Advanced information, computation, communication II

COM - 108
Résumé de cours

Auteur :
François Dumoncel

Superviseur :
M.Rimoldi

Résumé du cours Advanced information, computation, communication II

Février 2020

Sommaire

1 Avertissement

Ce document présente les notions fondamentales du cours *Advanced Information, Computation, Communication II*. Il est principalement inspiré du livre *Introduction aux sciences de l'information* sur lequel le cours du prof. Rimoldi se base. **Cependant**, ce document ne remplace en **aucun cas** le cours dispensé par Bixio et ses assistants, entre autre, il y a des passages du cours que vous ne retrouverez que dans les slides. Il est donc conseillé d'utiliser ce document en tant qu'aide mémoire, de pense bête et/ou de support de cours.

Si une erreur s'est glissée, aussi petite soit elle :
francois.dumoncel-kessler@epfl.ch

Enjoy !

Dernière mise à jour : 11.06.2021

2 Rappel de probabilité

Nous appelons source la donnée d'un ensemble fini appelé alphabet $\mathcal{A}$ et d'une densité de probabilité. Les éléments de $\mathcal{A}$ sont appelés les symboles émis par la source $\mathcal{A}$.

Les symboles d'une source sont équiprobables si :

$$\forall s \in \Omega, \quad p(s) = \frac{1}{M}, \quad \text{avec } M = card(\mathcal{A})$$

Soient B et C deux sous-ensembles de $\mathcal{A}$ avec $P(C) > 0$. La probabilité conditionnelle de B sachant C est

$$P(B|C) := \frac{P(B \cap C)}{P(C)}$$

On dit qu'une source S est une source composée si son alphabet est de la forme $\mathcal{A} = \mathcal{A}_1 \times ...\mathcal{A}_n$. Cela modélise une suite de n observations. A partir d'une source composée on peut dériver n sources marginales, obtenues en considérant chaque composante individuellement, ce qu'on écrit

$$S = (S_1, S_2, ..., S_n)$$

On dit que les sources marginales $S_1, ..., S_n$ sont des sources indépendantes si

$$\forall (s_1, s_2, ..., s_n) \in \mathcal{A}, \quad p(s_1, s_2, ..., s_n) = p_{S_1}(s_1)p_{S_2}(s_2)...p_{S_n}(s_n) = \prod_{i=1}^{n} p_{S_i}(s_i)$$

Si S est une source composée à deux composantes S_1, S_2, alors la densité conditionnelle de la source marginale S_2 sachant que $S_2 = s_2$ est définie pour tout $s_2 \in \mathcal{A}_2$ tel que $p_{S_2}(s_2) > 0$ par

$$p_{S_1|S_2}(s_1|s_2) := \frac{p_{S_1,S_2}(s_1, s_2)}{p_{S_2}(s_2)}, \quad \forall (s_1, s_2) \in \mathcal{A}$$

La notion d'indépendance signifie que l'observation d'une source marginale ne donne aucune information sur l'autre, pour un observateur qui connaît la probabilité de la source S.

Théorème. Soit $S = (S_1, S_2)$ une source composée. Les propriétés suivantes sont équivalentes :

1. les deux sources marginales S_1 et S_2 sont indépendantes, c-à-d
$$p(s_1, s_2) = p_{S_1}(s_1)p_{S_2}(s_2),\ \forall (s_1, s_2) \in \mathcal{A}$$
2. la densité conditionnelle de S_2 sachant $S_1 = s_1$ vaut
$$p_{S_1|S_2}(s_1|s_2) = p_{S_2}(s_2),\ \forall (s_1, s_2) \in \mathcal{A}$$
3. la densité conditionnelle de S_2 sachant que $S_1 = s_1$ ne dépend pas de la valeur de s_1.

3 Entropie et codage de source

3.1 Entropie et information

Considérons une source discrète d'information S qui délivre un message $s \in \mathcal{A}$ avec probabilité $p(s)$. En recevant un tel message s, si $p(s) = 1$, il n'y a aucune surprise à recevoir le symbole s, celui-ci n'apporte aucune information. Par contre si $p(s) = 0.0001$, la "surprise" de recevoir s parmi les M symboles de la source est beaucoup plus grande. L'information réside donc dans l'effet de surprise qu'elle engendre. Un événement probable ne donnera pas beaucoup d'information tandis qu'un événement qui a peut de chance de se produire en apporte beaucoup.

Soit $(\mathcal{A}, p)$ une source et E un sous-ensemble de $\mathcal{A}$. L'information de l'événement E est
$$I(E) := -\log_2(P(E))$$

Malheureusement, la notion d'information ne *suffit pas*. C'est pourquoi on utilise l'information *moyenne*, appelée entropie

Soit $S = (\mathcal{A}, p)$ une source. L'entropie de S est
$$H(S) := -\sum_{s \in \mathcal{A}} p(s) \log_2(p(s))$$

Par convention si $p(s) = 0$ pour un certain s alors $0 \log_2(0) = 0$.

Entropie d'une source binaire. Une source qui émet uniquement deux symboles

est appelé source binaire est son entropie vaut

$$h(q) := -q \log_2(q) - (1-q) \log_2(1-q)$$

Théorème.

1. $H(S) \geq 0$
2. Si pour un certain $s \in \mathcal{A}, p(s) = 1$ alors $H(S) = 0$
3. Réciproquement, si $H(S) = 0$, alors il existe un $s \in \mathcal{A}$ tel que $p(s) = 1$.

Théorème. *(Inégalité de Jensen)* Le logarithme d'une moyenne est supérieur ou égal à la moyenne des logarithmes.

Théorème. Soit S une source avec un alphabet de M symboles.

1. $H(S) \leq \log_2(M)$
2. Si les M symboles de la sources sont équiprobables, alors $H(S) = \log_2(M)$
3. Si $H(S) = \log_2(M)$ alors les M symboles de la source sont équiprobables.

Pour une source composée, la somme des entropies de ses marginales est plus grande que son entropie propre. Voici le théorème.

Théorème. Soit $S = (S_1, ..., S_n)$ une source composée.

1. $H(S_1, ..., S_n) \leq H(S_1) + ... + H(S_n)$
2. $H(S_1, ..., S_n) = H(S_1) + ... + H(S_n)$ si est seulement si les sources marginales $S_1, ..., S_n$ sont indépendantes.

3.2 Codage de source

On appelle *codage de source* l'opération qui traduit les symboles d'une source en des symboles utilisables par une machine (par exemple en *bits*). Une raison d'utiliser le codage de source est une raison d'efficacité : nous voulons *comprimer* autant que possible la source et prendre le moins de place possible, ceci sans aucune altération (*compression sans perte*).

Soit une source S d'alphabet $\mathcal{A} = \{s_1, ..., s_M\}$. Nous avons maintenant un deuxième alphabet, l'alphabet de code $\mathcal{D}$, qui est un ensemble de D symboles de code. Le plus souvent $D = 2$ et alors $\mathcal{D} = \{0, 1\}$. Les éléments de D sont les symboles de code.

Un dictionnaire $\mathcal{C}$ est un sous-ensemble fini de suites finies d'éléments de $\mathcal{D}$. Un élément de $\mathcal{C}$ est appelé mot de code.

Définition. *(Code de source)* Un code de source, ou encodage, est une application bijective $\Gamma : \mathcal{A} \to \mathcal{C}$.

Le code Γ permet donc de traduire tout symbole de la source en un mot de code, de façon que pour chaque symbole il existe un mot de code, et inversement (puisque Γ est bijective). Pour que Γ puisse être bijective il faut que le dictionnaire $\mathcal{C}$ comporte exactement S mots de code, comme l'alphabet de la source.

La longueur d'un mot $c \in \mathcal{C}$ est $l(c) =$ le nombre de symboles de code de c. On dit qu'un code est à longueur constante si tous les mots de code sont de mêmes longueurs. Sinon on dit que c'est une code à longueur variable.

Puisqu'un code est une application bijective, elle peut être inversée. L'application inverse $\Gamma^{-1} : \mathcal{C} \to \mathcal{A}$, qui consiste à traduire *un* mot de code en *un* symbole de S, est appelée décodage.

Définition. Soit Γ un code de source. Γ est à décodage unique si pour toute suite de symboles de code, qui résulte de l'encodage d'une suite de symboles de S, il existe un décodage unique.
Parmi tous les codes possibles, il en existe de plus faciles à manipuler, ce sont les codes instantanés.
Un code est sans préfixe si aucune mot de code n'est préfixe d'un autre mot de code.

Définition. Nous disons qu'un code est instantané

1. s'il est à décodage unique
2. et si il est sans préfixe.

Théorème. *Un code est sans préfixe si et seulement si il est instantané.*

Théorème. (*Kraft-McMillan*) Soit Γ un D-aire code dont les longeurs des M mots de code sont $l_1, ..., l_m$. Si Γ est à décodage unique alors il satisfait **l'inégalité de Kraft** :

$$D^{-l_1} + ... + D^{-l_M} \leq 1$$

Réciproquement si l'inégalité de Kraft est vérifié alors il existe un D-aire code à décodage unique dont le dictionnaire possède M mots de code et dont les longueurs des mots de codes sont $l_1, ..., l_M$

Attention : Un code peut vérifier l'inégalité de Kraft sans pour autant être à décodage unique. En revanche un code à décodage unique doit impérativement valider l'inégalité de Kraft.

Enfin, une conséquence spectaculaire du théorème de Kraft-McMillan est qu'on peut toujours remplacer un code à décodage unique par un code instantané :

Théorème. Pour tout code à décodage unique, il existe un code instantané sur les mêmes alphabets de source et de code, qui a les mêmes longueurs de mots.

Exemples importants : Soit Γ un encodage.
Vrai

- Γ est instantané $\Rightarrow$ Γ vérifie l'inégalité de Kraft.
- Γ est à décodage unique $\Rightarrow$ Γ vérifie l'inégalité de Kraft.
- Γ ne vérifie pas l'inégalité de Kraft $\Rightarrow$ Γ n'est pas instantané.
- Γ ne vérifie pas l'inégalité de Kraft $\Rightarrow$ Γ est avec préfixe.
- On peut toujours remplacer un code à décodage unique par un code sans préfixe qui à les mêmes longueurs de mots.

Faux

- Γ est avec préfixe $\Rightarrow$ Γ n'est pas à décodage unique.
- Γ vérifie l'inégalité de Kraft $\Rightarrow$ Γ est à décodage unique.
- Γ vérifie l'inégalité de Kraft $\Rightarrow$ Γ est sans préfixe.
- Γ est avec préfixe $\Rightarrow$ Γ vérifie l'inégalité de Kraft.
- Γ n'est pas instantané $\Rightarrow$ Γ ne vérifie pas l'inégalité de Kraft.

3.3 Efficacité d'un code de source

La quantité d'intérêt pour l'efficacité d'un code est sa longueur moyenne, définie comme le nombre moyen de symboles de code par symbole de source.

Définition. Soit une source S d'alphabet $\mathcal{A}$ et de densité de probabilité p, et soit Γ un D-aire code de la source S. La longueur moyenne du code Γ est

$$L(\Gamma) := \sum_{s \in \mathcal{A}} p(s) l(\Gamma(s))$$

Théorème. (*Première inégalité de l'entropie*) Soit une source S d'entrope $H(S)$ et soit Γ un D-aire code de la source S. Si Γ est à décodage unique, sa longueur moyenne satisfait

$$L(\Gamma) \geq \frac{H(S)}{\log_2(D)}$$

Donc pour un code Γ binaire on a $L(\Gamma) \geq H(S)$

Pour obtenir un code efficace, une idée est de donner des longueurs de code petites aux mots de codes les plus fréquents. Ainsi les codes de **Shannon-Fano** choisissent un D-aire code avec $\lceil \log_D(1/p(s)) \rceil$ comme longueur de mot de code pour le symbole s.

Théorème. Soit une source S avec M symboles dont les densités de probabilité sont $p_1, ..., p_M$. Il existe des D-aires codes instantanés (donc à décodage unique) dont les longueurs des mots sont $l_i = \lceil \log_D(1/p_i) \rceil$ pour $i = 1, ..., M$. On appelle de tels codes des D-aires codes de **Shannon-Fano.**

Les codes de **Shannon-Fano** sont assez efficaces, sans être en général, les plus efficaces. Cependant, et c'est leur principal atout, ils sont *garantis* : ils ne peuvent pas être à plus d'une unité de la borne inférieure de l'entropie.

Théorème. (*Deuxième inégalité de l'entropie*) La longueur moyenne $L(\Gamma_{SF})$ d'un D-aire code de Shannon-Fano d'une source d'entropie $H(S)$ vérifie

$$\frac{H(S)}{\log_2(D)} \leq L(\Gamma_{SF}) \leq \frac{H(S)}{\log_2(D)} + 1$$

Ce qui donne dans le cas d'un code binaire

$$H(S) \leq L(\Gamma_{SF}) \leq H(S) + 1$$

Nous avons obtenue une borne inférieure sur la longueur de tout code et nous avons vu que les codes de SF sont à au plus une unité de cette borne. Mais existe-il de meilleurs codes ? En général la réponse est non. Par contre, nous allons maintenant voir qu'il est possible de créer des codes optimaux, c-à-d de longueur minimal parmi tous les codes à décodage unique : ce sont les code de Huffman. La procédure pour la construction d'un *code de Huffman* ne sera pas décrite ici mais est supposée connu par le lecteur.

Théorème. (*Code de Huffman*) Le méthode pour créer un code de Huffman produit un code binaire instantané Γ_H optimal, c-à-d que pour tout autre code binaire à décodage unique Γ pour la même source, on a

$$L(\Gamma_H) \leq L(\Gamma)$$

Donc finalement pour les codes binaires

$$H(S) \leq L(\Gamma_H) \leq L(\Gamma_{SF}) \leq H(S) + 1$$

On dit que le code de Huffman domine le code de Shannon-Fano.

3.4 Entropie conditionnelle

Définition. Soit $S = (S_1, S_2)$ une source composée. L'entropie conditionnelle de S_2 sachant que $S_1 = s_1$ est l'entropie de la densité conditionnelle de S_2 sachant que $S_1 = s_1$ ce qui s'écrit

$$H(S_2|S_1 = s_1) := -\sum_{s_2 \in \mathcal{A}_2} p_{S_2|S_1}(s_2|s_1) \log_2(p_{S_2|S_1}(s_2|s_1))$$

L'entropie conditionnelle de S_2 sachant S_1 en est la moyenne :

$$H(S_2|S_1) := \sum_{s_1 \in \mathcal{A}_1} H(S_2|S_1 = s_1) p_{S_1}(s_1)$$

L'entropie conditionnelle mesure la quantité d'information moyenne que l'on reçoit quand on observe une source, après avoir observé l'autre, ce qu'on peut aussi appeler information "supplémentaire".

Il est souvent plus facile de calculer l'entropie conditionnelle en utilisant le théorème suivant, qui exprime que l'information que nous délivre la source composée est la somme de l'information délivrée par une composante plus l'information supplémentaire délivrée par l'autre composante.

Théorème. *(Calcul de l'entropie conditionnelle)* Soit $S = (S_1, S_2)$ une source composée

$$\begin{aligned} H(S_1, S_2) &= H(S_1) + H(S_2|S_1) \\ &= H(S_2) + H(S_1|S_2) \end{aligned}$$

Et comme l'entropie conditionnelle est positive ou nulle il suit que

$$H(S_1) \leq H(S_1, S_2)$$

Théorème. *(Conditionner réduit l'entropie)* Soit $S = (S_1, S_2)$ une source composée

1. $H(S_2|S_1) \leq H(S_2)$
2. $H(S_2|S_1) = H(S_2)$ si et seulement si S_1 et S_2 sont indépendantes.

(Conditionner réduit l'entropie, suite) Soit $S = (S_1, S_2, S_3)$ une source composée

$$H(S_3|S_1, S_2) \leq H(S_3|S_2)$$

Note : Dans une notation telle que $S = (S_1, S_2, S_3)$, on considère qu'il y a une source composée $S = ((S_1, S_2), S_3)$ dont la première composante est elle-même une source composée.
(*Calcul incrémental de l'entropie conditionnelle*) Soit $S = (S_1, S_2, ..., S_n)$ une source composée de n composantes

$$H(S_1, S_2, ..., S_n) = H(S_1) + H(S_2|S_1) + ... + H(S_n|S_1, ..., S_{n-1})$$

Le dernier théorème est facile à retenir si l'on interprète l'entropie conditionnelle comme l'information supplémentaire : l'information totale délivrée par la source S est l'information délivrée par S_1, plus l'information supplémentaire délivrée par S_2, plus etc., plus l'information délivrée par S_n

Définition. Soit $S = (S_1, S_2)$ une source composée. On dit que S_2 se déduit de manière déterministe de S_1 ou encore que S_2 est fonction de S_1 si pour tout $s_1 \in S_1$ tel que $p_{S_1}(s_1) > 0$ il existe un unique $s_2 \in S_2$ tel que $p_{S_2|S_1}(s_2|s_1) = 1$.

Théorème. S_2 est fonction de S_1 si et seulement si $H(S_2|S_1) = 0$. De ce théorème découle le suivant : *(Traitement de l'information)*. Si S_2 est fonction de S_1 alors

$$H(S_1, S_2) = H(S_1)$$
$$H(S_2) \leq H(S_1)$$

Intuitivement, l'inégalité du traitement de l'information explique ce qui se passe quand on applique un algorithme à un message S_1 pour produire un message S_2.S_2 n'apporte aucune information (si on connaît l'algorithme de S_1) donc l'entropie de $S = (S_1, S_2)$ est égale à l'entropie de S_1. De plus, S_2 ne peut pas contenir plus d'information que celle déjà présente dans S_1, donc son entropie est au mieux égale à celle de S_1.

Théorème. Soit $S = (S_1, S_2)$ une source composée telle que S_2 se déduit de manière déterministe de S_1, et vice-versa, S_1 se déduit de manière déterministe de S_2. Alors $H(S_1) = H(S_2) = H(S_1, S_2)$

3.5 Théorème du codage de source

Nous allons maintenant considérer un modèle plus complexe de source, celui de "source étendue", qui modélise mieux la production d'un texte en français par un humain. Pour une source étendue, le concept-clé est celui d'"entropie par symbole", définie à partir de l'entropie conditionnelle.

Définition. Une source étendue $\mathcal{S}$ sur un alphabet $\mathcal{A}$ est la donnée d'une famille de source S^n définies pour tout $n = 1, 2, 3...$ telles que

1. S^n est une source à n composantes, sur l'alphabet $\mathcal{A} \times ... \times \mathcal{A}$; notons p_{S^n} sa densité de probabilité.
2. la densité de probabilité de la source constituée des n premières sources marginales de S^{n+k} est égale à p_{S^n} pour tout $k \geq 1$ et $n \geq 1$.

On dit que la source étendue $\mathcal{S}$ est stationnaire si, pour tout n fixé, la densité de probabilité de $(S_{k+1}, ..., S_n)$ est la même pour toutes valeurs de $k \geq 0$. Intuitivement, le source ne change pas son comportement moyen quand le temps passe, ne rajeunit pas et ne vieillit pas.

Définition. La source étendue $\mathcal{S}$ est dite régulière si les limites

$$H(\mathcal{S}) := \lim_{n\to\infty} H(S_n)$$
$$H^*(\mathcal{S}) := \lim_{n\to\infty} H(S_n|S_1, S_2, ..., S_{n-1})$$

existent.
Pour une source étendue régulière $\mathcal{S}$, H(S) est appelée l'entropie d'un symbole et $H^*(\mathcal{S})$ est appelée l'entropie par symbole

A savoir : *Toute source stationnaire est régulière.*

Théorème. Soit $\mathcal{S}$ une source étendue régulière. Alors

1. $H^*(\mathcal{S}) \leq H(\mathcal{S})$
2. si les sources marginales sont indépendantes alors il y a égalité.

L'idée principal du théorème du codage de source est : le codage par bloc. Au lieu de coder un symbole de la source S_1, on code un bloc de n symboles, c-à-d que nous codons la source $S^n = (S_1, ..., S_n)$. Le premier résultat remarquable concerne l'entropie de S^n et dit que, pour n grand, $H(S_n)$ est à peu près égale à l'entropie par symbole.

Théorème. Soit $\mathcal{S}$ une source étendue régulière. Alors

$$\lim_{n\to\infty} \frac{H(S_1, ..., S_n)}{n} = H^*(\mathcal{S})$$

où $H^*(\mathcal{S})$ est l'entropie par symbole et $H(S_1, ..., S_n)$ l'entropie d'un bloc de n symboles.

Théorème. (*Premier théorème de Shannon*) Soit $\mathcal{S}$ une source étendue régulière et $H^*(\mathcal{S}$ son entropie par symbole. Soient L^n_{SF}, respectivement L^n_H, les longueurs moyennes des D-aires codes de Shannon-Fano, respectivement Huffman, pour un bloc de n symboles de la source. Alors

$$\lim_{n\to\infty} \frac{L^n_H}{n} = \lim_{n\to\infty} \frac{L^n_{SF}}{n} = \frac{H^*(\mathcal{S})}{\log_2(D)}$$

A savoir : *Dire qu'une source étendue régulière $\mathcal{T}$ est binaire et incompressible ("optimal") signifie que $H(\mathcal{T}) = 1$.*

4 Cryptographie et Arithmétique

4.1 La cryptographie

Le besoin de protection de l'information doit respecter les propriétés suivantes :

- l'intégrité : le message reçu est identique à celui qui à été envoyé,
- la confidentialité : seul le destinataire autorisé est capable de lire le message,
- l'authentification : le destinataire peut être certain que le message a vraiment été écrit par la personne qui prétend en être l'auteur.

Lors d'une opération de chiffrement, le texte clair P est transformé par une fonction E paramétrée par une clé K, pour ainsi obtenir un texte chiffré $C = E_K(P)$. Ce texte chiffré est alors transmis au récepteur, qui applique un algorithme de déchiffrement D_k muni d'une clé k, qui recouvre le texte clair original : $P = D_k(C) = D_k(E_K(P))$. Si la clé de chiffrement et de déchiffrement sont égales on parle de

cryptographie symétrique, sinon on parle de cryptographie asymétrique. Si le système est symétrique il est très important que la clé soit maintenue secrète.

On distingue trois types d'attaques :

1. Ciphertext-only attack *(attaque à texte chiffré seul)* : Le cryptanalyste a sa disposition plusieurs textes chiffrés mais pas de texte en clair.
2. Known plain attack (*attaque à texte clair connu)* : Le cryptanalyste possède maintenant les paires de code clairs et chiffrés et sait qu'ils ont été encryptés avec la même clé.
3. Chosen plaintext attack (*attaque à texte clair choisi*) : Le cryptanalyste peut donner n'importe quelle texte à encoder avec la même clé.

Il existe plusieurs types de chiffrement, plus ou moins anciens. Les plus connus sont :

- Le chiffrement par substitution monoalphabétique qui consiste à remplacer chaque lettre de l'alphabet du texte clair par une autre lettre, sans respecter de relation de rotation entre les deux alphabets (contrairement au chiffre de César). Le clé est tout simplement le tableau de correspondance entre les lettres de l'alphabet du texte clair et celui du texte crypté. Malheureusement ce chiffrement peut être cassé par analyse fréquentielle des lettres.
- Le chiffrement de Vigenère qui est un exemple de chiffrement à substitution polyalphabétique. La clé est en général un mot court et facile à mémoriser comme par exemple *BONJOUR*. La clé est répétée sous le texte en clair et on utilise une table de Vigenère pour encoder et décoder le message. Malheureusement ce code peut être casser en devinant la longueur de la clé.

alphabet d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
rangée A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
rangée B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
rangée C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
rangée D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
rangée E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
rangée F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
rangée G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
rangée H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
rangée I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
rangée J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
rangée K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
rangée L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
rangée M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
rangée N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
rangée O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
rangée P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
rangée Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
rangée R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
rangée S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
rangée T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
rangée U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
rangée V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
rangée W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
rangée X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
rangée Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
rangée Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

texte clair :	l	a	c	r	y	p	t	o	g	r	a	p	h	i	e
clé :	B	O	N	J	O	U	R	B	O	N	J	O	U	R	B
texte crypté :	M	O	P	A	M	J	K	P	U	E	J	D	B	Z	F

Figure 1 – Table de Vigenère

Définition. On dit que le cryptosystème est à confidentialité parfaite si les sources qui délivrent les messages P et C sont indépendantes.

Le chiffrement de Vernam ou *one time pad* en anglais, est un système utilisé dans les ambassades. Il fonctionne de la façon suivante :

- Le texte clair P est encodé comme une suite de n bits. La longueur n est constante et connu de tous.
- La clé K est aussi une suite de n bits, tirée au sort, indépendamment de P, et utilisée une et une seule fois. (Nous supposons que les choix de clé sont équiprobables).
- Le texte chiffré est alors
$$C = P \oplus K$$
et le déchiffrement se fait par
$$P = C \oplus K$$

De plus, nous pouvons en déduire la densité de probabilité conditionnelle du texte chiffré sachant que le texte clair est P :

$$p_{\mathcal{C}|\mathcal{P}}(C|P) = \frac{1}{2^n}$$

Théorème. *(Cryptographie symétrique)* Considérons un sytème de cryptographie symétrique (donc à clé secrète). Supposons de plus que la clé soit choisie indépendamment du texte clair. Alors
$$H(\mathcal{P}) \leq H(\mathcal{C})$$
où $H(\mathcal{P})$ est l'entropie du texte clair et $H(\mathcal{C})$ l'entropie du texte crypté.

Théorème. (*Confidentialité parfaite*) Considérons un système de cryptographie symétrique à confidentialité parfaite. Supposons que la clé soit choisie indépendamment du texte clair. Alors
$$H(\mathcal{P}) \leq H(\mathcal{C}) \leq H(\mathcal{K})$$
où $H(\mathcal{K})$ est l'entropie de la clé.

4.2 Arithmétique

Théorème. (*Théorème fondamental de l'arithmétique*). Pour tout a strictement positif, il existe une suite unique de nombres premiers $p_1 < ... < p_k$ et une suite d'exposant $\alpha_1 > 0, ..., \alpha_k > 0$ tels que

$$a = p_1^{\alpha_1} ... p_k^{\alpha_k}$$

Les nombres $p_1, ..., p_k$ sont appelés les **facteurs premiers** de a.

Intuitivement on dira alors que a divise b si et seulement les facteurs premiers de a apparaissent dans la décomposition de b, avec un exposant supérieur ou égal.

Théorème.

1. Deux nombres premiers distincts sont premiers entre eux,
2. Soit p un nombre premier et a un entier tel que $1 \leq a \leq p-1$. Alors a et p sont premiers entre eux.
3. Soient a et b deux nombres premiers entre eux, et c un entier. Si a et b divisent c alors ab divise c.

Définition. Soient a et b deux entiers et m un entier non nul. On dit alors que a est congru à b modulo m, et on écrit :

$$a \equiv b \pmod{m}$$

si b est le reste dans la division euclidienne de a par m. On a alors que

$$a \equiv 0 \pmod{m}$$

si m divise a.

Attention : Les opérations de congruence se combinent bien entre elles (car la congruence est une relation d'équivalence) mais il ne faut jamais diviser une relation de congruence !

A savoir : Tout nombre entier est congru modulo 9 à la somme de ses chiffres décimaux. (Preuve triviale)

Exemple : $298242 \equiv 2+9+8+2+4+2 \equiv 27 \equiv 0 \pmod{9}$ Donc 298242 est divisible par 9.

Exemple. *(MOD-97 et IBAN)*. Supposons que nous voulons transmettre une numéro de n chiffres décimaux. Pour détecter des erreurs simples, comme l'oubli d'un chiffre ou une interversion, nous pouvons y ajouter les deux chiffres de contrôle modulo 97, définis comme le reste dans la division par 97 du nombre original à n chiffres. Par exemple, les deux chiffres de contrôle pour le numéro $021\ 235\ 1234$ sont 95 car $212\ 351\ 1234 \equiv 95 \pmod{97}$.
Supposons maintenant que nous avons intervertis deux chiffres on a alors : $021\ 253\ 1234$ - 95. Le destinataire peut détecter l'erreur en recalculant les chiffres de contrôle, en effet $212\ 531\ 234 \equiv 63 \pmod{97}$ donc les deux chiffres de contrôle devraient être 63, et non pas 95. La procédure appelée MOD 97-10 est basée sur ce principe, avec les modifications suivantes :

1. *Ajouter 00 à la fin du numéro.*
2. *Calculer le reste r dans la division par 97 du numéro ainsi obtenu.*
3. *Les deux chiffres de contrôle MOD 97-10 sont les deux chiffres du complément à 98 de r. Remplacer le 00 final par ces deux chiffres. ($98 - r$)*
4. *Pour vérifier la validité d'un numéro, vérifier que le reste dans la division par 97 est égal à 1.*

4.3 Arithmétique modulaire

Définition. Soit $m \geq 2$ un entier fixé. Pour tout entier a, on appelle classe de congruence de a modulo m, et on note $[a]_m$ l'ensemble des entiers a' tels que $a \equiv a' \pmod{m}$

Théorème. Il y a exactement m classes de congruence modulo m, ce sont $[0]_m, [1]_m, ..., [m-1]_m$

Définition. On note $\mathbb{Z}/m\mathbb{Z}$ l'ensemble des classes de congruence modulo m.
A partir de maintenant, nous pouvons voir $\mathbb{Z}/m\mathbb{Z}$ comme un ensemble à m éléments.

$$\mathbb{Z}/m\mathbb{Z} = \{[0]_m, [1]_m, ..., [m-1]_m\}$$

Note : $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{F}_m$.

Définition. On définit la somme et le produit dans $\mathbb{Z}/m\mathbb{Z}$ par

$$[a]_m + [b]_m = [a+b]_m$$
$$[a]_m [b]_m = [ab]_m$$

L'addition a les propriétés suivantes :

- *Associativité* : $[a]_m + ([b]_m + [c]_m) = ([a]_m + [b]_m) + [c]_m$.
- *Elément neutre* : $[0]_m$ est l'élément neutre : $[a]_m + [0]_m = [a]_m$
- *Elément opposé* : $[-a]_m$ est l'opposé de $[a]_m$: $[a]_m + [-a]_m = [0]_m$
- *Commutativité* : $[a]_m + [b]_m = [b]_m + [a]_m$

La multiplication a les propriétés suivantes :

- *Associativité* : $[a]_m([b]_m[c]_m) = ([a]_m[b]_m)[c]_m$.
- *Elément neutre* : $[1]_m$ est l'élément neutre : $[a]_m + [1]_m = [a]_m$
- *Commutativité* : $[a]_m[b]_m = [b]_m[a]_m$

Les deux opérations ont la propriété suivante :

- *Distributivité* $[a]_m([b]_m + [c]_m) = [a]_m[b]_m + [a]_m[c]_m$

Un ensemble A muni de la loi $+$ et $\times$, qui satisfont les huit propriétés ci-dessus est appelé anneau commutatif. $(\mathbb{Z}, +, \times)$ et $(\mathbb{Z}/m\mathbb{Z}, +, \times)$ sont des anneaux commutatifs.

Bien évidemment on a aussi que, pour $k \in \mathbb{Z}$ et $a \in \mathbb{Z}$:

$$k[a]_m = [ka]_m$$

Définition. Si $ab = 0$ alors que ni a, ni b ne sont nuls, on dit qu'ils sont des diviseurs de zéros. On dira désormais que tout anneau, différent de l'anneau nul, qui ne possède aucun diviseurs de zéro est un anneau intègre.

Nous avons vu que la division pouvait poser problème dans $\mathbb{Z}/m\mathbb{Z}$, mais il est quand même possible de définir des éléments inversibles.

Théorème. *(Elément inversible)* Soit $m \geq 2$. On dit que $a \in \mathbb{Z}/m\mathbb{Z}$ est **inversible** s'il existe $a' \in \mathbb{Z}/m\mathbb{Z}$ tel que $aa' = [1]_m$. Un tel élément a', s'il existe, est unique. Il est appelé l'**inverse** de a et est noté a^{-1}.

Théorème. Si $a \in \mathbb{Z}/m\mathbb{Z}$ est inversible alors a^{-1} l'est aussi et

$$(a^{-1})^{-1} = a$$

Théorème. *(Algorithme d'Euclide)*. Soient a et b deux entiers avec $b \neq 0$, et soit $a = bq + r$ la division euclidienne de a par b. Alors

$$pgcd(a, b) = pgcd(b, r)$$

En particulier si $r = 0$ alors $pgcd(a, b) = b$

Théorème. *(Identité de Bézout)*. Soient a et b deux entiers, il existe deux nombres entiers u et v tels que

$$au + bv = pgcd(a, b)$$

En particulier si a et b sont premiers entre eux alors

$$au + bv = 1$$

Théorème. *(Eléments inversible de $\mathbb{Z}/m\mathbb{Z}$)*. Soient a et m deux entiers avec $m \geq 2$, $[a]_m$ est inversible si et seulement si a et m sont premiers entre eux.

Théorème. *(Cas de $\mathbb{Z}/p\mathbb{Z}$ avec p premier)*. Si p est premier, tous les éléments de $\mathbb{Z}/p\mathbb{Z}$ sauf $[0]_p$ sont inversibles.

Définition. Pour tout entier $m \geq 1$ on appelle indicatrice d'Euler $\varphi(m)$ le nombre d'éléments inversible de $(\mathbb{Z}/m\mathbb{Z}, \cdot)$, $\varphi(m)$ est donc égal au nombre de nombres d'entier n positifs qui sont inférieurs à m et premiers avec m.

Propriété de l'indicatrice d'Euler :

- $\varphi(p) = p - 1$
- $\varphi(mn) = \varphi(m)\varphi(n)\dfrac{d}{\varphi(d)}$ avec $d = pgdc(m, n)$

En particulier on a :

$$\varphi(2m) = \begin{cases} 2\varphi(m) & \text{si } m \text{ est pair} \\ \varphi(m) & \text{si } m \text{ est impair} \end{cases}$$

$$\forall k \geq 1, \varphi(n^k) = n^{k-1}\varphi(n)$$

4.4 Eléments d'algèbre abstraite

Définition. Soit $(G, \star)$ un ensemble G muni d'une opération binaire $\star$, c-à-d un mécanisme qui associe à deux éléments a et b de G, distincts ou non, un élément de G noté $a \star b$. Cette relation est appelée loi de composition interne du groupe G.

$(G, \star)$ est appelé groupe commutatif ou groupe abélien s'il vérifie les conditions suivantes :

1. *(Associativité)* $a \star (b \star c) = (a \star b) \star c$, pour tout $a, b, c \in G$
2. *(Neutre)* Il existe un élément $e \in G$ tel que $a \star e = e \star a = a$ pour tout $a \in G$
3. *(Symétrique)* Pour tout $a \in G$, il existe un élément $a' \in G$ tel que $a \star a' = a' \star a = e$. a' est appelé élément symétrique de a.
4. *(Commutativité)* $a \star b = b \star a$ pour tout $a, b \in G$

Si l'opération binaire est notée $+$, on note habituellement l'élément neutre 0 et l'élément symétrique de a est appelé opposé de a et noté $-a$.
Si l'opération binaire est notée $\times$ ou $\cdot$, on note habituellement l'élément neutre 1 et l'élément symétrique de a est appelé inverse de a et noté a^{-1}.

Théorème. Soit $\mathbb{Z}/m\mathbb{Z}^*$ l'ensemble des éléments inversibles de $\mathbb{Z}/m\mathbb{Z}$,pour $m \geq 2$. $(\mathbb{Z}/m\mathbb{Z}^*, \cdot)$ est un groupe commutatif.

Théorème. Soient $(G, \star)$ et $(H, \star)$ deux ensembles munis chacun d'une opération binaire. L'**opération produit** est l'opération binaire $\star$ définie sur l'ensemble $G \times H$ par

$$(a, b) \star (a', b') = (a \star a', b \star b')$$

Si $(G, \star)$ et $(H, \star)$ sont des groupes alors $(G \times H, \star)$ aussi. On l'appelle le **groupe produit**.

Définition. *(Isomorphisme)* Soient $(G, \star)$ et $(H, \otimes)$ deux ensembles munis chacun d'une opération binaire. Un isomorphisme de $(G, \star)$ vers $(H, \otimes)$ est une application $\psi : G \to H$ telle que

- ψ est bijective,
- $\psi(a \star b) = \psi(a) \otimes \psi(b)$, pour tous $a, b \in G$.

On dit que $(G, \star)$ et $(H, \otimes)$ sont isomorphes si il existe un isomorphisme de G vers H.

Ainsi, l'application

$$\begin{aligned}\mathbb{Z}/2\mathbb{Z} &\to \mathbb{Z}/4\mathbb{Z}^* \\ 0 &\mapsto 1 \\ 1 &\mapsto 3\end{aligned}$$

est un isomorphisme de $(\mathbb{Z}/2\mathbb{Z}, +)$ $(\mathbb{Z}/4\mathbb{Z}^*, \cdot)$

Supposons que ψ est un isomorphisme de $(G, \star)$ vers $(H, \otimes)$ alors :

1. Si G et H sont finis, ils ont le même cardinal,
2. Si $(G, \star)$ est une groupe alors $(H, \otimes)$ aussi, et réciproquement. Dans un tel cas, ψ transforme l'élément neutre de $(G, \star)$ en l'élément neutre de $(H, \otimes)$ et l'élément symétrique de a dans $(G, \star)$ en l'élément symétrique $\psi(a)$ dans $(H, \otimes)$,
3. l'application réciproque $\psi^{-1} : (H, \otimes) \to (G, \star)$ est aussi un isomorphisme.

Théorème. Soit $(G, \star)$ un groupe commutatif fini dans l'élément neutre est noté e.

1. Pour tout élément $a \in G$, il existe un entier $k \geq 1$ tel que $\underbrace{a \star a \star \ldots \star a = e}_{k \text{ fois}}$.

   Le plus petit de ces entiers est appelé **la période ou ordre** de a
2. Pour tout entier positif l, $\underbrace{a \star a \star \ldots \star a = e}_{l \text{ fois}}$ si et seulement si la période de a divise l

Théorème. Soient $(G, \star)$ et $(H, \otimes)$ deux groupes isomorphes et ψ un isomorphisme $G \to H$. Pour tout $x \in G$, la période de x est égal à $\psi(x)$

Théorème. *(Lagrange)*. Soit $(G, \star)$ un groupe commutatif de cardinal fini n. La période de tout élément de G divise n. En particulier, notons e l'élément neutre de G. Pour tout $a \in G$:

$$\underbrace{a \star a \star \ldots \star a = e}_{n \text{ fois}}$$

A savoir : Le cardinal de $\mathbb{Z}/m\mathbb{Z}^*$ est $\varphi(m)$.

Corollaire. *(Théorème d'Euler)*. Pour tout entier positif m et tout entier a premier avec m, on a :

$$([a]_m)^{\varphi(m)} = [1]_m$$

ou encore

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

Corollaire. *(Théorème de Fermat)*. Si p est un nombre premier, pour tout entier a, on a :

$$([a]_p)^p = [a]_p$$

ou encore

$$a^p \equiv a \pmod{p}$$

4.5 Cryptographie asymétrique

On définit l'application

$$\psi : \begin{cases} \mathbb{Z}/m_1m_2\mathbb{Z} & \to \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_2\mathbb{Z} \\ [k]_{m_1m_2} & \mapsto ([k]_{m_1}, [k]_{m_2}) \end{cases}$$

Théorème. *(Théorème des Restes Chinois)*. Soient m_1 et m_2 deux entiers ≥ 2

1. Si m_1 et m_2 sont premiers entre eux, l'application ψ définie au-dessus est bijective,
2. De plus c'est un isomorphisme à la fois pour l'addition et la multiplication,
3. Si m_1 et m_2 ne sont pas premiers entre eux, l'application ψ n'est ni surjective, ni injective.

Rappelons qu'un algorithme de chiffrement E, paramétré par une clé K, transforme le texte clair P en cryptogramme $C = E_K(P)$. L'algorithme de déchiffrement, paramétré par une clé k, effectue la transformation inverse $P = D_k(C)$. Nous allons désormais étudier un système de chiffrement ou la clé de chiffrement K est connu de tous alors que la clé de déchiffrement k est secrète et connu uniquement par le destinataire. Pour qu'un tel système fonctionne, il faut que certaine conditions soient remplies :

1. *(Exactitude)* L'algorithme de déchiffrement doit rétablir le texte clair : $D_k(E_K(P))$
2. Le chiffrement $E_K(P)$ d'un message clair P est une opération aisée et rapide
3. Le déchiffrement $D_k(C) = P$ d'un cryptogramme C est une opération aisée et rapide pour quiconque connaît la clé k
4. Par contre, le déchiffrement sans la connaissance de k est extrêmement difficile, et impossible à effectuer en temps raisonnable.
5. Enfin, il doit extrêmement difficile de deviner la clé privée k.

Le système RSA satisfait à ces conditions.

L'algorithme de Rivest-Shamir-Adleman (RSA) encode tous les messages (clairs ou chiffrés) comme des entiers modulo K (c-à-d des éléments de $\mathbb{Z}/K\mathbb{Z}$. Les longs messages sont subdivisés en blocs, si bien que tous les messages sont compris entre 0 et $K-1$.

1. Le module K est toujours de la forme $K = pq$ où p et q sont des nombres premiers. Le module K est la clé publique.
   La clé privée k est le plus petit multiple commun à $p-1$ et $q-1$
   Les facteurs p et q sont secrets et utilisés seulement pour construire la clé publique K. Ils peuvent être détruits une fois k calculée.
2. Le chiffrement est défini par $C := E_K(P)$ tel que $[C]_K = ([P]_K)^e$ ou l'exposant e est connu et public.
3. La clé publique K doit être telle que e est premier avec k
   En pratique, on prend souvent $e = 65537$, qui est premier, donc il suffit que e ne divise ni $(p-1)$ ni $(q-1)$.
4. Le déchiffrement est obtenu à l'aide d'un nombre f tel que $[f]_k = [e]_k^{-1}$. Notons que cet inverse existe précisément car e est premier avec k. Il peut être calculé simplement avec l'identité de Bézout, si on connaît la clé privée k.
   Le déchiffrement se fait alors par $P' = D_k(C)$ avec $[P']_K = ([C]_K)^f$

Théorème. *(Exactitude de RSA).* Soient p et q deux nombres premiers distincts. Soit m un multiple commun à $p-1$ et $q-1$. Pour tout entier n :

$$([n]_{pq})^{1+m} = [n]_{pq}$$

On admettra que le système RSA vérifie les conditions sus-nommées.

Cependant, il est nécessaire de bien choisir les nombres entiers p et q, sans quoi certains problèmes peuvent survenir, en particulier celui des messages non cachés. En général les textes clairs P qui ne sont pas cachés sont ceux qui satisfont

$$([P]_K)^e = [P]_K$$

Il peut y avoir beaucoup de solutions à cette équation en P, sauf si on choisit pour p et q des nombres premiers "sûrs".

Définition. Un nombre premier p est dit sûr s'il est de la forme $p = 2p' + 1$ où p' est aussi un nombre premier.

Le théorème suivant dit que dans un tel cas, et avec un exposant de chiffrement (e) bien choisi, le nombre de message P qui ne sont pas cachés est 9. Comme $K = pq$ est très grand, leur nombre est infime et il doit donc être possible d'éviter de tels messages avec grande probabilité.

Théorème. Soient p et q des nombres premiers sûrs disctints supérieurs à 5 et supposons que l'exposant de chiffrement e est tel que $e - 1$ est une puissance de 2. Le nombre de solutions $[P]_K \in \mathbb{Z}/K\mathbb{Z}$ de l'équation $([P]_K)^e = [P]_K$ est égal à 9

A savoir : Pour signer un document t, on ajoute à ce dernier une fonction de hachage $h(t)$ de telle sorte que seul le signataire ait pu le faire. On peut le faire en utilisant des fonctions à sens unique comme cela :

- Soit t le texte clair que Alice veut signer,
- Soit f_A la fonction à sens unique de Alice (connu de tous)
- Soit h la fonction de hachage (connu de tous, la même pour tout le monde)
- La signature digitale est alors $s = f_A^{-1}(h(t))$
- Le document signé est désormais (t, s)

5 Codes correcteurs d'erreurs

5.1 Les codes correcteurs ou détecteurs

Pour protéger l'information, l'idée est d'ajouter des bits (appelés bits de redondance, en utilisant un code correcteur ou détecteur. Ces bits de redondance sont utilisés lors du décodage pour reconstruire l'information initiale même s'il y a des bits perdus ou erronés. Les chiffres de contrôle MOD 97-10 de l'IBAN que nous rencontrés sont un tel code. Il est peu efficace, et nous allons voir comment fabriquer des codes bien meilleurs. En particulier, nous allons construire les codes de Reed-Salomon, qui sont très utilisés.

Définition. *(Code en bloc)* Un code en bloc de longueur n, défini sur un alphabet $\mathcal{A}$, est un sous-ensemble $\mathcal{C}$ de $\mathcal{A}^n$, c-à-d un ensemble de suites de n éléments de $\mathcal{A}$. Les éléments du code sont appelés mots de code.

Le rendement ou débit du code est défini par

$$r = \frac{1}{n} \log_{card(\mathcal{A})} card(\mathcal{C})$$

Pour évaluer l'efficacité d'un code on utilise la *distance de Hamming* et la *distance minimale*.

Définition. *(Distance de Hamming)*. Soit $\mathcal{A}$ un ensemble fini et $n \geq 1$ un entier. Soient $x = (x_1, ..., x_n) \in \mathcal{A}^n$ et $y = (y_1, ..., y_n) \in \mathcal{A}^n$ deux suites de n éléments de $\mathcal{A}$. La distance de Hamming $d(x, y)$ est le nombre de positions où x et y diffèrent :

$$d(x, y) := card\{i \in \{1, ..., n\} \text{ tels que } x_i \neq y_i\}$$

Par exemple

$$x = (1, 0, 1, 1, 1, 0)$$
$$y = (1, 0, 0, 1, 1, 1)$$

ne diffèrent qu'en leur troisième et dernière position donc leur distance de Hamming est $d(x, y) = 2$.

Théorème. *(Distance de Hamming)*. La distance de Hamming possède les trois propriétés suivantes. Pour tout $x, y, z \in \mathcal{A}^n$

1. $d(x, y) \geq 0$ et $d(x, y) = 0$ si et seulement si $x = y$
2. *(symétrie)* $d(x, y) = d(y, x)$
3. *(inégalité triangulaire)* $d(x, z) \leq d(x, y) + d(y, z)$

On pourra alors dire que l'ensemble $\mathcal{A}^n$ muni de la distance de Hamming est un **espace métrique**.

Définition. *(Distance minimale)*. La distance minimale d'un code en bloc $\mathcal{C}$, notée $d_{min}(\mathcal{C})$, est

$$d_{min}(\mathcal{C}) := \min_{x,y \in \mathcal{C}, x \neq y} d(x, y)$$

Autrement dit, la distance minimale d'un mot d'un code est la plus petite distance de Hamming entre deux mots de code distincts. La distance minimale reflète bien la capacité d'un code à détecter ou corriger des erreurs, et un bon code est un code qui a une grande distance minimale.

Définition. *(Canal à effacements)*. Pour le canal à effacements, nous supposons que chaque composante du mot de code est soit connue parfaitement, soit effacée. L'effacement signifie que le symbole transmis à une position effacée a été remplacé par un symbole spécial, disons le symbole " ?". Le destinataire sait quelles positions

ont été effacées, mais ne sait pas quelles étaient présents avant l'effacement. Le poids d'un effacement est le nombre de positions qui sont modifiées.

Définition. *(Canal à erreur)*. Pour le canal à erreurs, chaque composante du mot de code est soit reçue parfaitement, soit échangée pour un autre symbole de l'alphabet. Le destinataire ne sait pas quelles positions sont victimes d'erreurs. Le poids d'une erreur est le nombre de positions qui sont modifiées.

Théorème. *(Détection d'erreurs)*.

1. Un code $\mathcal{C}$ est capable de détecter toutes les erreurs de poids $p < d_{min}(C)$.
2. Inversement, si un code $\mathcal{C}$ peut détecter toutes les erreurs de poids $\leq p$ alors $p \leq d_{min}(C)$

Théorème. *(Correction d'effacements)*.

1. Un code $\mathcal{C}$ est capable de corriger tous les effacements de poids $p < d_{min}(C)$.
2. Inversement, si un code $\mathcal{C}$ peut corriger tous les effacements de poids $\leq p$ alors $p \leq d_{min}(C)$

Par exemple, si $x = (0100111)$ et le mot reçu $y = (0?001?1)$. Le poids de l'effacement est 2 et nous savons que la distance minimale de ce code est 3, donc cet effacement peut être corrigé. Effectivement, en inspectant la liste de tous les mots du code, nous voyons que $x = (0100111)$ est le seul mot de code compatible avec y.

Théorème. *(Correction d'erreurs)*.

1. Un code $\mathcal{C}$ est capable de corriger tous les erreurs de poids $p < \dfrac{d_{min}(C)}{2}$.
   Plus précisément, si l'erreur est de poids $p < \dfrac{d_{min}(C)}{2}$, alors le mot transmis x est le mot le proche du mot reçu y.
2. Inversement, si un code $\mathcal{C}$ peut corriger tous les effacements de poids $\leq p$ alors $p < \dfrac{d_{min}(C)}{2}$

Enfin nous terminons par une inégalité qui montre que la distance minimale ne peut pas être arbitrairement grande.

Théorème. *(Borne de **Singleton**)*. Pour un code en bloc $\mathcal{C}$ de longueur n et de rendement r la distance minimale satisfait

$$d_{min}(\mathcal{C}) \leq n(1-r)+1$$

Les codes qui vérifient cette égalité sont alors appelés des codes **MDS** pour *Maximal Distance Separable*.

5.2 Corps finis et espaces vectoriels

Il peut parfait être compliqué de calculer la distance minimale d'un code ou d'en concevoir avec des distances minimales aussi grande que possible. Pour résoudre ce problème, nous allons utiliser des codes linéaires sur des corps finis.

Définition. Soit $(\mathcal{K}, +, \cdot)$ un ensemble muni de deux opérations binaires notées $+$ et $\cdot$. Nous disons que c'est un corps commutatif (en anglais field) si :

1. L'addition fait de $\mathcal{K}$ un groupe commutatif. Son élément neutre est noté 1.
2. La multiplication fait de l'ensemble $\mathcal{K}$ privé de 0 un groupe commutatif. En particulier, tous les éléments sauf 0 sont inversibles. L'élément neutre de la multiplication est noté 1.
3. La multiplication est distributive par rapport à l'addition : $a(x+y) = ax + ay$

Un premier exemple de corps commutatif est l'ensemble $(\mathbb{Z}/p\mathbb{Z}, +, \cdot)$, avec p premier. En effet si p n'est pas premier alors il existe des éléments qui ne sont pas inversibles. Par exemple dans $(\mathbb{Z}/6\mathbb{Z})$, $[3]_6$ n'a pas d'inverse alors que $[3]_6 \neq [0]_6$.

Théorème. Dans un corps fini, il existe un plus petit entier $p > 0$ tel que $p \cdot 1 = 0$, c-à-d tel que

$$\underbrace{1+1+\ldots+1=0}_{p \text{ fois}}$$

Ce nombre p est premier. Il est appelé **la caractéristique** du corps.
Intuitivement, il représente le nombre de fois que nous devons composer l'élément neutre de la multiplication avec l'addition pour obtenir l'élément neutre de l'addition.

Ce nombre p est assuré d'exister puisqu'il s'agit en fait de la période de 1 dans le groupe $(\mathcal{K}, +)$.

Théorème.

1. Le cardinal d'un corps fini est une puissance de sa caractéristique.
2. Tous les corps finis de même cardinal sont isomorphes.
3. Pour tout nombre premier p et tout entier $m \geq 1$, il existe un corps fini de cardinal p^m.

Définition. Soit $\mathcal{K}$ un corps commutatif et $(\mathcal{V}, +)$ un groupe commutatif, muni de la loi $+$. Supposons qu'une loi de composition externe est définie sur $\mathcal{K}$ et $\mathcal{V}$, c-à-d une application qui à $\lambda \in \mathcal{K}$ et $\vec{x} \in \mathcal{V}$ associe un élément, noté $\lambda\vec{x}$ de $\mathcal{V}$. Cette opération externe est appelé multiplication scalaire. Nous disons que $\mathcal{V}$ muni de ces deux opérations est un espace vectoriel sur le corps $\mathcal{K}$ (ou encore que $\mathcal{V}$ est un $\mathcal{K}$-espace-vectoriel) si les trois propriétés suivantes sont vraies pour tous scalaires λ, μ et vecteurs $\vec{u}, \vec{v}$

- *associativité* pour la multiplication scalaire : $\lambda(\mu\vec{v}) = (\lambda\mu)\vec{v}$,
- *identité* : $1 \cdot \vec{v} = \vec{v}$
- *distributivité* : $\lambda(\vec{u} + \vec{v}) = \lambda\vec{u} + \lambda\vec{v}$ et $(\lambda + \mu)\vec{u} = \lambda\vec{u} + \mu\vec{u}$

Les vecteurs $\vec{v_i}$ sont linéairement indépendants si et seulement si

$$\sum_{i=1}^{m} \lambda_i \vec{v_i} = 0 \Rightarrow \lambda_i = 0, i = 1, ..., m$$

Une suite de vecteur $\vec{v_i}, i = 1, ..., m$ est une base de $\mathcal{V}$ si les vecteurs sont linéairement indépendants et si ils engendrent tout l'espace vectoriel $\mathcal{V}$. Cela implique que pour tout vecteur de $\mathcal{V}$ on peut le décomposer en une combinaison linéaire des éléments de la bases. Les coefficients d'une telle combinaison s'appellent les coordonnées du vecteur relativement à cette base.

On définit la dimension d'un espace vectoriel $\mathcal{V}$ comme le cardinal d'une base de $\mathcal{V}$. On le note $\dim(\mathcal{V})$. Le concept de dimension possède quelques propriétés intéressantes :

1. si une suite de $n = \dim(\mathcal{V})$ vecteurs est linéairement indépendante, alors elle engendre $\mathcal{V}$.
2. si une suite de $n = \dim(\mathcal{V})$ vecteurs engendre $\mathcal{V}$, alors elle est linéairement indépendante.

Théorème. Si $\mathcal{V}$ est un espace vectoriel de dimension n sur un corps fini $\mathcal{K}$, alors $\mathcal{V}$ est fini et $card(\mathcal{V}) = [card(\mathcal{K})]^n$.

Le rang d'une matrice rectangulaire A à coefficient dans $\mathcal{K}$ est la dimension du sous-espace vectoriel engendré par les colonnes / lignes de A. On rappelle les résultats suivant :

1. Le rang de A est égal à la dimension du sous-espace engendré par les colonnes/lignes de A.
2. Si A est de taille $n \times m$ alors $rg(A) \leq \min(n, m)$.
3. Le rang d'une matrice triangulaire $n \times n$ dont les termes diagonaux sont non-nuls est n.
4. Le rang de A peut être calculer avec la méthode du pivot de Gauss.

5.3 Codes linéaires

Définition. Soit $\mathcal{C}$ un code en bloc de longueur n. Nous disons que $\mathcal{C}$ est un code linéaire si

1. L'alphabet du code est un corps fini $\mathcal{K}$.
2. Le code est un sous-espace vectoriel de $\mathcal{K}^n$

Puisqu'un code linéaire est un s.e.v, il a une dimension, que nous noterons k. Le nombre de mots de code est donc card$(\mathcal{K}^k)$. Donc le rendement d'un tel code est $r = \frac{k}{n}$ et la borne de Singleton devient donc $d_{min}(\mathcal{C}) \leq n - k + 1$

La première simplification qu'apporte un code linéaire est que sa distance minimale peut être calculer plus efficacement

Théorème. Si $\mathcal{K}$ est un corps fini, le **poids de Hamming** de $\vec{x} \in \mathcal{K}^n$ est le nombre de composantes non nulles, c-à-d aussi $w(\vec{x}) := d(\vec{0}, \vec{x})$. La distance minimale d'un code linéaire $\mathcal{C}$ est égale à

$$d_{min}(\mathcal{C}) = \min_{\vec{x} \in \mathcal{C}, \vec{x} \neq 0} w(\vec{x})$$

Autrement dit, la distance minimale est le plus petit poids de Hamming d'un mot de code non nul.

Définition. Soit $\mathcal{C}$ un code linéaire sur le corps $\mathcal{K}$, de longueur n et de dimension k. Soit $(\vec{v_1}, ..., \vec{v_k})$ une base de $\mathcal{C}$. La matrice obtenue en écrivant à la *i-ième* ligne le vecteur $\vec{v_i}$ est appelée matrice génératrice du code.

On note G cette matrice. Soit $\vec{u} = (u_1, ..., u_k)$ un mot quelconque à encoder, alors le mot de correspondant $\vec{x}$ s'obtient par

$$\vec{x} = \vec{u}G$$

Le sous espace vectoriel engendré par les lignes de G reste inchangé si l'on permute ces lignes ou si l'on ajoute une ligne à l'autre. Ainsi, nous pouvons appliquer des opérations élémentaires sur les lignes de G dans le but de mettre G dans un forme plus simple à utiliser.

Définition. Une matrice G à k lignes et $n > k$ colonnes est dites sous forme systématique si

$$G = [I_k \, P]$$

où I_k est la matrice identité de taille k et P une matrice de dimension $k \times (n-k)$.

Définition. Soit $\mathcal{C}$ un code linéaire sur un corps $\mathcal{K}$, de longueur n et de dimension k. Une matrice de contrôle du code est une matrice de taille $(n-k) \times n$ et dont les lignes sont les vecteurs de coefficients $n-k$ équations linéaires qui définissent le s.e.v $\mathcal{C}$ dans $\mathcal{K}^n$. Ces $n-k$ lignes sont nécessairement indépendantes. En écriture matricielle, H est une matrice de contrôle du code si l'équation

$$\vec{x}H^T = \vec{0}$$

définit les mots de code $\vec{x}$.

A savoir : Pour trouver une matrice de contrôle, il faut mettre le s.e.v en équations.

Une matrice de contrôle permet de détecter simplement des erreurs. Soit $\vec{x}$ le mot de code transmis et $\vec{y}$ le mot de code reçu. Le syndrome est par définition

$$\vec{s} = \vec{y}H^T$$

S'il n'y a pas d'erreur, le syndrome est égal à 0. Une méthode simple de détection consiste donc à déclarer une erreur si le syndrome est non-nul.

Théorème. Si la matrice génératrice G est sous forme systématique $G = [I_k \, P]$ alors une matrice de contrôle est

$$H = [-P^T \, I_{n-k}]$$

5.4 Codes de Reed-Solomon

Définition. *(Polynômes)*. Soit $\mathcal{K}$ un corps fini. Un polynôme P à coefficients dans $\mathcal{K}$ est une application $\mathcal{K} \to \mathcal{K}$ de la forme

$$X \mapsto P(X) = a_1 + a_2 X + ... + a_{m+1} X^m$$

où $a_1, ..., a_{m+1}$ sont des éléments de $\mathcal{K}$. Le degré du polynôme est la plus grande puissance affectée d'un coefficient non nul.

Pour tout suite $\vec{u} = (u_1, ..., u_k) \in \mathcal{K}^k$ de k élément de $\mathcal{K}$ nous appelons $P_{\vec{u}}$ le polynôme dont les coefficients sont $u_1, ..., u_k$, par ordre de puissance croissante, autrement dit

$$P_{\vec{u}} = u_1 + u_2 X + ... + u_k X^{k-1}$$

et donc le polynôme est de degré $\leq k - 1$. Nous pouvons maintenant définir les codes de Reed-Solomon.

Définition. *(Reed-Solomon)*. Soient n et k des entiers avec $1 \leq k \leq n$. Un code de Reed-Solomon de paramètre (n, k) est défini comme suit :

1. L'alphabet est un corps fini $\mathcal{K}$ de cardinal $\geq n$.
2. Choissions une suite de n éléments distincts de $\mathcal{K}$, $(a_1, ..., a_n)$. Une suite de k symboles $\vec{u} = (u_1, ..., u_k) \in \mathcal{K}^k$ est encodée en la suite de n symboles $\vec{x} = (x_1, ..., x_n) \in \mathcal{K}^n$ définie par

   $$x_i = P_{\vec{u}}(a_i) \quad \text{pour } i = 1, ..., n$$

   Le code de Reed-Solomon $\mathcal{C}$ est l'ensemble des tous les encodages $\vec{x}$ possibles, pour tous les $\vec{u} \in \mathcal{K}^k$. C'est donc un code en bloc de longueur n.

Rappel : Un polynôme a un nombre de racines inférieurs ou égal à son degré.

Théorème. Un code de Reed-Solomon de paramètres (n, k) est un code en bloc linéaire de taille n et de dimension k.

Théorème. Un code de Reed-Solomon de paramètres (n, k) a pour distance minimale $d_{min} = n - k + 1$. En d'autres termes, il atteint la borne de Singleton, et sa distance minimale est la plus grande possible pour un code en bloc de longueur n et de dimension k.