Cours de combinatoire de M. Arfi

FMdKdD
fmdkdd [à] free.fr

Université du Havre
Année 2010–2011

Table des matières

Chapitre 1

Codes

Soit l'alphabet $\Gamma = \{\alpha, \beta, \gamma\}$. On voudrait pouvoir coder les mots de Γ^* en utilisant la partie $X = \{a, ab, ba\}$ de $\Sigma^* = (a + b)^*$ et le morphisme, dit « de codage », donné par : $\varphi(\alpha) = a, \varphi(\beta) = ab$ et $\varphi(\gamma) = ba$. Peut-on décoder de manière unique ? Si l'on considère par exemple le mot aba, on peut écrire :

$$\begin{aligned} aba &= \underline{a}\ \underline{ba} = \varphi(\alpha)\varphi(\gamma) = \varphi(\alpha\gamma) \\ &= \underline{ab}\ \underline{b} = \varphi(\beta)\varphi(\alpha) = \varphi(\beta\alpha) \end{aligned}$$

Avait-on $\alpha\gamma$ ou $\beta\alpha$ comme message d'origine ?

1.1 Définitions

Définition 1.1. Soient Σ un alphabet et L un langage de Σ^*. On appelle factorisation d'un mot $w \in \Sigma^*$ en mot de L toute écriture $w = w_1 w_2 \dots w_n$, $w_i \in L$, $1 \leqslant i \leqslant n$.

Définition 1.2. Un langage $X \subseteq \Sigma^*$ est un code si tout mot de Σ^* admet au plus une factorisation en mots de X.

De façon équivalente, $X \subseteq \Sigma^*$ est un code si tout mot de X^+ admet exactement une factorisation en mots de X.

La définition généralement adoptée, et équivalente : un langage $x \subseteq \Sigma^*$ est un code si toute égalité de la forme $x_1 x_2 \dots x_n = x'_1 x'_2 \dots x'_{n'}$, x_i, x'_i tous dans X, entraîne nécessairement $n = n'$ et $x_i = x'_i$, $\forall i \in \{1, \dots, n\}$.

Remarque. Le mot vide ε n'appartient jamais à un code, puisqu'il vérifie l'égalité $\varepsilon = \varepsilon\varepsilon$ et aurait de ce fait deux factorisations distinctes en mots de X.

Exemples.

1. $X = \{a, ab, ba\}$, sur $\Sigma = \{a, b\}$, n'est pas un code car le mot aba peut être factorisé de deux façons en mot de X : $\underline{a}\ \underline{ba} = \underline{ab}\ \underline{a}$.
2. Sur tout alphabet Σ, le langage $X = \Sigma$ est un code. En effet, d'après la propriété universelle du monoïde libre, on sait que tout mot non vide possède une écriture unique comme produit de lettres de l'alphabet.
3. Considérons, sur $\Sigma = \{a, b\}$, le langage $X = \{a, ba, b^2\}$. Il s'agit d'un code. Supposons par l'absurde qu'il n'en est rien. Il existe alors au moins un mot $w \in \Sigma^*$ en deux factorisations distinctes en mots de X :

$$w = x_1 x_2 \dots x_n = x'_1 x'_2 \dots x'_{n'}$$

pour $x_i, x'_i \in X$. Choisissons w de longueur minimale. On peut alors affirmer que $x_1 \neq x'_1$, et plus encore, que $|x_1| \neq |x'_1|$. Supposons alors $|x_1| < |x'_1|$, l'autre cas est symétrique. Nécessairement, x_1 est un préfixe propre de x'_1 : ce qui est impossible car aucun mot de X n'est préfixe propre d'un autre mot de X.

Exercice. Généralisation du deuxième exemple. Pour tout alphabet Σ, montrer que Σ^p est un code, $\forall p > 0$. Ce code est appelé code uniforme des mots de longueur p.

Soit une relation $x_1 \dots x_n = x'_1 \dots x'_{n'}$, $x_i, x'_i \in \Sigma^p$. Supposons, sans perte de généralité, $n \leqslant n'$. Puisque $|x_1| = |x'_1|$, on en déduit $x_1 = x'_1$, d'où $x_2 \dots x_n = x'_2 \dots x'_{n'}$. De proche en proche, on arrive à $x_n = x'_n$ et $\varepsilon = x'_{n+1} \dots x'_{n'}$. Donc finalement, $n = n'$ et $x_i = x'_i$, $\forall i \leqslant n$.

1.2 Codes préfixes, suffixes et bipréfixes

Définition 1.3. Soit L un langage de Σ^*. On dit que L est préfixe (resp. suffixe) lorsqu'aucun mot de L n'est facteur gauche (resp. droit) propre d'un autre mot de L.

L est dit bipréfixe, ou simplement bifixe, s'il est à la fois préfixe et suffixe.

Proposition 1.1. *Tout langage préfixe (suffixe ou bifixe) qui n'est pas réduit au mot vide, ou, ce qui est équivalent, qui ne contient pas le mot vide, est un code.*

Les langages préfixes (resp. suffixes, resp. bifixes) ne contenant pas ε sont appelés codes préfixes (resp. suffixes, resp. bifixes).

Démonstration. Similaire à celle du troisième exemple plus haut. □

Exemples.

1. $X = \{a, ba, b^2\}$, sur $\Sigma = \{a, b\}$, est un code préfixe.
2. Les codes uniformes (de la forme Σ^p) sont des codes bipréfixes.
3. $Y = a^*b$, sur $\Sigma = \{a, b\}$ est un code préfixe infini.
4. $Z = \{a^n b^n / n > 0\}$ est un code bifixe infini et non rationnel.

Exercice. On considère sur l'alphabet $\Sigma = \{a, b\}$ les langages $X = \{a, ba\}$ et $Y = \{a, ab\}$.

1. Montrer que X et Y sont des codes.
2. Qu'en est-il du produit $Z = XY$?

1. X est un langage préfixe non réduit à ε. C'est donc un code préfixe. Y est un langage suffixe non réduit à ε. C'est un code suffixe.
2. $Z = \{a^2, a^2b, ba^2, ba^2b\}$. Ce langage n'est ni préfixe, ni suffixe. De plus, il ne s'agit pas d'un code car, en particulier, le mot a^2ba^2b a deux factorisations sur Z : $\underline{a^2}\ \underline{ba^2b} = \underline{a^2b}\ \underline{a^2b}$.

En conclusion, le produit de deux codes ne constitue pas nécessairement un code.

1.3 Algorithme de Sardinas et Patterson

Définition 1.4. Soient Σ un alphabet, L un langage de Σ^* et u un mot. On note :

$$u^{-1}\mathrm{L} = \{v \in \Sigma^* / uv \in \mathrm{L}\}$$
$$\mathrm{L}u^{-1} = \{v \in \Sigma^* / vu \in \mathrm{L}\}$$

$u^{-1}\mathrm{L}$ et $\mathrm{L}u^{-1}$ sont respectivement appelés quotient à gauche et quotient à droite de L par u. On parle aussi de résiduel à gauche ou à droite.

Cette définition peut s'étendre à toute une partie K de Σ^*, en posant :

$$\mathrm{K}^{-1}\mathrm{L} = \bigcup_{u\in\mathrm{K}} u^{-1}\mathrm{L} = \{v \in \Sigma^* / \exists u \in \mathrm{K}, uv \in \mathrm{L}\} = \{v \in \Sigma^* / \mathrm{K}v \cap \mathrm{L} \neq \emptyset\}$$
$$\mathrm{L}\mathrm{K}^{-1} = \bigcup_{v\in\mathrm{K}} \mathrm{L}v^{-1} = \{u \in \Sigma^* / \exists v \in \mathrm{K}, uv \in \mathrm{L}\} = \{u \in \Sigma^* / u\mathrm{K} \cap \mathrm{L} \neq \emptyset\}$$

Exemples.
- Si $\mathrm{L} = \{\varepsilon\}$, on a $u^{-1}\mathrm{L} = \mathrm{L}$ si $u = \varepsilon$, $u^{-1}\mathrm{L} = \emptyset$ sinon.
- Si $\mathrm{L} = \Sigma^*$, $u^{-1}\mathrm{L} = \Sigma^*$, $\forall u \in \Sigma^*$.
- Si $\mathrm{L} = \Sigma^2$ et $a \in \Sigma$, $a^{-1}\mathrm{L} = \Sigma$. Par exemple, si $\Sigma = \{a, b\}$, on a $\Sigma^2 = \{a^2, ab, ba, b^2\}$. Donc $a^{-1}(\Sigma^2) = \{a, b\} = \Sigma$.

Propriétés. Si a désigne une lettre, u et v des mots et L, L_1 et L_2 des langages :

$$u^{-1}(\mathrm{L}_1 \cup \mathrm{L}_2) = u^{-1}\mathrm{L}_1 \cup u^{-1}\mathrm{L}_2$$
$$u^{-1}(\mathrm{L}_1 \cap \mathrm{L}_2) = u^{-1}\mathrm{L}_1 \cap u^{-1}\mathrm{L}_2$$
$$u^{-1}(\mathrm{L}_1 \setminus \mathrm{L}_2) = u^{-1}\mathrm{L}_1 \setminus u^{-1}\mathrm{L}_2$$
$$a^{-1}(\mathrm{L}_1\mathrm{L}_2) = \begin{cases} (a^{-1}\mathrm{L}_1)\mathrm{L}_2 \text{ si } \varepsilon \notin \mathrm{L}_1 \\ (a^{-1}\mathrm{L}_1)\mathrm{L}_2 \cup a^{-1}\mathrm{L}_2 \text{ sinon} \end{cases}$$
$$a^{-1}\mathrm{L}^* = (a^{-1}\mathrm{L})\mathrm{L}^*$$
$$(uv)^{-1}\mathrm{L} = v^{-1}(u^{-1}\mathrm{L})$$

Dans le cas des résiduels à droite, on dispose de formules symétriques.

Rappel. $\mathrm{L}^* = \bigcup_{n\geqslant 0} \mathrm{L}^n = \{u_1u_2\dots u_n / n \geqslant 0, u_i \in \mathrm{L}, 1 \leqslant i \leqslant n\}$, en posant par convention $\mathrm{L}^0 = \{\varepsilon\}$.

$\mathrm{L}^+ = \bigcup_{n>0} \mathrm{L}^n = \mathrm{L}^*\mathrm{L} = \mathrm{L}\mathrm{L}^*$. On a $\mathrm{L}^* = \mathrm{L}^+ \cup \{\varepsilon\}$, mais en général $\mathrm{L}^+ \neq \mathrm{L}^* \setminus \{\varepsilon\}$.

Nous allons donner sous forme de théorème, un algorithme permettant de tester si un langage $\mathrm{X} \subseteq \Sigma^*$ est un code. Pour cela, on va considérer la suite de langages donnée par :

$$\begin{cases} \mathrm{X}_0 = \mathrm{X}^{-1}\mathrm{X} \setminus \{\varepsilon\} \\ \mathrm{X}_n = \mathrm{X}^{-1}\mathrm{X}_{n-1} \cup \mathrm{X}_{n-1}^{-1}\mathrm{X}, n > 0 \end{cases}$$

Théorème 1.1 (Sardinas-Patterson)**.** X *est un code si et seulement si* $\varepsilon \notin \mathrm{X}_n$, $\forall n \geqslant 0$.

Exemple. Soit, sur $\Sigma = \{a, b\}$, le langage $\mathrm{C} = \{a, aba\}$. S'agit-il d'un code ? C n'est ni un langage préfixe, ni suffixe. On va donc lui appliquer l'algorithme.

$$\mathrm{C}_0 = \mathrm{C}^{-1}\mathrm{C} \setminus \{\varepsilon\} = a^{-1}\mathrm{C} + (aba)^{-1}\mathrm{C} \setminus \{\varepsilon\} = \{\varepsilon, ba\} \setminus \{\varepsilon\} = \{ba\}$$
$$\mathrm{C}_1 = \mathrm{C}^{-1}\mathrm{C}_0 + \mathrm{C}_0^{-1}\mathrm{C} = a^{-1}\{ba\} + (aba)^{-1}\{ba\} + (ba)^{-1}\{a, aba\} = \emptyset$$

Si l'on suppose $C_n = \emptyset$, on obtient :

$$C_{n+1} = C^{-1}C_n + C_n^{-1}C = C^{-1}\emptyset + \emptyset^{-1}C = \emptyset$$

Donc on obtient $C_0 = \{ba\}$, et $C_n = \emptyset$, $\forall n > 0$. D'après le théorème, C est un code.

Exercice. Sur l'alphabet $\Sigma = \{a, b\}$, montrer que $X = \{a^2, ab, a^2b, ab^2, b^2\}$ est un code.

Notons que X n'est ni un langage préfixe, ni suffixe. Calculons la suite :

$$X_0 = X^{-1}X \setminus \{\varepsilon\} = (a^2)^{-1}X + (ab)^{-1}X + (a^2b)^{-1}X + (ab^2)^{-1}X + (b^2)^{-1}X \setminus \{\varepsilon\} = \{b\}$$
$$X_1 = X^{-1}X_0 + X_0^{-1}X = \{a^2, ab, a^2b, ab^2, b^2\}^{-1}\{b\} + (b)^{-1}\{a^2, ab, a^2b, ab^2, b^2\}$$
$$= \emptyset + \{b\} = \{b\}$$

Il est facile de remarquer que l'on a : $X_n = \{b\}$, $\forall n \geqslant 0$. Puisque $\{\varepsilon\} \notin X_n$, $\forall n \geqslant 0$, X est un code en vertu du théorème.

Exercice. On considère, sur l'alphabet $A = \{a, b\}$, le langage $L = \{w \in A^* / |w|_b \text{ pair }\}$.

1. Montrer que L est un sous-monoïde de A^*.
2. Calculer l'automate minimal de L.
3. En déduire une écriture de L sous la forme C^*, C étant un langage de A^* que l'on explicitera.
4. Montrer que C est un code.

1.4 Codes maximaux

Définition 1.5. Un code X sur un alphabet A est dit maximal s'il n'existe pas de partie de A^* contenant strictement X qui soit elle-même un code.

De manière équivalente, si X est un code maximal de A^* si et seulement si $\forall u \in A^+ \setminus X$, $X \cup \{u\}$ n'est pas un code.

On admettra l'énoncé suivant.

Proposition 1.2. *Tout code* X *sur un alphabet* A *est contenu dans un code maximal.*

Exercice. Montrer que, pour tout $p > 0$, le code uniforme A^p des mots de longueur p sur l'alphabet A est maximal.

Pour tout $u \in A^+ \setminus A^p$, posons $X = A^p \cup \{u\}$ et considérons le mot $u^p = \underbrace{uu\ldots u}_{p \text{ fois}}$. Ce mot a pour longueur $p|u|$, avec $p \neq |u|$. On constate alors que l'on a aussi $u^p = x_1x_2\ldots x_{|u|}$, avec $x_i \in A^p \subset X$. Vu que $p \neq |u|$, il est clair que le mot u^p admet deux factorisations (au moins) en mots de X. On en déduit que X ne saurait être un code.

Par exemple, A^2 est un code maximal sur $A = \{a, b\}$. Prenons $u = aba$; on a $u^2 = \underline{aba}\ \underline{aba} = \underline{ab}\ \underline{aa}\ \underline{ba}$.

1.5 Mesure d'un code

Soit Σ un alphabet (fini). On considère l'application $\pi : \Sigma \to \mathbb{Q}$ donnée par $\pi(\sigma) = \frac{1}{|\Sigma|}$. D'après la propriété universelle du monoïde libre, cette application s'étend de façon unique en un morphisme de monoïde, que l'on notera toujours π, du Σ^* dans $(\mathbb{Q}, \times)$. Si $w = a_1 \dots a_n$, $a_i \in \Sigma$, est un mot de Σ^*, on obtient alors :

$$\pi(w) = \pi(a_1 \dots a_n) = \pi(a_1) \dots \pi(a_n) = \left(\frac{1}{|\Sigma|}\right)^n = |\Sigma|^{-|w|}$$

Ce morphisme s'étend, encore une fois, en une simple application de $\mathscr{P}(\Sigma^*)$ dans $\mathbb{Q} \cup \{\infty\}$, en posant pour tout langage L : $\pi(\mathrm{L}) = \sum_{w \in \mathrm{L}} \pi(w)$. Ce n'est plus un morphisme : si $\mathrm{L} = \{a, ba\}$ et $\mathrm{K} = \{a, ab\}$ sur $\Sigma = \{a, b\}$, on a :

$$\pi(\mathrm{L} + \mathrm{K}) = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 1 \neq \frac{3}{2} = \frac{3}{4} + \frac{3}{4} = \pi(\mathrm{L}) + \pi(\mathrm{K})$$

Exercice. Soit $p \geqslant 0$, calculer $\pi(\mathrm{A}^p)$, $\forall \mathrm{A}$.

Remarque. La mesure d'un langage peut être infinie. Par exemple, pour Σ^*, on obtient :

$$\pi(\Sigma^*) = \Sigma_{w \in \Sigma^*} \pi(w) = \Sigma_{p \in \mathbb{N}} \pi(\Sigma^p) = \Sigma_{p \geqslant 0} 1 = +\infty$$

Exercice. L'application π définie ci-dessus n'est pas non plus compatible avec le produit de concaténation des langages. Trouver deux langages L et K tels que $\pi(\mathrm{LK}) \neq \pi(\mathrm{L})\pi(\mathrm{K})$.

Exemples. Calculons la mesure des langages suivants, sur $\mathrm{A} = \{a, b\}$:

- $\mathrm{L}_1 = \{a, ab, ba, b^2\}$, $\pi(\mathrm{L}_1) = \frac{5}{4}$, L_1 n'est pas un code.
- $\mathrm{L}_2 = \{a, a^2\}$, $\pi(\mathrm{L}_2) = \frac{3}{4}$, L_2 n'est pas un code.
- $\mathrm{L}_3 = \{a, aba\}$, $\pi(\mathrm{L}_3) = \frac{5}{8}$, L_3 est un code.
- $\mathrm{L}_4 = \{a, ab, b^2\}$, $\pi(\mathrm{L}_4) = 1$, L_4 est un code.
- $\mathrm{L}_5 = \mathrm{A}^p, p \geqslant 0$, $\pi(\mathrm{L}_5) = 1$, L_5 est un code pour $p > 0$.
- $\mathrm{L}_6 = \{a, ab, ba\}$, $\pi(\mathrm{L}_6) = 1$, L_6 n'est pas un code.

Théorème 1.2 (Kraft, McMillan)**.** *Soit* X *un code. Alors* $\pi(\mathrm{X}) \leqslant 1$.

La preuve de ce théorreme va requérir le lemme suivant.

Lemme 1.1. *Soit X un code fini. Alors* $\pi(\mathrm{X}^k) = \pi(\mathrm{X})^k$, $\forall k > 0$.

Démonstration. On a

$$\begin{aligned} \mathrm{X}^k &= \{u_1 u_2 \dots u_k / u_1, u_2, \dots, u_k \in \mathrm{X}\} \\ &= \bigcup_{u_1, u_2, \dots, u_k \in \mathrm{X}} \{u_1 u_2 \dots u_k\} \end{aligned}$$

Cette réunion est disjointe, puisque X est un code. D'où

$$\begin{aligned}
\pi(\mathrm{X}^k) &= \sum_{w\in\mathrm{X}^k}\pi(w) = \sum_{u_1,u_2,\ldots,u_k\in\mathrm{X}}\pi(u_1u_2\ldots u_k)\\
&= \sum_{u_1,u_2,\ldots,u_k\in\mathrm{X}}\pi(u_1)\pi(u_2)\ldots\pi(u_k)\\
&= \sum_{u_1\in\mathrm{X}}\sum_{u_2\in\mathrm{X}}\ldots\sum_{u_k\in\mathrm{X}}\pi(u_1)\pi(u_2)\ldots\pi(u_k)\\
&= \left[\sum_{u_1\in\mathrm{X}}\pi(u_1)\right]\times\left[\sum_{u_2\in\mathrm{X}}\ldots\sum_{u_k\in\mathrm{X}}\pi(u_2)\ldots\pi(u_k)\right]\\
&= \left[\sum_{u_1\in\mathrm{X}}\pi(u_1)\right]\times\cdots\times\left[\sum_{u_k\in\mathrm{X}}\pi(u_k)\right]\\
&= \underbrace{\pi(\mathrm{X})\times\cdots\times\pi(\mathrm{X})}_{k\text{ fois}}\\
&= \pi(\mathrm{X})^k
\end{aligned}$$

□

Démontrons maintenant le théorème.

Démonstration. Pour effectuer la démonstration, on va se restreindre au cas d'un code fini, tout en sachant que l'énoncé demeure valide quel que soit le code.

Soit X un code fini. Posons :

$$s = \inf\{|w|, w\in\mathrm{X}\},\quad t = \sup\{|w|, w\in\mathrm{X}\}$$

Ainsi, on a : $\mathrm{X}\subseteq\mathrm{A}^s+\mathrm{A}^{s+1}+\cdots+\mathrm{A}^t$. Ce qui donne $\mathrm{X}^k\subseteq\mathrm{A}^{ks}+\mathrm{A}^{ks+1}+\cdots+\mathrm{A}^{kt}$. Et alors,

$$\begin{aligned}
\pi(\mathrm{X}^k) &\leqslant \pi(\mathrm{A}^{ks}+\cdots+\mathrm{A}^{kt})\\
\pi(\mathrm{X})^k = \pi(\mathrm{X}^k) &\leqslant \sum_{p=ks}^{kt}\pi(\mathrm{A}^p)\\
&\leqslant \sum_{p=ks}^{kt}1 = kt-ks+1 = k(t-s)+1
\end{aligned}$$

Si l'on pose $\alpha=\pi(\mathrm{X})$ et $\beta=t-s$, on obtient :

$$\alpha^k\leqslant k\beta+1\quad\forall k$$

On doit donc avoir nécessairement $\pi(\mathrm{X})\leqslant 1$. □

Corollaire 1.1. *Si* X *est un code vérifiant* $\pi(\mathrm{X})=1$*, alors* X *est maximal.*

Démonstration. Soit X un code vérifiant $\pi(\mathrm{X})=1$ et soit $\mathrm{Y}=\mathrm{X}\cup\{u\}$, $u\in\mathrm{A}^+\setminus\mathrm{X}$. On a $\pi(u)>0$, donc $\pi(\mathrm{Y})=\pi(\mathrm{X})+\pi(u)>1$. Donc, pour tout mot $u\notin\mathrm{X}$, Y ne saurait être un code. X est donc maximal. □

Exercice. On considère sur l'alphabet $\mathrm{A}=\{a,b\}$ le langage $\mathrm{L}=\{w\in\mathrm{A}^*/|w|_a \text{ impair}\}$.

1. Calculer l'automate minimal de L.
2. En déduire une écriture de L sous la forme $(Xa + b)^*X$, X étant un langage de A^* que l'on explicitera.
3. Montrer que X est un code maximal, et que le langage $Y = Xa + b$ est aussi un code maximal.
4. Donner une description du sous-monoïde Y^* de A^* engendré par Y.