb{N}\}$
- $L_6 = \{m \in \Sigma^* \mid |m|_a = |m|_b\}$

Si on considère maintenant l'ensemble des symboles ASCII comme alphabet, alors toute chaîne de caractères ASCII est un mot. Par exemple, tout programme informatique est un mot.

Définition 1.19 (Opérations sur les langages)
Soient Σ un alphabet et $L_1, L_2 \in \mathscr{P}(\Sigma^*)$.

On définit les langages ci-dessous :

- union : $L_1 \cup L_2 = \{m \in \Sigma^* \mid m \in L_1 \text{ ou } m \in L_2\}$;
- intersection : $L_1 \cap L_2 = \{m \in \Sigma^* \mid m \in L_1 \text{ et } m \in L_2\}$;
- différence : $L_1 \setminus L_2 = \{m \in \Sigma^* \mid m \in L_1 \text{ et } m \notin L_2\}$;
- différence symétrique : $L_1 \,\Delta\, L_2 = \{m \in \Sigma^* \mid m \in L_1 \text{ et } m \notin L_2 \text{ ou } m \notin L_1 \text{ et } m \in L_2\}$;
- concaténation : $L_1 L_2 = \{uv \mid (u, v) \in L_1 \times L_2\}$.

Exemple 1.20
On pose $\Sigma = \{a, b\}$, $L_1 = \{a, ab, \varepsilon\}$ et $L_2 = \{b, a, \varepsilon\}$.

On a $L_1 L_2 = \{ab, aa, abb, aba, b, a, \varepsilon\}$.

De plus, on a $L_1^2 = \{aa, aab, a, aba, abab, ab, \varepsilon\} \neq \{aa, abab, \varepsilon\}$.

Définition 1.21
Soit $L \in \mathscr{P}(\Sigma^*)$.

Pour $n \in \mathbb{N}^*$, on définit $L^n = \{u_1 u_2 \dots u_n \mid \forall i \in [\![1\,;n]\!],\ u_i \in L\}$ et on pose $L^0 = \{\varepsilon\} \neq \varnothing$.

On définit $L^* = \bigcup_{n \in \mathbb{N}} L^n$.

On a les propriétés suivantes :

- $L^0 \subseteq L^*$ donc $\varepsilon \in L^*$;
- Pour tout $n \in \mathbb{N}$, $L^n \subseteq L^*$;
- $m \in L^* \iff \left| \begin{array}{l} \exists n \in \mathbb{N}^*,\ \exists (u_1, \dots, u_n) \in L^n,\ m = u_1 \dots u_n \\ \text{ou} \\ m = \varepsilon \end{array} \right.$

On définit $L^+ = \bigcup_{n \in \mathbb{N}^*} L^n$.

On a $L^+ \subseteq L^*$.

Remarque 1.22
La définition de Σ^* est cohérente en voyant Σ comme un langage.

Proposition 1.23
Soient Σ un alphabet et $L, L_1, L_2 \in \mathscr{P}(\Sigma^*)$.

On a :

- $L.\varnothing = \varnothing.L = \varnothing$;
- $L.\{\varepsilon\} = \{\varepsilon\}.L = L$;
- $L^n.L^m = L^m.L^n = L^{n+m}$;
- $L.(L_1 \cup L_2) = L.L_1 \cup L.L_2$;

- $L.(L_1 \cap L_2) \subseteq L.L_1 \cap L.L_2$;
- $L^*.L = L.L^* = L^+$;
- $L^* = L^+ \cup \{\varepsilon\}$;
- $(L^*)^* = L^*$

Démonstration 1.24
Cf. Exercice 1.9. ■

1.2 Langage régulier

Soit Σ un alphabet.

Définition 1.25
On définit par induction l'ensemble des langages réguliers LRat et l'ensemble des expressions régulières (représentation symbolique des langages) R.

Cas de base :

$$\varnothing \in \mathrm{LRat} \qquad \{\varepsilon\} \in \mathrm{LRat} \qquad \{a \mid a \in \Sigma\} \in \mathrm{LRat}\,.$$

Cas induits : pour $(L_1, L_2) \in \mathrm{LRat}^2$, on a les propriétés de stabilité suivantes :

$$L_1 \cup L_2 \in \mathrm{LRat} \qquad L_1 L_2 \in \mathrm{LRat} \qquad L_1^* \in \mathrm{LRat}\,.$$

De même, on a les cas de bases :

$$\varnothing \in \mathrm{R} \qquad \varepsilon \in \mathrm{R} \qquad a \in \mathrm{R} \text{ avec } a \in \Sigma$$

et les cas induits : pour $(r_1, r_2) \in \mathrm{R}^2$, on a les propriétés de stabilité suivantes :

$$r_1 \mid r_2 \in \mathrm{R} \qquad r_1 r_2 \in \mathrm{R} \qquad r_1^* \in \mathrm{R}\,.$$

Remarque 1.26 (HP)
LRat et R sont isomorphes.

Définition 1.27
On dit qu'une expression régulière dénote un langage régulier et on note $\mathscr{L}(r)$ le langage dénoté par $r \in \mathrm{R}$.

Exemple 1.28
On pose $\Sigma = \{a, b\}$ et $r = a\,(a \mid b)^* \in \mathrm{R}$.

Alors $\mathscr{L}(r) = \{a\} \,.\, (\{a\} \cup \{b\})^*$ est le langage contenant les mots qui commencent par un a.

Proposition 1.29
Tout langage contenant un unique mot est régulier.

Exemple 1.30
On a $\{aba\} = \{a\} \,.\, \{b\} \,.\, \{a\} \in \mathrm{LRat}$.

Proposition 1.31
Tout langage fini est régulier.

Exemple 1.32
On a $\{aa, abc, c\} = \{aa\} \cup \{abc\} \cup \{c\} \in \mathrm{LRat}$.

Proposition 1.33
Le langage Σ^* est régulier.

Démonstration 1.34
L'alphabet Σ un langage fini donc c'est un langage régulier donc Σ^* est régulier. ■

Définition 1.35
On dit qu'une expression régulière $r \in \mathrm{R}$ s'apparie à (ou « match ») tout mot de $\mathscr{L}(r)$.

Quelques cas particuliers :

- a s'apparie au mot a ;
- $r_1 r_2$ s'apparie à un mot de $\mathscr{L}(r_1)$ concaténé à un mot de $\mathscr{L}(r_2)$;
- $r_1 \mid r_2$ s'apparie à un mot de $\mathscr{L}(r_1)$ ou à un mot de $\mathscr{L}(r_2)$;
- r^* s'apparie à une succession de mots de $\mathscr{L}(r)$ (incluant la succession de zéro mot ε) ;

- ε s'apparie au mot vide ε ;
- $\varnothing$ ne s'apparie à aucun mot ;
- Σ s'apparie à n'importe quel symbole ;
- Σ^* s'apparie à n'importe quel mot.

Remarque 1.36 (Priorités)
S'appliquent d'abord les étoiles, puis les concaténations et enfin les unions. On peut parenthéser les expressions si nécessaire.

Exemple 1.37
Avec $\Sigma = \{a, b\}$:

- on pose $r_1 = (a \mid b)^*$.
  On a $\mathscr{L}(r_1) = \{\text{mots composés de } a \text{ ou de } b\}$;
  $= \{a, b\}^*$
- on pose $r_2 = aa^*$.
  On a $\mathscr{L}(r_2) = \{a^n \mid n \in \mathbb{N}^*\}$;
- on pose $r_3 = (aa)^*$.
  On a $\mathscr{L}(r_3) = \left\{a^{2k} \;\middle|\; k \in \mathbb{N}\right\}$;
- on pose $r_4 = a\Sigma^*$.
  On a $\mathscr{L}(r_4) = \{\text{mots commençant par } a\}$.

Avec $\Sigma = \{a, \ldots, z, \_\}$ (où _ désigne une espace) :

- l'expression régulière $ch\,(at \mid ien)$ s'apparie aux mots *chat* et *chien* ;
- l'expression régulière $(tres\_)^*\,bien$ s'apparie aux mots *bien*, *tres_bien*, *tres_tres_bien*, ...

Chapitre 2

Automates finis

⋆⋆ À venir ⋆⋆

Chapitre 3

Théorème de Kleene

⋆⋆ À venir ⋆⋆

Deuxième partie

Exercices

Chapitre 1

Langages réguliers

Sommaire

Illustrations du cours

Exercice 1.1 (Exercice 1, vocabulaire sur les mots)
(1) Donner tous les préfixes propres du mot $babbb$.

(2) Donner tous les facteurs du mot $babbb$.

(3) Donner tous les sous-mots de $babbb$.

Correction 1.2
⋆⋆ À venir ⋆⋆

Exercice 1.3 (Exercice 2, opérations sur les langages)
On considère un alphabet $\Sigma = \{a, b\}$ et deux langages $L_1 = \{\varepsilon, a, ba\}$ et $L_2 = \{a, aa\}$.

Donner les éléments des langages suivants : $L_1 \cup L_2$, L_1^0, L_1^1, L_1L_2 et L_2^3.

Correction 1.4
⋆⋆ À venir ⋆⋆

Exercice 1.5 (Exercice 3, langages réguliers, expressions régulières)
On considère l'alphabet $\Sigma = \{a, b\}$.

Représenter par une expression régulière :

(1) l'ensemble des mots composés de symboles a puis de symboles b ;

(2) l'ensemble des mots qui terminent par b ;

(3) l'ensemble des mots qui contiennent trois symboles a consécutifs.

Exprimer en français l'ensemble des mots dénoté par l'expression régulière :

(4) $r_4 = a\,(a \mid b)^*\,a$;

(5) $r_5 = (a \mid b)\,(a \mid b)\,(a \mid b)$;

(6) $r_6 = a^*ba^*$.

Correction 1.6
⋆⋆ À venir ⋆⋆

Compléments du cours

Exercice 1.7 (Exercice 4, sous-mot)
Formaliser la notion de sous-mot en utilisant :

(1) une suite de mots et de lettres ;

(2) une suite d'indices strictement croissante ;

(3) une fonction strictement croissante à valeurs entières.

Correction 1.8
⋆⋆ À venir ⋆⋆

Exercice 1.9 (Exercice 5, propriétés sur les langages)
Soient Σ un alphabet et L, L_1, L_2 des langages sur Σ.

Démontrer les propriétés suivantes :

(1) $L.\varnothing = \varnothing.L = \varnothing$

(2) $L.\{\varepsilon\} = \{\varepsilon\}.L = L$

(3) $L^n.L^m = L^{n+m}$

(4) $L.(L_1 \cup L_2) = L.L_1 \cup L.L_2$

(5) $L^*.L = L.L^* = L^+$

(6) $L^* = L^+ \cup \{\varepsilon\}$

(7) $(L^*)^* = L^*$

(8) $L.(L_1 \cap L_2) \subseteq L.L_1 \cap L.L_2$ et trouver un contre-exemple pour l'inclusion réciproque.

Correction 1.10
⋆⋆ À venir ⋆⋆

Exercices

Exercice 1.11 (Exercice 6, quelques questions sur les mots)
Soit Σ un alphabet.

(1) Soient $x, y, z \in \Sigma^*$. Montrer que $xy = xz \iff y = z$ et que $yx = zx \iff y = z$.

(2) Soient $u, v, w \in \Sigma^*$ tels que $u \leqslant_p w$ et $v \leqslant_p w$. Montrer que $u \leqslant_p v$ ou $v \leqslant_p u$.

(3) Soient $a, b \in \Sigma$ et $u \in \Sigma^*$ tels que $au = ub$. Montrer que $a = b$ et $u \in \{a\}^*$.

(4) Soient $x, y, u, v \in \Sigma^*$ tels que $uv = xy$. Montrer qu'il existe un unique mot $t \in \Sigma^*$ tel que $u = xt$ et $y = tv$ ou $x = ut$ et $v = ty$.

Correction 1.12
⋆⋆ À venir ⋆⋆

Exercice 1.13 (Exercice 7, dénombrement)
Soit Σ un alphabet. On considère un mot $m \in \Sigma^*$ de n symboles distincts.

(1) Combien le mot m a-t-il de préfixes, suffixes, facteurs et sous-mots ?

(2) Que peut-on dire si les symboles ne sont pas supposés distincts ?

Correction 1.14
⋆⋆ À venir ⋆⋆

Exercice 1.15 (Exercice 8, préfixe : une relation d'ordre)
Soit Σ un alphabet.

(1) Justifier que $\leqslant_p$ définit une relation d'ordre sur Σ^*.

(2) Cet ordre est-il total ?

(3) Quel est l'ordre strict associé ?

Correction 1.16
⋆⋆ À venir ⋆⋆

Exercice 1.17 (Exercice 9, racine carrée d'un langage)
Soit Σ un alphabet. Pour $L \in \mathscr{P}(\Sigma^*)$, on définit $\sqrt{L} = \left\{u \in \Sigma^* \;\middle|\; u^2 \in L\right\}$.

(1) Que vaut $\sqrt{L}$ pour $L = \{\varepsilon, a, b, aa, ab, bbb, bbbb\}$?

(2) Montrer que pour tout $L \in \mathscr{P}(\Sigma^*)$, $L \subseteq \sqrt{L^2}$. Trouver un contre-exemple montrant qu'il n'y a pas égalité.

Correction 1.18
⋆⋆ À venir ⋆⋆

Exercice 1.19 (Exercice 10, quotient à gauche d'un langage)
Soit Σ un alphabet. Pour $L \in \mathscr{P}(\Sigma^*)$ et $a \in \Sigma$, on définit $a^{-1}L = \{u \in \Sigma^* \mid au \in L\}$.

(1) Que vaut $b^{-1}L$ pour $L = \{\varepsilon, a, b, ab, bab, bbbb\}$?

Pour $R, L \in \mathscr{P}(\Sigma^*)$, on définit $R^{-1}L = \{u \in \Sigma^* \mid \exists r \in R,\ \ ru \in L\}$.

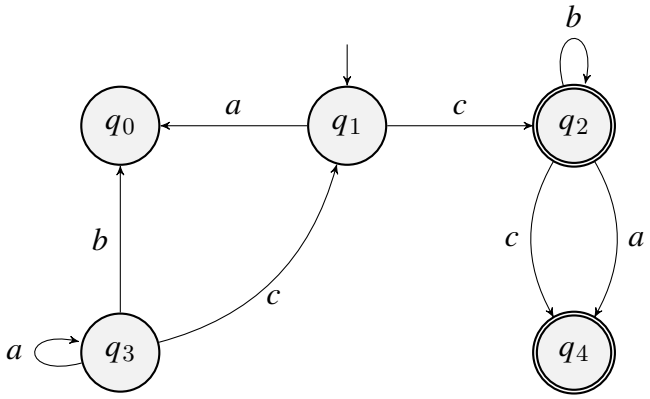
(2) Que vaut $R^{-1}L$ pour $R = \{a, bb\}$ et $L = \{\varepsilon, a, b, aa, ab, bab, bbbb\}$?

Correction 1.20
⋆⋆ À venir ⋆⋆

Chapitre 2

Automates finis

Exercice 2.1 (Exercice 1, exemple)
On considère l'automate A_0 ci-dessous :



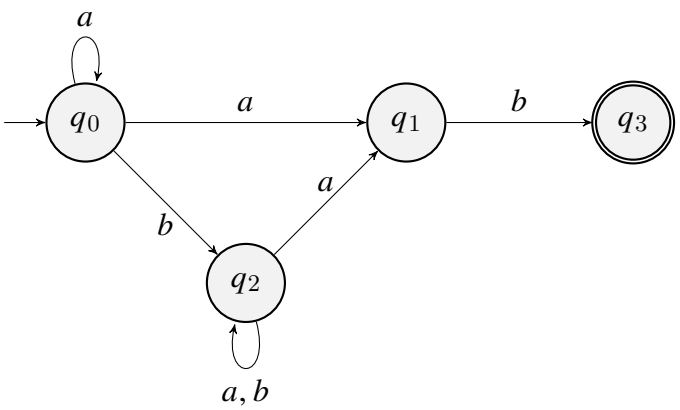
(1) Utiliser le vocabulaire approprié pour décrire l'automate A_0. En particulier, donner sa définition formelle et représenter sa table de transition.

(2) Donner quelques exemples de mots acceptés par l'automate et de mots rejetés par l'automate.

(3) Donner son équivalent complet puis l'émonder.

Correction 2.2
⋆⋆ À venir ⋆⋆

Exercice 2.3 (Exercice 2, déterminisation)
On considère l'automate A_1 ci-dessous :



(1) Justifier que l'automate A_1 n'est pas déterministe.

(2) Donner un mot m pouvant donner lieu à au moins deux calculs, l'un aboutissant à un état final, l'autre aboutissant à un état non-final. Ce mot est-il reconnu par l'automate ?

(3) Déterminer cet automate pour obtenir $A_{1\ \mathrm{det}}$. Le mot m est-il reconnu par $A_{1\ \mathrm{det}}$?

Correction 2.4
⋆⋆ À venir ⋆⋆

Exercice 2.5 (Exercice 3, automates « évidents »)
On considère l'alphabet $\Sigma = \{a, b, c\}$.

Pour chaque ensemble de mots L_i, donner une expression régulière r_i dénotant L_i et dessiner un automate A_i (déterministe ou non) reconnaissant L_i :

(1) L_1 : les mots de trois lettres qui commencent par a ;

(2) L_2 : les mots qui ne contiennent qu'un seul b ;

(3) L_3 : les mots qui ont ab pour préfixe ;

(4) L_4 : les mots qui ont bac pour suffixe ;

(5) L_5 : les mots qui ont $abba$ pour facteur ;

(6) L_6 : les mots qui ont $abba$ pour sous-mot.

Correction 2.6
⋆⋆ À venir ⋆⋆

Exercice 2.7 (Exercice 4, donner un sens aux états)
On considère l'alphabet $\Sigma = \{0, 1\}$. Représenter un automate reconnaissant :

(1) l'ensemble des mots contenant un nombre pair de symboles ;

(2) l'ensemble des mots tels que le nombre d'occurrences de 1 soit multiple de 3 ;

(3) l'ensemble des mots m tels que $|m|_0 - |m|_1 \equiv 1\ [9]$.

Correction 2.8
⋆⋆ À venir ⋆⋆

Exercice 2.9 (Exercice 5, automate déterministe complet)
On définit un automate déterministe incomplet comme un quintuplet

$$A = (Q, \Sigma, q_i, Q_F, \delta : D \subseteq Q \times \Sigma \longrightarrow Q)\,.$$

(1) Définir formellement l'automate complet A_c associé.

(2) Justifier formellement que ces deux automates sont équivalents.

Correction 2.10
⋆⋆ À venir ⋆⋆

Exercice 2.11 (Exercice 6, automate produit)
On considère l'alphabet $\Sigma = \{a, b\}$.

(1) Représenter un automate déterministe A reconnaissant les mots ayant un nombre pair de symboles.

(2) Représenter un automate déterministe A' reconnaissant les mots composés de symboles a puis de symboles b.

(3) En utilisant l'automate produit, représenter un automate reconnaissant les mots contenant un nombre pair de symboles qui sont composés de symboles a puis de symboles b. Donner un sens à chaque état de l'automate obtenu.

Correction 2.12
⋆⋆ À venir ⋆⋆

Exercice 2.13 (Exercice 7, déterminisation « explosive »)
On considère l'alphabet $\Sigma = \{a, b\}$ et $n \in \mathbb{N}^*$. On pose L_n le langage des mots ayant un a « n lettres avant la fin du mot ». Par exemple, L_1 est le langage des mots ayant un a une lettre avant la fin, *i.e.* en dernière lettre ; L_2 est le langage des mots ayant un a deux lettres avant la fin, *i.e.* en avant-dernière lettre.

(1) Donner une expression régulière e_n dénotant L_n.

(2) Donner un automate A_n non-déterministe à $n+1$ états qui reconnaît L_n. On nommera $q_0, \dots, q_n$ les états.

(3) Déterminiser A_3. Combien d'états l'automate obtenu a-t-il ?

(4) Trouver un mot de trois lettres permettant, depuis q_0, d'atteindre exactement l'ensemble des états $\{q_0, q_2\}$.

Plus généralement, on va montrer que l'automate déterminisé associé à A_n possède 2^n états.

Pour $m \in \Sigma^*$ et q un état, on note $A(m)$ l'ensemble des états de A_n accessibles à la lecture du mot m et $L_{\longrightarrow q}$ le langage reconnu par l'automate A_n légèrement modifié de sorte que q soit son unique état final.

(5) Décrire en français les langages $L_{\longrightarrow q_0}, L_{\longrightarrow q_1}, \dots, L_{\longrightarrow q_n}$.

(6) Montrer que pour $m \in \Sigma^*$, $A(m) = \left\{q \;\middle|\; m \in L_{\longrightarrow q}\right\}$.

(7) Montrer que pour chaque ensemble d'états Q contenant q_0, il existe un mot m tel que $|m| = n$ et $A(m) = Q$.

(8) En déduire, le nombre d'états de l'automate déterminisé associé à A_n.

Peut-être que la méthode de déterminisation construit un automate inutilement compliqué... Est-il possible de trouver un automate déterministe de moins de 2^n états qui reconnaisse L_n ? Ce qui suit montre que non.

Supposons qu'il existe un automate déterministe complet ayant strictement moins de 2^n états qui reconnaît L_n.

(9) Justifier qu'il existe deux mots u et v de longueur n dont la lecture aboutit au même état.

(10) En considérant la première lettre qui différencie ces deux mots, et en les rallongeant, aboutir à une contradiction.

Correction 2.14
⋆⋆ À venir ⋆⋆

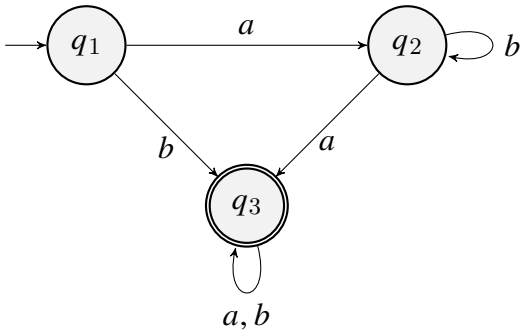
Exercice 2.15 (Exercice 8, lemme d'Arden et application)
Lemme d'Arden : si L est un langage vérifiant l'égalité $L = E.L \cup F$ où E, F sont des langages tels que $\varepsilon \notin E$, alors $L = E^*.F$.

On peut rencontrer la formulation suivante : $E^*.F$ est l'unique solution de l'équation $X = E.X \cup F$ d'inconnue X, lorsque $\varepsilon \notin E$.

(1) Démontrer le lemme d'Arden (version équation). Montrer que le langage proposé est bien solution de l'équation puis montrer que c'est l'unique solution de l'équation.

Application : le lemme d'Arden peut servir à exprimer le langage reconnu par un automate sous forme d'expression régulière.

Exemple : on considère l'automate A ci-dessous :



Pour $i \in \{1, 2, 3\}$, on note L_i le langage reconnu en prenant q_i pour état initial.

(2) Pour chaque état q_i, établir une relation entre les L_i.

(3) À l'aide du lemme d'Arden, donner une expression pour les langages L_i.

(4) En déduire le langage reconnu par l'automate.

Correction 2.16
⋆⋆ À venir ⋆⋆

Exercice 2.17 (Exercice 9, langages non-rationnels)
Montrer que les langages ci-dessous ne sont pas rationnels :

(1) $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$;

(2) $L_2 = \{a^p \mid p \text{ est premier}\}$;

(3) $L_3 = \{m \mid |m|_a = |m|_b\}$.

Correction 2.18
⋆⋆ À venir ⋆⋆

Exercice 2.19 (Exercice 10, vers l'automate de Glushkov : automate local)
Montrer que la procédure de construction de l'automate local A_{loc} associé à un langage local LL est correcte, c'est-à-dire que $\mathscr{L}(A_{\text{loc}}) = LL$.

Correction 2.20
⋆⋆ À venir ⋆⋆

Exercice 2.21 (Exercice 11, vers l'automate de Glushkov : ensembles « caractéristiques »)
Pour les langages suivants, déterminer les ensembles caractéristiques $P(L)$, $S(L)$, $F(L)$ et $N(L)$, puis déterminer si le langage est local :

(1) $L_1 = \mathscr{L}(abab)$;

(2) $L_2 = \mathscr{L}(abc)$;

(3) $L_3 = \mathscr{L}(a^*)$;

(4) $L_4 = \mathscr{L}\left((ab)^*\right)$;

(5) $L_5 = \mathscr{L}\left((ab)^* a^*\right)$.

Correction 2.22
⋆⋆ À venir ⋆⋆

Exercice 2.23 (Exercice 12, vers l'automate de Glushkov : propriétés des langages linéaires)

(1) Soit L un langage local. Exprimer $P(L^*)$, $S(L^*)$ et $F(L^*)$ en fonction de $P(L)$, $S(L)$ et $F(L)$; montrer que L^* est un langage local.

(2) De même, montrer que l'union de deux langages locaux sur des alphabets distincts est un langage local.

(3) De même, montrer que la concaténation de deux langages locaux sur des alphabets distincts est un langage local.

(4) Montrer que $\varnothing$, $\{\varepsilon\}$ et $\{a\}$ avec $a \in \Sigma$ sont des langages locaux.

(5) En déduire que le langage dénoté par une expression régulière linéaire est local.

Correction 2.24

⋆⋆ À venir ⋆⋆

Chapitre 3

Théorème de Kleene

⋆⋆ À venir ⋆⋆