REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF DE M'SILA

FACULTE DES SCIENCES ET SCIENCES DE L'INGENIEUR

DEPARTEMENT DE MATHEMATIQUES

Mémoire présenté Par :

LADJELAT LAHCENE

EN VUE DE l'obtention du diplôme de

MAGISTER

OPTION : MATHEMATIQUES DISCRETES

Sujet de la thèse

ETUDE DE L'EQUIVALENCE DE DEUX CODES SUR UN CORPS FINI

Thèse soutenue Le : .... / .... / 2004

Devant la commission d'examen composée de :

Mr : BEN SALEM Naceurdine	**Président**	**M.C .Université de setif**
Mr: MIHOUBI DOUADI	**promoteur**	**M.C .Université de M'sila**
Mr : BOUDAOUD Abdelmadjid	**Examinateur**	**M.C .Université de M'sila**
Mr : AMROUNE Abdelaziz	**Examinateur**	**M.C .Université de M'sila**
Mr : MEZRAG Lahcene	**Examinateur**	**M.C .Université de M'sila**
Mr : MIDOUNE Noureddine	**Invité**	**M.A.C.C .Université de M'sila**

ملخص

شفرتان متكافئتان بتبديلة إذا كانت إحداهما تساوي الأخرى بتبديل الإحداثيات ، نقدم هنا دراسة تسمح بحساب هذه التبديلة . نقدم مفهوم التوقيع كخاصية لوضعية شفرة . من أجل تحديد التبديلة بين شفرتين متكافيئتين يكفي إيجاد توقيع مميزا تماما . إهتمامنا منصب حينما يكون هذا التوقيع غير مميز تماما.

الكلمات المفاتيح : شفرة ، تكافؤ ، تبديلة ،توقيع ، لامتغير ، معدد الوزن

Résumé

Deux codes sont équivalents par permutation s'ils sont égaux à une permutation prés de leurs coordonnées , nous présentons ici une étude capable de calculer cette permutation. Nous introduisons la notion de signature comme une propriété d'une position d'un code. Pour déterminer la permutation entre deux codes equivalents , il suffit de trouver une signature totalement discriminante. Notre intention porte sur le cas où cette signature n'est pas totalement discriminante

Mots clés : *code , équivalence , permutation , signature , invariant , enumérateur de poids* .

Abstract

Two codes are permutation equivalent if they are equal up to a fixed permutation of the code words coordinates , we present here a study able to compute this permutation , we introduce the concept of signature : a property of a position of a code. To compute the permutation between two équivalent codes , one needs a signature which is fully discriminant , our intention stand over the case which the signature is not fully discriminant .

Key words : *Code, equivalence , permutation , signature , invariant , weight enumerator* .

ملخص الرسالة باللغة العربية

العنوان : دراسة حول تكافؤ شفرتين على حقل منته

ETUDE DE L 'EQUIVALENCE DE DEUX CODES SUR UN CORPS FINI

تتناول هذه الأطروحة موضوعا هاما جدا في نظرية الشفرات المصححة للأخطاء وهو موضوع دراسة تكافؤ شفرتين على حقل منته وإيجاد التبديلة التي تسمح بالانتقال من شفرة إلى أخرى مكافئة لها .

بحيث تطرقنا في الفصل الأول للمفاهيم العامة حول نظرية الزمر، الحقول المنتهية وكيفية بنائها ونظرية الفضاءات الشعاعية، ويعتبر هذا الفصل الأداة الرياضية لمتابعة الفصول الأخرى.

أما في الفصل الثاني فقد تطرقنا إلى مبادئ نظرية الشفرات المصححة للأخطاء والمعرفة على حقول منتهية وهو يعتبر الخطوة الأولى لهذه النظرية وكذلك يعتبر مهما لدراسة الفصول المتبقية وذكرنا نتائج خاصة بالشفرات الخطية، والشفرات الدورية .

أما في الفصل الثالث فقد تطرقنا إلى دراسة بنوع من الاسهاب لزمرة تبديلات شفرة، ولتكافؤ شفرتين ثم ذكرنا خواصا مهمة تتعلق بالمفهومين السابقين.

في الفصل الأخير: ذكرنا كيف يمكن حساب التبديلية التي تعين التكافؤ وذلك باستعمال مفهوم اللامتغير والتوقيع (*invariant et signature*) المعرفين من طرف ساندريي نيكولا N. Sendrier وكذلك كيف يمكن لتوقيع مميزا تماما أن يعين هذه التبديلية .

ثم تطرقنا للحالة التي لا يكون فيها هذا التوقيع مميزا تماما ودرسناها وبينا أنه في حالة كون تطبيق معرف بعناية متباينا فإنه يمكن في هذه الحالة حساب هذه التبديلية وصدقنا ذلك بمثال.

وفي الخاتمة ذكرنا بعض التساؤلات التي طرحت علينا أثناء قيامنا بهذا العمل آملين أن تكون مسائل جادة للبحث مستقبلا .

Dédicaces

A mes parents

A ma famille

A Ahmed L et à toute sa famille

A tous mes amis

A R. Azedine & Redouane

Remerciements

Je tiens à remercier vivement mon promoteur D^r : MIHOUBI. D, d'avoir accepté de diriger ce travail et de créer autour de moi un environnement de recherche par ses consiels et son soutien permanent .

Comme je tiens à remercier monsieur le D^r : BEN SALEM. N pour avoir accèpté la présidence du jury .

Je remercie également messieurs les docteurs : BOUDAOUD. A, AMROUNE. A, MEZREG. L, MIDOUNE. N, pour avoir accepté de juger ce travail et de faire partie du jury

Je ne peux oublier de remercier tous les docteurs d'avoir contribué à notre formation durant l'année de la post-graduation, ainsi qu'à toute l'équipe du département de mathématiques.

Enfin , je remercie tous les amis qui m'ont aidé pour l'élaboration de ce travail.

Etude de l'équivalence de deux codes sur un corps fini

Plan de travail

- Introduction
- Codes correcteurs d’erreurs
- Equivalence des codes
- Determination de l’équivalence
- Exemple
- conclusion

SOMMAIRE

NOTATIONS

$|G|$: L'ordre d'un groupe fini *G* ou le cardinal d'un ensemble fini *G*.
$[G : H]$: L'indice du sous groupe *H* dans *G*.
$\boldsymbol{Z}$: L'ensemble des entiers relatifs.
$\boldsymbol{N}$: L'ensemble des entiers naturels.
$\boldsymbol{Z} / n\boldsymbol{Z}$: L'ensemble des entiers modulo $n \in N^*$.
S_n : Le groupe symétrique de degré *n*.
$\bar{x}$: La classe (*l'orbite*) de *x* modulo une relation d'équivalence.
K^* : Le groupe multiplicatif d'un corps *K* avec $K^* = K - \{0\}$
F_q : Un corps fini de cardinal *q*.
$A[x]$: L'anneau des polynômes à une indéterminée *x* sur un anneau *A*.
(f) : L'idéal engendré par *f* dans $A[x]$.
$\cong$: Isomorphisme de groupes, de corps, d'espaces vectoriels,…
$[x]$: La partie entière d'un réel *x*.
$\omega(x)$: Le poids de *Hamming* d'un mot *x*.
$rg\ H$: Le rang d'une matrice *H*.
$Ker\ H$: L'espace nul d'une matrice *H*.
I_k : Matrice identique de taille $k \times k$.
${}^t A$: La transposée d'une matrice *A*.
$A[x]/(f)$: L'anneau quotient de $A[x]$ modulo *f*.
$< x, y >$: Le produit scalaire de *x* et *y*.
$C^{\perp}$: Le dual d'un code *C*.
$W_C(x,y)$ *ou* $W(x,y)$: Le polynôme énumérateur des poids d'un code *C*.
I : Un ensemble ordonné de cardinal *n*.
$\sigma(c)$: L'action de $\sigma \in S_n$ sur le mot *c*.
$\sigma(C)$: L'action de $\sigma \in S_n$ sur le code *C*.
$\sigma(i)$: L'image de $i \in I$ par σ.
Perm (C) : Le groupe de permutations d'un code *C*.
$S_q(I)$: Le groupe des permutations monomiales de *I*.
Aut (C) : Le groupe d'automorphismes d'un code *C*.
C^* : Le code étalé de *C*.
$C \sim C'$: Les codes *C* et C' sont équivalents par permutation.
C_i : Le code C poinçonné en *i*.
$\nu\ (C)$: L'image de *C* par l'invariant ν.
$S(C,i)$: L'image du couple *(C, i)* par la signature *S*.

Notations et définitions

Soit E un ensemble non vide

l'ensemble $S(E)$ des bijections de E sur E, muni de la loi de composition des applications, est un groupe appelé le groupe symétrique de E.

Si E est fini, de cardinal $n \geq 1$,on note S_n le groupe symétrique de E. les éléments de S_n sont appelés des permutations de E.

Action d'un groupe fini sur un ensemble

G désigne un groupe fini et E un ensemble fini non vide.

Définition :

le groupe G opère sur l'ensemble E s'il existe une application $\Phi : G \times E \rightarrow E$

Ou $\phi(g,x)$ sera noté $g.x$, satisfaisante, pour tous $g_1, g_2 \in G, x \in E$ aux conditions :

$i)\ g_1 . (g_2 . x) = g_1 g_2 . x$

$ii)\ 1.x = x$ ou 1 est l'élément neutre de G

Définition

Un corps de q éléments est dit un *corps fini* de cardinal q

On note souvent F_q un corps fini à q éléments.

Codes correcteurs d'erreurs

2. les codes

Soit *Q* un ensemble non vide à *q* éléments, *Q* sera appelé un alphabet.

Soient *k* et *n* deux entiers naturels non nuls avec $k \le n$ ..

Un $k-uplets$ *a* de Q^k sera appelé un message ou un mot de longueur *k*, et sera noté

soit : $a = (a_1, a_2, ..., a_k)$ soit : $a = a_1 a_2 ... a_k$.

L'ensemble des messages sera une partie *E* de Q^k .

On introduit ainsi une application injective :

$$f : \quad E \to Q^n$$
$$a = (a_1, a_2, ..., a_k) \mapsto c = (c_1, c_2, ..., c_n)$$

appelée application de codage ou encodeur

Notons $C = f(E)$ l'image de *f*.

C est appelé code de longueur *n* sur *Q* , et les éléments de *C* s'appellent les mots du code.

Le cardinal du code est par définition celui de *C*.

Distance de Hamming

La *distance de Hamming* entre deux mots $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$, que l'on notera $d(x, y)$,est le nombre d'indices *i* tels que $x_i \neq y_i$.

C'est-à-dire :

$$d(x, y) = \left| \{ i / x_i \neq y_i \} \right| .$$

Distance minimale d'un code

La distance minimale d'un code *C* est la distance minimum entre deux mots distincts de ce code. On la note *d* :

$$d = Min\{d(x,y) / x,y \in C \text{ et } x \neq y\}.$$

Dans la suite , nous prenons $Q = F_q$

Le poids de Hamming

Le poids de Hamming d'un mot $x=(x_1,x_2,...,x_n)$ de F_q^n, noté $\omega(x)$, est le nombre d'indices i tels que $x_i \neq 0$.

$$\omega(x) = |\{i / x_i \neq 0\}|.$$

Polynôme énumérateur des poids

Définitions

1) La distribution de poids d'un code *C* de longueur *n* sur un corps fini F_q est la suite $A_0,A_1,...,A_n$ où chaque A_i ; $0 \leq i \leq n$, est le nombre de mots dans *C* de poids *i* .

2) Le polynôme énumérateur $W_C(x,y)$ de *C* est le polynôme de $Z[x,y]$ défini par *:*

$$W_C(x,y) = \sum_{i=0}^{n} A_i x^{n-1} y^i.$$

Equivalence de deux codes

Groupe de permutations d'un code

Soient n un entier positif non nul et q une puissance d'un nombre premier

soit $\boldsymbol{I}$ un ensemble ordonné de cardinal n utilisé pour indexer les cordonnées des mots de F_q^n (dans la suite nous prenons $\boldsymbol{I} = \{1,2,...n\}$)

une permutation $\sigma \in S_n$ agit sur les mots de F_q^n comme suit :

si $c = (c_i)_{i \in I}$ est un mot de F_q^n, alors :

$$\sigma(c) = (c_{\sigma(i)})_{i \in I} = (c_{\sigma(1)}, c_{\sigma(2)}, ..., c_{\sigma(n)})$$

Soit maintenant C un code de longueur n sur F_q .

La permutation σ de S_n définit une action sur le code C comme suit :

$$(\sigma, C) \mapsto \sigma(C)$$

avec $$\sigma(C) = \{\sigma(c) / c \in C\} .$$

Notation :

Soit C un code de longueur n sur F_q **.**

Notons $perm(C)$ le sous ensemble de tous les éléments σ de S_n ;tels que $\sigma(C) = C$.

i .e :

$$perm(C) = \{\sigma \in S_n / \sigma(C) = C\}.$$

Proposition :

L'ensemble $perm(C)$, *muni du produit usuel des permutations, est un sous groupe de* S_n .

Définition

Le sous groupe $perm(C)$ de S_n est appelé le groupe de permutations du code C .

Equivalence par permutation

Soit $\mathbb{C}_n$ l'ensemble des codes de longueur *n* sur F_q .
Définissons l'application φ de $S_n \times \mathbb{C}_{n\,//}$ dans $\mathbb{C}_n$ par :

$$\varphi(\sigma, C) = \sigma(C)$$

avec :

$$\sigma(C) = \{\sigma(x) / \, x \in C\}.$$

Proposition

L'application φ définie une opération de S_n sur $\mathbb{C}_n$. (c'est-à-dire S_n opère sur $\mathbb{C}_n$.)

Soit ~ la relation sur $\mathbb{C}_n$ définie par :

Pour deux codes C et C' de $\mathbb{C}_n$;

$$C \sim C' \Leftrightarrow \exists \sigma \in S_n ; C' = \sigma(C) \text{ .}$$

Proposition

La relation $\sim$ définie sur $\mathbb{C}_n$ est une relation d'équivalence.

Définition

Deux codes de même longueur n sur F_q ,sont équivalents par permutations s'ils sont équivalents au sens de la relation $\sim$ définie ci – dessus .

Cela revient à dire que deux codes C et C' de même longueur sont équivalents par permutation s'il existe une permutation $\sigma \in S_n$ telle que :

$$C' = \sigma(C).$$

Proposition

Le nombre des codes équivalents par permutation à un code C de longueur n est

$$\frac{n!}{|perm(C)|} \quad .$$

Proposition (résultat original)

Soit C un code de longueur n sur F_q*. alors :*

1) *Le nombre des permutations de* S_n *qui produisent un même code équivalent par permutation à C , est* $|perm(C)|$.
2) *Si* C' *est un code équivalent à C par une permutation* σ *, alors* σ *est unique si et seulement si le groupe de permutation de C est trivial (c'est à dire réduit à l'identité).*

Proposition

Deux codes de même longueur équivalents par permutation, ont des groupes de permutation isomorphes.

Proposition

Deux codes équivalents par permutation Ont :

1) *même longueur .*
2) *même distribution des poids .*
3) *même distance minimale .*
4) *même polynôme énumérateur des poids.*

Determination de l'équivalence de deux codes

Il s'agit ici d'essayer de répondre aux deux questions suivantes :

1. Etant donné deux codes C et C', décider si C et C' sont équivalents par permutation ?
2. Si C et C' sont équivalents par permutation, retrouver la permutation σ telle que $\sigma(C)=C'$?.

.Notations et définitions

I désigne toujours l'ensemble $\{1,2,\ldots,n\} \subset N$.

Pour tout $x \in F_q^n$, $\text{supp}(x)=\{i \in I : x_i \neq 0\}$ désigne le support de x .

Codes poinçonnés

Soit C un code de longueur n sur F_q .

Si J est une partie non vide de I

$$E_J=\{x \in F_q^n : \text{supp}(x) \subset J\}$$

Définition

Le code C poinçonné en i est par définition :

$$C_i = (C + E_i) \cap E_{I \setminus \{i\}}$$

Equivalence : Pour tout code C de longueur n , pour tout $i \in I$, et pour toute permutation $\sigma \in S_n$., nous avons :

$$\sigma(C_i) = \sigma(C)_{\sigma(i)} \ .$$

Notations

Notons $\mathbb{C}_n$ l'ensemble de tous les codes de longueur n sur F_q.

l'ensemble $\mathbb{C} = \bigcup_{n \geq 1} \mathbb{C}_n$ est l'ensemble de tous les codes sur F_q .

Soit E un ensemble non vide sur lequel la notion d'égalité est définie.

Définition

Un invariant sur E est une application $v: C \rightarrow E$ telle que deux codes équivalents prennent la même valeur par v, c'est a dire :

$$\forall C \in \mathbb{C}_n \ , \ \forall \sigma \in S_n : v(\sigma(C)) = v(C).$$

Corollaire

Soit v un invariant et soit C un code de longueur n, pour tout $i \in I$ et pour toute permutation $\sigma \in S_n$; nous avons :

$$v(C_i) = v(\sigma(C)_{\sigma(i)}).$$

Définition

Une signature S sur un ensemble E est une application qui à tout code C de longueur n et à tout élément i de I, associe un élément $S(C, i)$ de E

et telle que pour toute permutation $\sigma \in S_n$ et pour tout i de I :

$$S(\sigma(C), \sigma(i)) = S(C, i).$$

Définition

Une signature S est *totalement discriminante* pour un code C de longueur n si

$S(C, i) \neq S(C, j)$ pour tout i, j distincts de I.

Cela veut dire que l'application $S_C : i \rightarrow S(C, i)$ est injective, pour C donné. Nous dirons alors que les positions $i \in I$ sont discriminées.

Proposition

Soit C un code de longueur n, S est une signature totalement discriminante pour C ; alors :

1) le groupe de permutation de C est trivial.

2) la signature S est totalement discriminante pour tout code C' équivalent à C.

Proposition

Etant donnés deux codes équivalents C et C' tels que $C' = \sigma(C)$, et une signature S totalement discriminante pour C , alors σ est bien déterminée et elle est unique.

Soient *A, B* respectivement les ensembles des valeurs $S(C, i)$ et $S(C', j)$, pour tout $(i, j) \in I \times I$;

C'est à dire :

$$A = \{ S(C, i) / i \in I \}$$

$$B = \{ S(C', j) / j \in I \}.$$

Remarque importante

La méthode ainsi présentée dans la preuve est appelée algorithme de séparation des supports dû à NICOLAS SENDRIER, il permet de calculer la permutation σ Si la signature est totalement discriminante.

Pour décider si *C* et C' sont équivalents, l'algorithme apporte la réponse si $|A| = |B| \leq n$ ([NS 96] et [NS 02]).

Rappels et notations

Dans tout ce qui suit C et C' représentent deux codes équivalents de longueur n tels que $\sigma(C) = C'$

et S une signature définie comme suite :

$$S(C, i) = v(C_i) \text{ où } v \text{ est un invariant.}$$

Supposons dans la suite que S n'est pas discriminante en s positions seulement, c'est à dire que :

$$S(C, i_1) = S(C, i_2) = \ldots = S(C, i_s) \text{ avec } i_1, i_2, \ldots I_s \in I .$$

et $2 \leq s \leq n-2$.

Partition associée à une signature

Si S est une signature et C est un code de longueur n. Soit la relation R_C définie sur I par :

$$\forall i, j \in I : i \; R_C \; j \Leftrightarrow S(C, i) = S(C, j)$$

Lemme

la relation R_C ainsi définie est une relation d'équivalence sur I et elle permet de définir une partition de I selon S et C.

Proposition

Si $\{ \bar{j}_1 , \bar{j}_2 , \ldots\ldots , \bar{j}_d \}$ est une partition de I selon C et S et si $C' = \sigma(C)$, alors la partition de I selon C' et S est :

$$\{ \overline{\sigma(j_1)}, \overline{\sigma(j_2)}, \ldots\ldots, \overline{\sigma(j_d)} \} .$$

Proposition

Si $C' = \sigma(C)$ et S n'est pas discriminante en s positions $i_1 , i_2 , \ldots , i_s$ de I pour C, alors S n'est pas discriminante en s positions $\sigma(i_1), \sigma(i_2) , \ldots , \sigma(i_s)$ de I pour C'.

Proposition

Si S est une signature non discriminante en i_1 , i_2 , …. , i_s de I pour C ; alors

{ i_1 , i_2,…, i_s } est stable par l'action du groupe de permutations de C.

Soit pour *i* fixé ; l'application φ_C de *{ i_1 , i_2 , …. , i_s }* dans *E* définie par:

$$\varphi_C(\alpha) = v(C_{\{i,\alpha\}})$$

Proposition

Si l'application φ_C est injective , alors les images de i_1 , i_2 , …. , i_s par σ peuvent être déterminés immédiatement avec $C' = \sigma(C)$.

Exemple

Considérons les deux codes *C* et C' définis par :

$$C = \{ 01101 , 01011, 01110 , 10101 , 11110 \}$$

$$C' = \{ 10101 , 00111 , 10011 , 11100, 11011 \}$$

Prenons comme signature :

$S(C, i) = v(C_i) = W(C_i)$, le polynôme énumérateur des poids de C_i pour $i = 1, 2, \ldots, 5$. Alors :

$C_1 = \{ 01101, 01011, 01110, 00101 \} \rightarrow x^2 + 3x^3$

$C_2 = \{ 00101, 00011, 00110, 10101, 10110 \} \rightarrow 3x^2 + 2x^3$

$C_3 = \{ 01001, 01011, 01010, 10001, 11010 \} \rightarrow 3x^2 + 2x^3$

$C_4 = \{ 01101, 01001, 01100, 10101, 11100 \} \rightarrow 2x^2 + 3x^3$

$C_5 = \{ 01100, 01010, 01110, 10100, 11110 \} \rightarrow 3x^2 + x^3 + x^4$

Remarquons que les positions 2 et 3 ne peut être discriminées.

Pour C' : $C'_1 = \{ 00101, 00111, 00011, 01100, 01011 \} \rightarrow 3x^2 + 2x^3$

$C'_2 = \{ 10101, 00111, 10011, 10100 \} \rightarrow x^2 + 3x^3$

$C'_3 = \{ 10001, 00011, 10011, 11000, 11011 \} \rightarrow 3x^2 + x^3 + x^4$

$C'_4 = \{ 10101, 00101, 10001, 11100, 11001 \} \rightarrow 2x^2 + 3x^3$

$C'_5 = \{ 10100, 00110, 10010, 11100, 11010 \} \rightarrow 3x^2 + 2x^3$

Remarquons que pour C', les positions 1 et 5, ne peuvent être discriminées.

D'après: $S(C, 1) = S(C', 2)$ nous tirons $\sigma(1) = 2$.

$S(C, 4) = S(C', 4)$ nous tirons $\sigma(4) = 4$.

$S(C, 5) = S(C', 3)$ nous tirons $\sigma(5) = 3$.

Soient les applications :

$\varphi_C(\alpha) = W(C_{\{1, \alpha\}})$, pour $\alpha \in \{ 2, 3 \}$.

$\varphi_{C'}(\beta) = W(C'_{\{2, \beta\}})$, pour $\beta \in \{ 1, 5 \}$.

alors : $C_{\{1,2\}} = \{ 00101, 00011, 00110 \} \rightarrow \varphi_C(2) = 3x^2$

$C_{\{1,3\}} = \{ 01001, 01011, 01010, 00001 \} \rightarrow \varphi_C(3) = x + 2x^2 + x^3$

$C'_{\{2,1\}} = \{ 00101, 00111, 00011, 00100 \} \rightarrow \varphi_{C'}(1) = x + 2x^2 + x^3$

$C'_{\{2,5\}} = \{ 10100, 00110, 10010 \} \rightarrow \varphi_{C'}(5) = 3x^2$

donc φ_C et $\varphi_{C'}$ sont injectives et de $\varphi_C(2) = \varphi_{C'}(5)$, nous tirons $\sigma(2) = 5$,

et de $\varphi_C(3) = \varphi_{C'}(1)$, nous tirons $\sigma(3) = 1$..

ainsi la permutation σ est :

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 1 & 4 & 3 \end{pmatrix}$$

Conclusion

Questions :

- L'invariant proposé par Sendrier est le polynôme énumérateur des poids est-il possible de trouver un autre invariant qui serait mieux efficace pour les codes ou au moins pour certaines classes de codes ?
- Est il possible de construire une signature qui serait totalement discriminante pour les codes ou au moins pour certaines classes de codes?
- Si le groupe de permutations d'un code C n'est pas trivial, comment rendre ce travail de détermination de la permutation de la permutation $\sigma \in S_n$; telle que $\sigma(C) = C'$ applicable ?

Enfin , nous pensons qu'on peut commencer par construire un code $L(L')$ à partir de $C(C')$ dont le groupe de permutations *perm (L)* est le groupe quotient $perm(C)/perm(C)$ qui est trivial à condition que les codes L et L' ainsi construits sont équivalents par permutation unique π ssi $C\ et C'$ sont équivalents par permutation σ ou on peut déduire σ à partir de π.

Introduction

L’étude de l’équivalence de deux codes est un problème important en théorie des codes correcteurs d’erreurs : supposons que nous ayons deux codes. Il s’agit de trouver une permutation (ou une matrice monomiale..) telle que l’image du premier code par cette permutation est le deuxième code.

Toute une classe de cryptosystèmes fondés sur la théorie des codes correcteurs d’erreurs utilisent comme clef secrète un code linéaire très structuré, pour lequel on dispose d’un algorithme de décodage rapide, et fournissent comme clef publique un code équivalent.

Outre son importance en cryptographie, l’équivalence aide à classifier les codes, en particulier, les codes auto duaux. Deux codes équivalents ont la même structure : même distance minimale, même distribution des poids, leurs groupes de permutations sont isomorphes.

INTRODUCTION GENERALE

Présentation du problème

Il est possible de définir la notion d'équivalence de deux codes de plusieurs manières :

- comme équivalence par permutation .
- comme équivalence par matrices monomiales .
- comme équivalence par matrices monomiales et automorphismes de corps de base.

Chaque définition étant plus générale que les précédentes. Dans ce travail nous nous intéresserons à deux premiers cas. S'il s'agit des permutations, nous parlerons d'équivalence par permutation, sinon d'équivalence.

L'étude de l'équivalence de deux codes est un problème important en théorie des codes correcteurs d'erreurs : supposons que nous ayons deux codes. Il s'agit de trouver une permutation (ou une matrice monomiale..) telle que l'image du premier code par cette permutation est le deuxième code.

Toute une classe de cryptosystèmes fondés sur la théorie des codes correcteurs d'erreurs utilisent comme clef secrète un code linéaire très structuré, pour lequel on dispose d'un algorithme de décodage rapide, et fournissent comme clef publique un code équivalent.
Outre son importance en cryptographie, l'équivalence aide à classifier les codes, en particulier, les codes auto duaux. Deux codes équivalents ont la même structure : même distance minimale, même distribution des poids, leurs groupes de permutations sont isomorphes.

Tout cela nous a incité à nous intéresser à la détermination de l'équivalence et de l'équivalence par permutation des codes.

Déroulement de la thèse

Dans cette thèse, nous nous intéressons à l'étude théorique des codes équivalents par permutation ainsi à la détermination de la permutation qui assure cette équivalence lorsqu'elle est unique tout en s'appuyant sur les notions d'invariants et des signatures fondées à partir des polynômes énumérateurs de poids.

Le premier chapitre est un chapitre d'introduction où nous présentons des notions et des propriétés fondamentales concernant les groupes, les corps finis et les espaces vectoriels. Nous avons étudié un peu plus en détail les groupes de permutations et les corps finis car ses notions interviennent beaucoup dans les chapitres qui suivent. Les notions citées dans ce chapitre représentent l'outil mathématique utilisé pour l'étude des codes correcteurs d'erreurs.

Le second chapitre regroupe les définitions et les propriétés fondamentales des codes correcteurs d'erreurs ; nous étudions les codes correcteurs et leurs paramètres (section 2.2). Nous nous traitons une classe particulière des codes ; à savoir les codes linéaires ainsi que leur décodage (section 2.3 ; 2.4 ; 2.5 et 2.6). La section 2.7 représente une étude sur la dualité où nous avons pu calculer le code dual d'un code auto orthogonal dans un cas particulier en montrant la proposition 2.7.6. Enfin la section 2.8 représente la définition du polynôme énumérateur qui sera utilisé comme invariant dans le chapitre quatre.

Le troisième chapitre est consacré à l'étude, aussi peu détaillée, des groupes de permutations et de l'équivalence des codes :nous étudions les définitions et les propriétés des groupes de permutations des codes et la notion d'équivalence sur deux niveaux ,à savoir équivalence par permutations et isomorphisme. Nous montrons que nous pouvons représenter le groupe d'automorphisme comme groupe de permutations et l'équivalence comme équivalence par permutations. Enfin nous étudions les propriétés de deux codes équivalents.

Enfin, dans le quatrième et dernier chapitre, nous introduisons les notions des codes poinçonnés et leurs propriétés, les notions d'invariants et signatures introduites par NICOLAS SENDRIER.

Nous étudions la détermination de la permutation de l'équivalence en considérant une signature totalement discriminante. Enfin nous discutons un cas où la signature utilisée n'est pas totalement discriminante et nous montrons que si une certaine condition est vérifiée, la permutation peut être calculé dont nous avons pu le démontrer.

Citons finalement que ce travail a pour d'origine les rapports de recherche de NICOLAS SENDRIER [NS96] et [NS02] qui a étudié le problème d'équivalence des codes en introduisant les notions d'invariants et de signatures.

DEFINITIONS ET PROPRIETES ELEMENETAIRES

1. Introduction

Ce chapitre est un chapitre de préliminaires. Il s'agit ici de présenter la terminologie est les principales notations; tout en ciblant les objets étudiés.

Les définitions et résultats énoncés constituent la base pour explorer ces objets. D'autres éléments viendront les compléter au cours des différents chapitres.

2. Groupes

Dans cette section nous rappelons les définitions et les notations usuelles de la théorie des groupes. Pour une description détaillée voir [RG 97] ; [LJG 73] ; [AD 79] .

Définition 2.1

Soit G un ensemble non vide.

Une *loi de composition interne* sur G est une application φ de $G\times G$ dans G.

Notation 2.2

L'image $\varphi(x,y)$ de $(x,y)\in G\times G$ par φ sera noté $x\varphi y$ ou xy si aucune confusion n'est à craindre.

Définition 2.3

Soit G un ensemble non vide muni d'une loi de composition interne définie par :

$$(x,y)\mapsto xy$$

G possède une *structure de groupe* (ou par abus de langage que G est un groupe) si :

(i) cette loi est associative : quelques soient les éléments x,y,z de G

$$x(yz)=(xy)z$$

(ii) G possède un élément neutre e pour cette loi : pour tout $x\in G$

$$xe=ex=x$$

(iii) tout élément $x\in G$ possède un symétrique x' (ou x^{-1}) de G, tel que :

$$xx'=x'x=e$$

Exemple 2.4

Soit E un ensemble non vide, l'ensemble $S(E)$ des bijections de E sur E, muni de la loi de composition des applications, est un groupe appelé le groupe symétrique de E. Si E est fini, de cardinal $n\geq 1$,on note S_n le groupe symétrique de E. les éléments de S_n sont appelés des permutations de E.

Remarque 2.5

1. si la loi de *G* est commutative, c'est-à-dire c'est pour tout $x, y \in G,\ xy = yx$, le groupe *G* est dit commutatif ou abélien.
2. si le cardinal de *G* est fini, *G* est dit fini et le nombre de ses éléments est appelé l'ordre de *G* , et est noté $|G|$.

Définitions 2.6

Soit *H* une partie non vide d'un groupe *G*. *H* est appelé *un sous-groupe* de *G* si :

i) $x, y \in H \Rightarrow xy \in H.$

ii) $x \in H \Rightarrow x^{-1} \in H.$

où x^{-1} désigne l'élément symétrique de *x* dans *G*.

Soit *H* un sous-groupe de *G*. Pour tout élément $x \in G$, on définit l'ensemble $Hx = \{hx / h \in H\}$ appelé la classe à gauche de x modulo *H*.

De même, on définit $xH = \{xh / h \in H\}$ la classe à droite de x modulo H.

Désignons par $(G/H)_g$ (respectivement $(G/H)_d$) l'ensemble des classes à gauche (respectivement à droite) modulo *H*.

Lemme 2.7

Soit H un sous-groupe de G ; il existe une bijection de $(G/H)_d$ *sur* $(G/H)_g$.

Preuve

Puisque pour $x, y \in G$ on a :

$$Hx = Hy \Leftrightarrow xy^{-1} \in H \Leftrightarrow (x^{-1})^{-1} y^{-1} \in H \Leftrightarrow x^{-1}H = y^{-1}H.$$

On vérifie facilement que la correspondance $Hx \mapsto x^{-1}H$ est une application bijective de $(G/H)_g$ dans $(G/H)_d$. *c.q.f.d*

Si le groupe G est fini, les ensembles $(G/H)_d$ et $(G/H)_g$ sont donc de même cardinal , ce dernier est appelé l'indice de H dans G et noté $[G:H]$.

Théorème (de Lagrange) 2.8

L'ordre et l'indice d'un sous-groupe H d'un groupe fini G sont des diviseurs de l'ordre de ce groupe et $[G:H] = \dfrac{|G|}{|H|}$.

Preuve

Soit H un sous-groupe d'un groupe G et soit $x \in G$ .l'application $y \mapsto yx$ est une bijection de H sur Hx, toutes les classes à gauche sont équipotentes à H, donc sont équipotentes entre-elles.

Pour tout $x, y \in G$; soit $Hx = Hy$ ou bien soit $Hx \cap Hy = \emptyset$.

Cela permet de conclure que les classes à gauche forment une partition de G.

Comme le cardinal de $(G / H)_g$ est $[G : H]$ nous avons :

$$|G| = |H|.[G : H] \qquad c.q.f.d$$

Groupe cyclique 2.9

Il est clair, en utilisant la définition 2.6 de voir que l'intersection d'une famille quelconque $(H_i)_{i \in I}$ de sous groupes d'un groupe G est un sous groupe de G.

Lemme 2.9.1

Soit x un élément de G, il existe un plus petit sous-groupe de G contenant x.

Preuve

G est un sous groupe de G contenant x, soit $(H_i)_{i \in I}$ la famille non vide des sous-groupe de G contenant x, et soit. $H = \bigcap_{i \in I} H_i$. C'est un sous groupe de G contenant x. Si L est un sous-groupe de G contenant x, L est l'un des H_i , donc $H \subset L$. *c.q.f.d*

Le plus petit sous-groupe d'un groupe G contenant x est appelé le sous-groupe de G engendré par x, et il est noté $gp(x)$. L'élément x est dit élément générateur de $gp(x)$.

Exemples 2.9.2

1. le sous-groupe de $(Z,+)$ engendré par $n \in N$ est l'ensemble des multiples de n dans Z.
2. le sous-groupe $gp(\bar{3})$ de $Z / 4Z$ est $\{\bar{0},\bar{3},\bar{2},\bar{1}\}$.
3. Le sous-groupe $gp(\bar{2})$ de $Z / 4Z$ est $\{\bar{0},\bar{2}\}$.
4. le sous-groupe

$$gp(\tau) \text{ de } S_3 \text{ où } \tau(1)=1, \tau(2)=3, \tau(3)=2, \text{ est } \{id, \tau\}$$

avec id est l'application identique de $\{1,2,3\}$.

Définition 2.9.3

Un groupe G est dit *cyclique* s'il existe un élément x qui sera appelé générateur tel que $G = gp(x)$.

Il est aisé de vérifier qu'un groupe *G* est cyclique de générateur *x* si, et seulement si , tout élément *y* de *G* s'écrit x^s où $s \in Z$.

C'est-à-dire que : $gp(x) = \{x^s / s \in Z\}$ avec :

$$x^s = \begin{cases} x.x.x...x & s \text{ fois si } s > 0 \\ e & \text{si } s = 0 \\ x^{-1}.x^{-1}..x^{-1}, & -s \text{ fois si } s < 0 \end{cases}$$

Pour plus de détails sur les groupes cycliques nous renvoyons à [LJG 73] ou à [RG 97] .

Le théorème suivant permet de classifier tous les groupes cycliques, nous le citons sans démonstration, car cette dernière peut être tirée de n'importe quel ouvrage traitant les groupes cycliques.

Théorème 2.9.4

Soit G un groupe cyclique.

1. *si G est infini ; G est isomorphe à Z*
2. *si G est fini d'ordre* $k \geq 1$ *,G est isomorphe à* Z / kZ .

Groupes abéliens finis 2.10

Lors de la caractérisation du groupe multiplicatif d'un corps fini,nous aurons besoin au théorème principal des groupes abéliens finis qui permet de parcourir tous les modèles des groupes abéliens finis. Pour une démonstration voir par exemple [AD79].

Théorème principal des groupes abéliens finis 2.10.1

Tout groupe abélien fini G est isomorphe à un produit direct $\prod_{i=1}^{r} H_i$ *de groupes cycliques non triviaux tels que pour* $i = 1,2,3...r$.

$$|H_i| \text{ divise } |H_{i+1}| .$$

Exemple 2.10.2

Il y a exactement quatre types de groupes abéliens finis non isomorphes d'ordre *100,* qui sont : $Z / 100Z,\ Z / 2Z \times Z / 50Z,\ Z / 5Z \times Z / 20Z,\ Z / 10Z \times Z / 10Z.$

Groupe symétrique S_n : 2.11

L'étude des codes équivalents à un code est basé sur le groupe de permutations de ce code, ce groupe est un sous-groupe du groupe symétrique d'un ensemble fini utilisé pour indexer les positions des mots du code. La section 3.2 du chapitre III concernant le groupe

d'automorphisme d'un code donné montrera ce dernier comme groupe de permutations d'un autre code. Ce résultat particulier est inspiré par le théorème de CAYLEY qui permet de représenter les groupes comme groupes de permutations.

Théorème (de CAYLEY) 2.11.1

Tout groupe G est isomorphe à un sous-groupe du groupe $(S(G),\circ)$.

Preuve

Soit $f : G \to S(G)$ définie par $f(g) = f_g$ où $f_g(x) = gx$ pour tout x de G. Pour un $g_0 \in G$, si $g \in G$, l'équation en x, $g_0.x = g$ possède une solution unique x et il en résulte que f_{g_0} est une bijection de G sur G. on vérifie facilement que f est injective. Comme pour tous $x, g, g' \in G$,

$$f_{gg'}(x) = gg'x = f_g(g'x) = f_g(f_{g'}(x)) = f_g \circ f_{g'}(x)$$

On constate que f est un isomorphisme de G sur $f(G)$, sous groupe de $(S(G),\circ)$ *c.q.f.d.*

L'ordre de S_n 2.11.2

Pour tout $n \geq 1$, le groupe symétrique S_n est d'ordre $n!$

où $n! = n \times (n-1) \times ... \times 2 \times 1$.

Pour une permutation σ de S_n, nous écrivons

$$\sigma = \begin{pmatrix} 1 & ... & k & ... & n \\ i_1 & ... & i_k & ... & i_n \end{pmatrix}$$

avec

$$\sigma(k) = i_k \text{ et } i_k \in \{1,2,...,n\} \text{ pour tout } k = 1,...,n .$$

Groupe de permutations 2.11.3

Soit G un sous-groupe de $S(E)$. On l'appelle un groupe de permutations de E.

Le cardinal $|E|$ de E est le degré de G.

Le cardinal $|G|$ de G est l'ordre de G.

Nous noterons gx l'image $g(x)$ de $x \in E$ sous l'action de la permutation $g \in G$.

Si $gx = x$, nous disons que g fixe x (ou x est fixé par g).

Nous noterons id l'élément neutre de $S(E)$ et g^{-1} le symétrique de g.

Propriété 2.11.4

$$\text{Si } g,h \in S(E), \text{alors } (gh)^{-1} = h^{-1}g^{-1}.$$

Preuve

Il suffit de vérifier que $h^{-1}g^{-1}$ est le symétrique de gh dans $S(\text{E})$. *c.q.f.d*

Proposition 2.11.5

Si G est un groupe de permutations de E , alors il en est de même pour xGx^{-1}, *pour toute permutation* x de E *avec :*

$$xGx^{-1} = \{xgx^{-1} / g \in G\}.$$

Preuve

Il suffit de montrer que xGx^{-1} est un sous groupe de $S(E)$.

Soient $\alpha, \beta \in xGx^{-1}$; alors ils existent $g_1, g_2 \in G$ tels que :

$$\alpha = xg_1x^{-1} \text{ et } \beta = xg_2x^{-1}.$$

$$\alpha\beta = (xg_1x^{-1})(xg_2x^{-1}) = x(g_1x^{-1}xg_2)x^{-1} = xg_1g_2x^{-1} \in xGx^{-1}$$

$$\alpha^{-1} = (xg_1x^{-1})^{-1} = xg_1^{-1}x^{-1} \text{ ; (par la proposition 2.11.4)}$$

Donc : $\alpha^{-1} \in xGx^{-1}$

En vertu de la définition 2.6, xGx^{-1} est un sous-groupe de $S(E)$, pour tout $x \in S(E)$.

Définition 2.11.6

Les notions sont celles de proposition 2.11.5 le groupe xGx^{-1} est appelé le groupe conjugué de *G*.

Action d'un groupe fini sur un ensemble 2.12

Dans toute cette section, *G* désigne un groupe fini et *E* un ensemble fini non vide.

Définition 2.12.1

On dit que le groupe *G* opère sur l'ensemble *E* si *G* est isomorphe à un sous groupe du groupe symétrique $S(E)$.

La définition précédente peut être facilement montrer qu'elle est équivalente à la définition suivante :

Définition 2.12.2

le groupe *G* opère sur l'ensemble *E* s'il existe une application $\Phi : G \times E \to E$

Ou $\phi(g,x)$ sera noté $g.x$, satisfaisante, pour tous $g_1, g_2 \in G$, $x \in E$ aux conditions :

i. $g_1 . (g_2 . x) = g_1 g_2 . x$

ii. $1.x = x$ ou *1* est l'élément neutre de *G*.

Exemple 2.12.3

Soit $E = \{1,2,...,n\}$. S_n opère sur *E* comme suit : pour $\sigma \in S_n$,$x \in E$; $(\sigma, x) \to \sigma(x)$

Remarque et définitions 2.12.4

Soit *G* un groupe opérant sur un ensemble *E* . la relation $\Re$ définie sur *E* par :

$$x \Re y \Leftrightarrow \exists g \in G ; y = g.x$$

est une relation d'équivalence .

La classe de $x \in E$ modulo cette relation est appelée *l'orbite* de *x* qui sera noté par $G.x$ ou $\bar{x}$ ou *orb (x)*

$$G.x = \{g.x / g \in G\} = \{y \in E / \exists g \in G ; g.x = y\}$$

L'ensemble $G_x = \{g \in G / g.x = x\}$ est un sous groupe de *G* appelé le *stabilisateur* de *x.*

3. Corps finis

Nous rappelons dans cette section , des définitions et des propriétés liées au corps finis , nous le faisons très brièvement ,pour une étude plus détaillée nous renvoyons par exemple à [RG 97] ou [MS 77] .

Caractéristique d'un corps 3.1

Soit *K* un corps, désignons par *0* (resp. *1*) l'élément neutre pour l'addition (resp. la multiplication) du corps *K* , on a un unique homomorphisme d'anneaux $\varphi : Z \to K$ définie par :

$$\varphi(n) = n.1$$

- Si φ est injectif , il identifie Z à un sous-anneau de *K* ; alors *K* contient aussi le corps des fractions *Q* de *Z* , on dit que *K* est de caractéristique nulle .
- Si φ n'est pas injectif , son noyau est un idéal pZ où $p > 0$; Z / pZ s'identifie à un sous anneau de *K*,il est donc intègre,de sorte que *p* est un nombre premier; on dit que *K* est de caractéristique *p*. dans ce cas Z/pZ est un corps qu'on note F_p .

Le sous corps Q ou F_p de *K* est le plus petit sous corps de *K* ; on l'appelle le sous corps premier de *K*.

Définition 3.2

Un corps de *q* éléments est dit un *corps fini* de cardinal *q*

Exemple 3.3

Pour *p* premier; F_p est un corps fini de cardinal *p*

Théorème 3.4

Soit K un corps fini. Posons $q = card(K)$ *alors*

i. La caractéristique de K est un nombre premier p

ii. K est un espace vectoriel de dimension finie n sur F_p *et on a* $q = p^n$

Preuve

Comme *Z* est infini, *K* ne peut être de caractéristique nulle. Donc il contient F_p avec p premier. Ainsi *K* est un espace vectoriel sur F_p, sa dimension *n* est finie, sinon *K* serait infini. En tant qu'espace vectoriel *K* est isomorphe à F_p^n ; donc K a p^n éléments. *c.q.f.d*

Théorème 3.5

Soit K un corps fini de cardinal q

Le groupe multiplicatif $(K^*,\cdot)$ *est cyclique d'ordre q-1.*

Preuve

Le groupe $(K^*,\cdot)$ est commutatif (car tout corps fini est commutatif, c'est le théorème de WEDDERBURN) et fini. D'après le théorème 2.10.1, il existent des groupes cycliques $H_1, \ldots, H_r$ tels que :

$$K^* \cong H_1 \times H_2 \times \ldots \times H_r$$

et pour tout $i = 1,2,\ldots,r-1$, $|H_i|$ divise $|H_{i+1}|$

L'entier $s = |H_r|$ est donc un exposant de chaque élément de K^*

donc pour tout $x \in K^*$, $x^s = 1$; en d'autre terme tout les éléments de K^* sont racine du polynôme $x^s - 1 \in K[x]$ or ce polynôme admet au plus *s* racines donc

$$|K^*| \leq s$$

Mais $|H_r| = s$ divise $|K^*|$,d'où $|H_r| = |K^*|$, et comme K^* est fini cela entraîne que $K^* = H_r$. *c.q.f.d*

Théorème 3.6

Soit K un corps fini de cardinal q.

Pour tout $x \in K^*$ *on a* $x^{q-1} = 1$, *et pour tout* $x \in K$ *on a* $x^q = x$.

Preuve

D'après le théorème 3.5, l'ordre de K^* est *q-1* ; donc pour tout $x \in K^*$ on a $x^{q-1} = 1$ et par conséquent pour tout $x \in K$ on a $x^q = x$. *c.q.f.d*

Il en résulte du théorème 3.6. qu'un corps fini *K à q* éléments est l'ensemble des racines du polynôme $x^q - x$.

On note souvent F_q un corps fini à *q* éléments.

Définition 3.8

Un élément générateur du groupe cyclique F_q^* d'un corps fini F_q est appelé un *élément primitif* de F_q .

Théorème 3.8

Soit α *un élément primitif d'un corps fini* F_q *; alors*

$$F_q = \{0,1,\alpha,\alpha^2,...,\alpha^{q-2}\}$$

avec $\alpha^{q-1} = 1$, *de plus* α^k *est primitif si et seulement si k et q-1 sont étrangers.*

Preuve

Elle repose sur le théorème 3.5. et sur le théorème 2.8. en tenant compte que l'ordre de α^k est $\dfrac{q\text{-}1}{p\ gcd(k,q\text{-}1)}$. *c.q.f.d*

le théorème 3.8. permet de construire un corps fini F_q , il suffit de remarquer qu'un élément primitif de F_q est une racine (*q-1*)- *ième* de l'unité .

Deux corps finis de même cardinal *q* sont, par le théorème 3.6, corps de racines du polynôme $x^q - x$ qui est unique à un isomorphisme prés,donc ils sont uniques à un isomorphisme prés,c'est-à-dire qu'ils sont isomorphes.

Construction et théorème 3.9

Soit *p* un nombre premier, comme F_p est un corps, les idéaux de l'anneau principal $F_p[x]$ sont principaux. Soit $f \in F_p[x]$ un polynôme de degré $n > 0$ et soit (*f*) l'idéal principal engendré par *f*.

Soit $g + (f)$ un élément arbitraire de l'anneau quotient $F_p[x]/(f)$.

Par la division euclidienne, ils existent $h, r \in F_p[x]$ tels que :

$g = hf + r$ avec *deg r < deg f = n*, puisque $hf \in (f)$; il vient que $g + (f) = r + (f)$,

c.à.d. que tout élément de l'anneau $F_p[x]/(f)$ s'écrit d'une façon unique sous la forme :

$$a_0 + a_1 x + ... + a_{n-1} x^{n-1} + (f);\ a_i \in F_p$$

Si nous identifions F_p au sous anneau $\{a + (f) / a \in F_p\}$, tout élément de $F_p[x]/(f)$ peut s'écrire sous la forme :

$$a_0 + a_1(x + f) + ... + a_{n-1}(x + f)^{n-1}$$

Posons $x + (f) = \alpha$, nous pouvons donc écrire tout élément de $F_p[x]/(f)$ sous la forme unique $a_0 + a_1\alpha + ... + a_{n-1}\alpha^{n-1}$, et nous pouvons considérer $F_p[x]/(f)$ comme un espace vectoriel sur F_p dont une base est $\{1, \alpha, \alpha^2, ..., \alpha^{n-1}\}$

Comme $0 + (f) \in F_p[x]/(f)$, nous avons $\bar{f}(\alpha) = f + (f) = 0 + (f) = 0$

C'est-à-dire que α est une racine de f dans $F_p[x]/(f)$.

Donc nous pouvons énoncer le théorème suivant :

Théorème

Soit $f \in F_p[x]$ de degrés n>0. Alors $F_p[x]/(f)$ est un espace vectoriel de dimension n sur F_p de base $\{1, \alpha, \alpha^2, ..., \alpha^{n-1}\}$ avec $\alpha = x + (f)$

De plus $F_p[x]/(f)$ est un corps si et seulement si f est irréductible.

Exemple d' un cops fini 3.10

Le polynôme $f(x) = x^2 + x + 1$ est irréductible sur le corps fini $F_2 = \{0,1\}$ de degrés *2*. Donc $F_2[x]/(x^2 + x + 1)$ est un corps fini de cardinal $q = 2^2 = 4$, dont ses éléments peuvent être représentés sous forme $a + b\alpha$, avec *a*,b appartiennent *à* F_2 et *x* est satisfait à $\bar{f}(\alpha) = 0$. c'est-à-dire $\alpha^2 + \alpha + 1 = 0$ *ou* $\alpha^2 = \alpha + 1$.

Donc $F_4 = F_2[x]/(x^2 + x + 1) = \{0, 1, \alpha, 1 + \alpha\}$.

Par exemple $\alpha(1+\alpha) = \alpha + \alpha^2 = \alpha + \alpha + 1 = 1$

4. Espaces vectoriels

Dans cette section nous rappelons quelques définitions nécessaires de l'Algèbre linéaire , nous citons certaines propriétés qui seront utiles pour définir les codes linéaires à travers le chapitre II.

Notations 4.1

Soit V un espace vectoriel sur un corps K .

V^n désigne l'ensemble des n-uplets $(v_1, v_2, ..., v_n)$ avec $v_i \in V$

$dim_k V$ (ou $dimV$) désigne la dimension de V sur K.

A^t désigne la transposée d'une matrice A.

Proposition 4.2

Soit K un corps, l'ensemble K^n muni de l'addition définie par :

$$(x_1, x_2, ..., x_n) + (y_1, y_2, ..., y_n) = (x_1 + y_1, x_2 + y_2, ..., x_n + y_n)$$

et la multiplication par scalaire $\lambda \in K$:

$\lambda(x_1, x_2, ..., x_n) = (\lambda x_1, \lambda x_2, ..., \lambda x_n)$ est un espace vectoriel de dimension n sur K

Sous espace vectoriel 4.3

Soit W un sous ensemble d' un espace vectoriel V sur un corps K . W est un sous espace de V si et seulement si pour tout $\mu, v \in W, \lambda \in K$. $\mu v \in W$ et $\lambda\mu \in W$.

Exemples 4.4

- Soit $V = F_2^2$, $W = \{(a,0) / a \in F_2\}$ est un sous espace de V
- $W = \{v = (x_1, x_2, ..., x_n) \in F_q^n / x_1 + x_2 + ... + x_n = 0\}$ est un sous espace sur F_q

Définitions 4-5

Soit $f : V \to W$ une application linéaire et soit A la matrice associée à f .

$ker\ f = ker\ A = \{v \in V / f(v) = 0\}$ est le noyau de f ou l'espace nul de la matrice A .

$f(V) = A.V = \{f(v) / v \in V\} = \{A.v / v \in V\}$ est l'espace image de F (ou de la matrice A .)

On rappelle que *Ker f* (resp. *f (V)*) est un sous espace de V (resp. de W) .

Théorème 4-6

V et W sont des espaces vectoriels sur K de dimensions finies .

Soit $T : V \to W$ une application linéaire Alors

$$dim\ V = dim\ Ker\ T + dim\ T\ (V)$$

Preuve

En effet il est facile de remarquer que $V / \ker T$ est isomorphe à $T(V)$ en tant qu'espaces vectoriels .

Théorème 4.7

Soit V un espace vectoriel de dimension finie n = dim V sur un corps K Alors

i) V est isomorphe à K^n .

ii) Si K est un corps fini de cardinal q , V est fini de cardinal q^n .

Preuve

Pour *i)* , il suffit de considérer l'isomorphisme

$$V \to K^n$$
$$\alpha_1 e_1 + ... + \alpha_n e_n \mapsto (\alpha_1, \alpha_2, ..., \alpha_n)$$

où $(e_1, e_2, ..., e_n)$ est une base de *V* sur *K*.

Pour *ii)* , il découle de *i).* *c.q.f.d*

CODES CORRECTEURS D' ERREURS

1. Introduction

Pour communiquer des informations on les codes au moyen de chiffres, de lettres, de sons,…,etc. Malheureusement quand une transmission se prolonge des erreurs finissent toujours par se produire. Afin de corriger une erreur qui vient d'être détectée on pourrait recommencer la transmission, mais ce n'est pas toujours possible car le message original peut être perdu (satellite qui se déplace, résultat d'un calcul qui n'est pas enregistré,…,etc.) ou bien le temps peut manquer (suivi d'un phénomène physique en temps réel.). en outre il ne faut pas oublier qu'une nouvelle transmission risque d'introduire de nouvelles erreurs. C'est pourquoi,lorsqu'une erreur est détectée, il vaut mieux chercher à la corriger directement plutôt que de chercher à retransmettre le message. La théorie des codes vise les deux buts : la détection et la correction des erreurs.

2. les codes

Soit Q un ensemble non vide à q éléments, Q sera appelé un alphabet.

Soient k et n deux entiers naturels non nuls avec $k \leq n$ ..

Un $k-uplets$ a de Q^k sera appelé un message ou un mot de longueur k, et sera noté soit : $a = (a_1, a_2, ..., a_k)$ soit : $a = a_1 a_2 ... a_k$.

L'ensemble des messages sera une partie E de Q^k .

La technique du codage par bloc consiste à associer à chaque mot $a = (a_1, a_2, ..., a_k) \in E$,un mot plus long, c'est-à-dire un mot de Q^n , de façon unique. On introduit ainsi une application injective :

$$f : \quad E \to Q^n$$
$$a = (a_1, a_2, ..., a_k) \mapsto c = (c_1, c_2, ..., c_n)$$

appelée application de codage ou encodeur. Le message $a \in E$ est modifié pour fournir le mot $c = f(a) \in Q^n$. C'est le mot c qui sera transmis et lu par un système quelconque pour donner un message reçu $x = (x_1, x_2, ..., x_n)$ qui contient éventuellement quelques erreurs.

Notons $C = f(E)$ l'image de f. Comme f est injective, f réalise une bijection de E sur C et C peut être considéré comme l'ensemble de tous les messages possibles.

C est appelé code de longueur n sur Q , et les éléments de C s'appellent les mots du code. Le cardinal du code est par définition celui de C.

Exemples 2.1

i) $C_1 = \{133,311,111\}$ est un code de longueur *3* sur $Q_1 = \{1,3\}$.

ii) $C_2 = \{aaaa,bcde,xtlm,aabb\}$ est un code de longueur *4* et de cardinal *4* sur *Q* l'alphabet de la langue française.

iii) $C_3 = \{1\alpha,\alpha\alpha,0\alpha,11,\alpha 0\}$ est un code de longueur *2* et de cardinal *5* sur le corps fini F_4 construit en I.3.10.

Distance de Hamming 2.2

Pour compter le nombre d'erreurs, on introduit la distance de Hamming sur Q^n. elle permet de mesurer le degré de différence entre deux mots *x* et *y* de Q^n.

Définition 2.2.1

La *distance de Hamming* entre deux mots $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$, que l'on notera $d(x,y)$,est le nombre d'indices *i* tels que $x_i \neq y_i$.

C'est-à-dire :

$$d(x,y) = |\{i / x_i \neq y_i\}|.$$

Exemples 2.2.2

i) nous avons $d(133,111) = 2$ et $d(311,111) = 1$ pour l'exemple 2.1.*i).*

ii) $d(aaaa,aabb) = 2$ et $d(bcde,xtlm) = 4$ pour l'exemple 2.1.*ii).*

iii) $d(0\alpha,\alpha 0) = 2$ pour l'exemple 2.1.*iii*).

Proposition 2.2.3

La distance de Hamming est une métrique sur Q^n *, pour lequel* Q^n *est appelé espace de Hamming.*

Preuve

Au lieu de montrer la proposition de façon directe, nous donnons une preuve basée sur les lemmes suivants qui sont faciles à montrer :

Lemme 1

Soit E un ensemble non vide et soit $\delta : E \times E \to R^+$ *une application telle que :*

- $\delta(x,y) = 0 \Leftrightarrow x = y$.
- $\delta(x,y) = 1$ *pour tout x et y de E tels que* $x \neq y$.

Alors δ *est une distance sur E.*

Lemme 2

Soient $(E_1,d_1),(E_2,d_2),...,(E_n,d_n)$, *n espaces métriques. Alors* $E = E_1 \times E_2 \times ... \times E_n$ *est un espace métrique pour E muni de la distance* $d = d_1 + d_2 + ... + d_n$ *définie par :*

$$d(x,y) = d_1(x_1,y_1) + d_2(x_2,y_2) + ... + d_n(x_n,y_n)$$

avec $x = (x_1,x_2,...,x_n)$ *et* $y = (y_1,y_2,...,y_n)$ *de E.*

Si nous posons dans le deuxième lemme :

$E_1 = E_2 = ... = E_n = Q$ et $d_1 = d_2 = ... = d_n = \delta$, Q^n muni de $d = d_1 + d_2 + ... + d_n$ est un espace métrique avec :

$$\begin{aligned} d(x,y) &= \delta(x_1,y_1) + \delta(x_2,y_2) + ... + \delta(x_n,y_n) \\ &= \left|\{i / x_i \neq y_i\}\right| \end{aligned}$$

c.q.f.d

Distance minimale d'un code 2.2.4

La distance minimale d'un code *C* est la distance minimum entre deux mots distincts de ce code. On la note *d* :

$$d = Min\{d(x,y) / x,y \in C \text{ et } x \neq y\}.$$

Un code *C* de longueur *n*, de cardinal *M* et de distance minimale *d* est appelé un code $[n,M,d]$. Les nombres n,M,d sont les paramètres du code.

La distance minimale *d* permet d'obtenir le nombre maximum d'erreurs que le code peut corriger.

Si le mot $c = (c_1,c_2,...,c_n)$ a été envoyé avec moins de *t* erreurs de transmission, le mot obtenu $x = (x_1,x_2,...,x_n)$ vérifie $d(x,c) \leq t$ .ainsi l'on peut retrouver *c* à partir de *x*, si et seulement si, il existe un unique mot de code situé à une distance de $x \leq t$. Cela revient à dire que les boules fermées de rayon *t* centrées sur les éléments du code *C* soient disjointes. Un code corrigera *t* erreurs si cette condition est vérifiée.

Théorème 2.2.5

Un code C de distance minimale d corrige au plus $e = \left[\frac{d-1}{2}\right]$ *erreurs et en détecte* $d-1$.

Preuve

Le code *C* ne corrige pas *t* erreurs si et seulement si :

$$\exists x \in Q^n, \exists c, c' \in C, c \neq c', d(x,c) \leq t \text{ et } d(x,c') \leq t \quad (1)$$

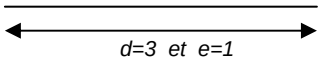
et cela entraîne

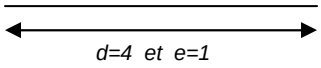
$$d \leq d(c,c') \leq d(c,x) + d(x,c') \leq 2t \text{ ,soit } d \leq 2t \quad (2)$$

La réciproque est vraie, en effet ,si (2) est vérifiée , on peut toujours trouver deux mots de code c et c' situés à la distance d l'un de l'autre,et les noter $c = (c_1, c_2, ..., c_n)$ et $c' = (c'_1, c'_2, ..., c'_d, c'_{d+1}, ..., c'_n)$ quitte à permuter les coordonnées. Il existe deux entiers naturels p et q inférieurs ou égaux à t tels que $d = p + q \leq 2t$, et le mot $x = (c'_1, c'_2, ..., c'_p, c_{p+1}, ..., c_{p+q}, c_{d+1}, ..., c_n)$ vérifie bien $d(x,c) = p \leq t$ et $d(x,c') = q \leq t$ cela prouve (1).

En conclusion *C* corrige *t* erreurs si et seulement si $2t \leq d$, et cela équivaut à $t \leq \left[\frac{d-1}{2}\right]$.

c.q.f.d





Exemples 2.2.6

1) Dans F_{13}^2, le code $C = (54,29,91)$ est de distance minimale *2* , ce code détecte une erreur sans pouvoir la corriger.

2) Dans F_{13}^2, la droite vectorielle *5x-3y = 0* détermine un code C' , sa distance minimale est *2*. il est facile de savoir si un mot appartient à C' ,et d'exhiber l'encodage :

$$f : F_{13} \to F_{13}^2$$

$$t \to (3t, 5t)$$

Le poids de Hamming 2.2.7

Le poids de Hamming d'un mot $x = (x_1, x_2, ..., x_n)$ de F_q^n, noté $\omega(x)$, est le nombre d'indices i tels que $x_i \neq 0$.

$$\omega(x) = |\{i / x_i \neq 0\}|.$$

Nous pouvons remarquer que le poids $\omega(x) = d(x,0)$.

Par exemple, dans F_2^4, nous avons $\omega(1101) = 3$ et $\omega(0011) = 2$.

Théorème (Borne de Hamming) 2.2.8

Soit C un code sur F_q, de longueur n et de cardinal M

Si le code C corrige t erreurs alors :

$$M(1 + (q-1)C_n^1 + ... + (q-1)^t C_n^t) \leq q^n .$$

Preuve

Dans F_q^n, il y a $(q-1)^m C_n^m$ mots de longueur *n* et de poids *m*. les boules fermées de rayons *t* et de centres dans *C* sont disjointes. Chaque boule fermée,parmi les *M* boules, contient $1 + (q-1)C_n^1 + ... + (q-1)^t C_n^t$ mots. Le nombre total des mots de F_q^n est q^n , d'où le théorème. *c.q.f.d*

Remarque

Si l'égalité a lieu, le code *C* est dit parfait.

Théorème (Borne de Singleton) 2.2.9

Si C un $[n,M,d]$ code sur F_q, alors $M \leq q^{n-d+1}$.

Preuve

Si *C* est un $[n,M,d]$ code,alors la restriction à *C* de la projection $\pi : F_q^n \to F_q^{n-d+1}$ qui efface les $d-1$ derniers symboles du code, c'est-à-dire $(x_1, x_2, ..., x_n) \mapsto (x_1, x_2, ..., x_{n-d+1})$, est encore injection car *d* est minimale, alors $|C| \leq q^{n-d+1}$.

c.q.f.d

Pour pouvoir travailler avec les codes, il faut mettre plus de structure.

3. Les codes linéaires

si q est puissance d'un nombre premier, d'après I.3.6. et I.3.8. il existe à un isomorphisme prés un unique corps fini F_q de cardinal q .

choisissons $E = F_q^k$ comme ensemble de message. L'ensemble *E* devient maintenant un espace vectoriel de dimension *k* sur F_q, et il est naturel de ne considérer que les fonctions d'encodage f linéaires. Le code $C = f\left(F_q^k\right)$ est alors structuré en sous-espace vectoriel de F_q^n .

Définition 3.1

Un *code linéaire* de dimension *k* et de longueur *n* sur F_q est un sous-espace vectoriel de dimension *k* de F_q^n .

Si la distance minimale de *C* est *d*, on dit que *C* est un code de paramètres $[n,k,d]$, et si *q=2* ; le code *C* est dit *code binaire.*

Pour un code linéaire *C*, on retrouve la distance de Hamming par la formule $d(x,y)=\omega(x-y)$, et la distance minimale du code *C* par

$$d=min\{\omega(x)/\ x\in C \text{ et } x\neq 0\}.$$

Matrice génératrice 3.2

L'application linéaire $f: F_q^k \to F_q^n$ possède une matrice que l'on notera ${}^tG$ dans les bases canoniques. Ici l'écriture ${}^tG$ désigne la transposée de la matrice *G* qui possède *k* lignes et *n* colonnes, et tout mot de *C* s'écrira sous la forme :

$$c=f(x)=xG\ .$$

Où $c=(c_1,c_2,...,c_n)\in F_q^n$ et $x=(x_1,x_2,...,x_k)\in F_q^k$ sont des vecteurs lignes.

Définition 3.2.1

Une matrice génératrice du code *C* est une matrice *G* de taille $k\times n$ telle que :

$$C=\{c\in F_q^n\ /\ \exists x\in F_q^k\ ; c=xG\}.$$

Remarque 3.2.2

Puisqu'on peut se donner un sous-espace vectoriel (et donc un code linéaire) par une base,une matrice génératrice d'un code linéaire *C* est une matrice dont les lignes forment une base de *C*.

Exemple 3.2.3

Soit *G* la matrice génératrice du [5,2] code binaire *C* telle que :

$$G=\begin{pmatrix}0&1&1&1&1\\1&0&0&1&0\end{pmatrix}$$

Déterminions *C* :

$$C=\{c_1(0,1,1,1,1)+c_2(1,0,0,1,0)/\ c_1,c_2\in F_2\}$$

$$C=\{00000,01111,10010,11101\}$$

Ainsi le code *C* est de paramètres $[5,2,2]$ et, par exemple,le message *11* est codé par $c=11\,G=11101$.

Matrice de contrôle 3.3

On peut aussi se donner un sous-espace vectoriel par un système d'équations indépendantes. Une matrice de contrôle d'un code linéaire *C* est la matrice d'un système d'équations linéaires homogènes indépendantes dont l'espace des solutions est *C*. autrement dit :

Définition 3.3.1

Une matrice de contrôle *H* d'un code linéaire *C* est une matrice de taille $(n-k)\times n$ et de rang $(n-k)$ vérifiant : $C=\{c\in F_q^n / H\ {}^t c=0\}$.

On tire de la définition, que $C=KerH$ et $rgH=n-k$ avec *k* la dimension de *C* sur F_q.

Exemple 3.3.2

Supposons $q=2,n=6,k=3$ (donc $M=2^3=8$).

C le code linéaire de paramètres $[6,3]$ dont la matrice de contrôle est donnée par :

$$H=\begin{bmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\end{bmatrix}$$

Comme $C=KerH$

$$c=(c_1,c_2,c_3,c_4,c_5,c_6)\in H\Leftrightarrow H\ {}^t c=0\ \Leftrightarrow\begin{cases}c_1+c_2+c_4=0\\c_1+c_3+c_5=0\\c_2+c_3+c_6=0\end{cases}$$

Si le message $a=011$ est transmis,alors le mot de code correspond est $c=011110$.

Le code *C* contient 2^3 mots de code :

000000,001011,010101,011110,100110,101101,110011,111000.

Définition(syndrome d'un code) 3.3.3

Toute application linéaire $S: F_q^n\to F_q^m$ où $n\in N$ de noyau *C* est appelé syndrome de *C*.

L' application $S(x)=H\ {}^t x$ définit un syndrome de *C*.

Principe de décodage par syndrome 3.3.4

Supposons que $c\in C$ est le mot de code envoyé, et $r\in F_q^n$ le mot reçu. La différence $e=r-c$ est le vecteur d'erreur. Son poids $\omega(e)$ est le nombre de bits erronés dans le mot reçu.

Soit *H* la matrice de contrôle de *C* et soit le syndrome *s* du mot reçu *r* défini par :

$$^{t}s = H\ ^{t}r = H\ ^{t}e .$$

Le syndrome est nul si et seulement si $r \in C$. Le syndrome *S* définit un isomorphisme du quotient F_q^n / C sur F_q^{n-k}. Si le syndrome est non nul ,on corrige le mot reçu *r* , en appliquant le principe du maximum de vraisemblance : on soustrait à *r* un mot de poids minimum dans sa classe modulo *C,* c'est-à-dire un mot de poids minimum parmi ceux ayant même syndrome que *r*. dans le cas où $\omega(e)$ est strictement inférieur à $d/2$, alors e est l'unique mot de poids minimum dans la classe de *r* modulo *C* et on récupère bien le mot de code émis.

Exemple 3.3.5

Soit le code *C* de l'exemple 3.3.2.

Le tableau suivant est construit de la manière suivante :

La première ligne regroupe les représentants de poids minimums,des classes de F_2^6 modulo *C*. la deuxième ligne regroupe les syndromes associés avec $S(x)=H^{t}x.$.

000000	*100000*	*010000*	*001000*	*000100*	*000010*	*000001*	*001001*
000	*011*	*101*	*110*	*100*	*010*	*001*	*111*

Si $r = 101001$ est le mot reçu,on calcule son syndrome $S(r) = H\ ^{t}r = 100$, par le principe cité plus haut,le mot envoyé *c* est dans la classe de $e = 000100$ modulo *C,*

c'est-à-dire : $c = r\text{-}e = 101101$.

4. Codes systématiques

A chaque mot $x = (x_1, x_2, ..., x_k)$ du message on adjoint $n-k$ symboles $c_{k+1}, ..., c_n$ dépendant linéairement des x_i pour obtenir le mot de code $c = f(x)$.

Les symboles ajoutés sont appelé bits de contrôle.

On a :

$$c = (x_1, x_2, ..., x_k, c_{k+1}, ..., c_n) = (x_1, ..., x_k)(I_k / A).$$

Où (I_k / A) désigne la matrice $k \times n$ obtenue en écrivant cote à cote la matrice identique I_k de taille *k* et une matrice quelconque A .

Définition 4.1

Un code C sera dit systématique s'il possède une matrice génératrice de la forme $G = (I_k / A)$

Comme $c \in C \Leftrightarrow \exists x \in F_q^k ; \mathrm{c} = xG = x(I_k/\mathrm{A})$ on aura :

$$c \in C \Rightarrow (-{}^tA / I_{n-k})\ {}^tc = (-{}^tA / I_{n-k}) \begin{pmatrix} I_k \\ {}^tA \end{pmatrix} {}^tx = -{}^tA\,{}^tx + {}^tA\,{}^tx = 0 .$$

Autrement dit C est inclus dans le noyau de l'application linéaire de matrice $H = (-{}^tA / I_{n-k})$.

On notera $C \subset Ker\ H$.

Comme H est une matrice de rang $n - k$, on aura $dim\ C = dim\ Ker\ H = k$ et $C = Ker\ H$. On vient donc de montrer que la matrice H est une matrice de contrôle de C.

Exemples de codes binaires systématiques 4.2

a) L'encodage $f(x_1, x_2, x_3) = (x_1, x_2, x_3, x_1 + x_3, x_2 + x_3, x_1 + x_2 + x_3, x_3)$ définit un code systématique C de paramètres [7,3] sur F_2.

L'écriture :

$$(c_1, c_2, ..., c_7) = (x_1, x_2, x_3) \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

met en évidence une matrice génératrice de C, et l'on déduit la matrice de contrôle

$$H = \left(\begin{array}{ccc|cccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

b) la matrice génératrice G du code de parité binaire sera :

c)

$$G = \left(\begin{array}{ccc|c} 1 & & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & 1 & 1 \end{array}\right)$$

Puisque l'encodage est :

$$(x_1,x_2,...,x_k)\mapsto(x_1,x_2,...,x_k)\left(\begin{array}{ccc|c} 1 & & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & 1 & 1 \end{array}\right)=(x_1,x_2,...,x_k,\sum_{i=1}^{k}x_i)$$

la matrice du contrôle *H* associée est : $H=(1,1,...,1)$.

d) code de répétition :

répétons n fois le symbole x_1 , on obtient un code [*n,1*] pour lequel

$$G=(1,1,...,1)\ et\ H=\left(\begin{array}{c|cccc} 1 & 1 & 0 & . & 0 \\ . & 0 & 1 & 0 & 0 \\ . & . & . & . & . \\ . & . & . & . & 0 \\ 1 & 0 & . & 0 & 1 \end{array}\right)$$

5. Codes MDS

Théorème 5.1

La distance minimale d'un code linéaire de matrice de contrôle H est égale au nombre minimum de colonnes de H linéairement dépendantes . autrement dit :

$$d=min\{s\in N^*/\exists\ s\ \ colonnes\ de\ H\ linéairements\ dépendantes\}.$$

Preuve

Soit *C* un code de paramètres [*n, k, d*] et de matrice de contrôle $H=(h_1,h_2,...,h_n)$ où h_i désigne la *i-ème* colonne de *H* . le théorème provient des équivalences :

$$x=(x_1,x_2,...,x_n)\in C \Leftrightarrow H\ {}^t x=0 \Leftrightarrow (h_1,h_2,...,h_n)\ {}^t(x_1,x_2,...,x_n)=0$$
$$\Leftrightarrow \sum_{i=1}^{n}x_i h_i=0.$$

Si *x* représente un mot de code de poids *d*, et la relation $\sum_{i=1}^{n}x_i h_i=0$ montre une relation de dépendance d'exactement *d* colonnes de *H*, il existera donc *d* colonnes de *H* linéairement dépendantes. D'autre part, si les *s* colonnes $h_{i_1},h_{i_2},...,h_{i_s}$ *de H* sont linéairement dépendantes, il existe une *s-liste* $(x_{i_1},x_{i_2},...,x_{i_s})\neq(0,0,...,0)$ telle que $\sum_{j=1}^{s}x_{i_j}h_{i_j}=0.$ En posant $x_i=0$ si $i\notin\{i_1,i_2,...,i_s\}$ et $x=(x_1,x_2,...,x_n)$,

on constate que $x\in C$ et $\omega(x)\leq s$ cela entraîne $d\leq s$. *c.q.f.d*

corollaire 5.2

Les paramètres d'un code linéaire vérifient toujours l'inégalité

$$k + d \leq n + 1.$$

Preuve

La matrice *H* est de rang *n - k*, donc $n - k + 1$ de ses colonnes seront toujours liées et le théorème II.5.1. entraîne $d \leq n - k + 1.$ *c.q.f.d*

Définition 5.3

un code **MDS** (*Maximum Distance Separable*) est un code [*n, k, d*] tel que $d = n - k + 1.$

Corollaire 5.4

Le code C est un code MDS si,et seulement si, n - k colonnes quelconques d'une de ses matrices de contrôle H sont toujours linéairement indépendantes.

Exemples 5.5

a) Le sous-espace vectoriel *C* engendré par un vecteur $a \in F_q^n$ dont toutes les coordonnées sont non nulles est un code MDS de paramètres [*n, 1, n*].

b) L'hyperplan d'équation $x_1 + x_2 + ... + x_n = 0$ est un code MDS de paramètres [*n, n-1, 2*].

6. Codes cycliques

Définition 6.1

Un code linéaire *C* est *cyclique* s'il est stable par permutation à droite de ses composantes,c'est-à-dire s'il vérifie :

$$(a_0,...,a_{n-2},a_{n-1}) \in C \Rightarrow (a_{n-1},a_0,...,a_{n-2}) \in C.$$

Notons $F_q[x]$ l'algèbre des polynômes à coefficients dans le corps fini F_q , *et* $(x^n - 1)$ l'idéal engendré par le polynôme $x^n - 1$.

L'algèbre quotient $F_q[x]/(x^n - 1)$ est un F_q -espace vectoriel de dimension *n*, dont une base est $\{\dot{1}, \dot{x}, ..., \dot{x}^{n-1}\}$ .Cela nous permet d'identifier vectoriellement un élément $a = (a_0,...,a_{n-2},a_{n-1})$ *de* F_q^n au polynôme $a(x) = a_0 + a_1\dot{x} + ... + a_{n-1}\dot{x}^{n-1}$ de $F_q[x]/(x^n - 1)$.

Pour simplifier les notations, on oubliera de mettre le point au-dessus de la classe de *x*, de sorte que l'on écrira :

$$a(x) = a_0 + a_1 x + ... + a_{n-1} x^{n-1} \in F_q[x]/(x^n - 1).$$

le polynôme associé au vecteur $a = (a_0,...,a_{n-2},a_{n-1}) \in F_q^n$.

Avec cette identification le code *C* devient un sous-espace vectoriel de $F_q[x]/(x^n - 1)$.

Si *C* est cyclique :

$$a(x) = a_0 + a_1 x + ... + a_{n-1} x^{n-1} \in C \Rightarrow x \cdot a(x) = a_{n-1} + a_0 x + a_1 x^2 + ... + a_{n-2} x^{n-1} \in C$$

et de proche en proche, on obtient $x^i a(x) \in C$ pour tout entier i. Par linéarité le polynôme $b(x)a(x)$ appartiendra à *C* quel que soit le polynôme $b(x)$ de $F_q[x]/(x^n - 1)$, et *C* sera un idéal de l'anneau quotient $F_q[x]/(x^n - 1)$.

Réciproquement, tout idéal de $F_q[x]/(x^n - 1)$ est un sous-espace vectoriel stable par permutation à droite de coordonnées dans la base $\{1, x, ..., x^{n-1}\}$ et l'on peut annoncer :

Théorème 6.2

Un code cyclique est un idéal de $F_q[x]/(x^n - 1)$ *dés que l'on identifie les* F_q*-espaces vectoriels* F_q^n *et* $F_q[x]/(x^n - 1)$.

On peut démontrer le théorème suivant qui précise la nature des idéaux de $F_q[x]/(x^n - 1)$. (voir par exemple I.3.9. ou [RG97]).

Théorème 6.3

Soit K un corps et $f(x) \in K[x]$.

L'anneau $A = K[x]/(f(x))$ *est principal, et tout idéal de A est de la forme* $(g(x))$ *où g(x) est un polynôme unitaire de* $K[x]$ *qui divise* $f(x)$.

De plus un tel polynôme g(x) est unique.

Définition 6.4

Soit *C* un code cyclique. L'unique polynôme unitaire $g(x) \in F_q[x]$ qui divise $x^n - 1$ et tel que $C = (g(x))$ est appelé polynôme générateur du code cyclique *C*.

Soit *C* le code cyclique de longueur *n sur* F_q de polynôme générateur $g(x)$, écrivons $x^n - 1 = g(x)h(x)$ et posons degrés de $g(x) = n\text{-}k$. Le code *C* est l'image de l'application linéaire :

$$\gamma : F_q^{\,k} \rightarrow F_q^{\,n} \cong F_q[x]/(x^n - 1)$$
$$a = (a_0,...,a_{k-1}) \mapsto a(x) \cdot g(x)$$

Puisque tout élément de *C* est un multiple de $g(x)$ (dans l'identification des espaces vectoriels $F_q^{\,n}$ et $F_q[x]/(x^n - 1)$ et comme le degrés de $a(x)\,g(x)$ *est* $\leq n\text{-}1$, l'application γ sera injective et réalisera un encodage simple de *C*. On en déduit aussi :

Théorème 6.4

La dimension du code cyclique de longueur n sur F_q *et de polynôme générateur* $g(x)$ *est n - deg* $g(x)$.

Notons : $g(x) = g_0 + g_1 x + ... + g_{n-k} x^{n-k}$ et $a(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1}$

Alors

$$a(x)g(x) = a_0 g_0 + (a_0 g_1 + a_1 g_0)x + ... + a_{k-1} g_{n-k} x^{n-1}$$

et $\gamma(a) = (a_0, a_1, ..., a_k)G$ avec la matrice génératrice :

$$G = \begin{pmatrix} g_0 & g_1 & . & . & . & g_{n-k} & 0 & . & . & . & 0 \\ 0 & g_0 & & & & & . & & & & 0 \\ . & & . & & & & & . & & & . \\ . & & & . & & & & & . & & . \\ . & & & & . & & & & & . & 0 \\ 0 & . & . & . & . & g_0 & & & & & g_{n-k} \end{pmatrix}$$

Soit $h(x) = x^n - 1 / g(x)$.

L'application $S : a \mapsto S(a) = a(x)h(x)$ définit le syndrome de *C* puisque :

$$a(x) \in C \Leftrightarrow a(x)h(x) = 0$$

en effet , $a(x) \in C$ entraîne $a(x) = b(x)g(x)$ d'où :

$$a(x)h(x) = b(x)g(x)h(x) = 0.$$

Réciproquement, $a(x)h(x) = 0$ se traduit par :

$(x^n - 1)/a(x)h(x)$ qui entraîne $g(x) / a(x)$ et $a(x) \in C$. on déduit alors les équivalences :

$$a(x) \in C \Leftrightarrow (a_0 + a_1 x + ... + a_{n-1} x^{n-1})(h_0 + h_1 x + ... + h_k x^k) = 0$$
$$\Leftrightarrow H\ {}^t a = 0.$$

Avec :

$$H = \begin{pmatrix} h_0 & 0 & . & . & . & 0 & h_k & . & . & . & h_2 & h_1 \\ h_1 & h_0 & & & & & & & & & & h_2 \\ h_2 & . & . & & & & & & & & & . \\ . & & . & . & . & & & & . & . & & . \\ . & & & . & . & . & & & & . & . & . \\ . & & & & . & . & . & & & & . & h_k \\ h_k & & & & & . & . & . & & & & 0 \\ 0 & . & & & & & . & . & . & & & . \\ . & & . & & & & & . & . & . & & . \\ . & & & . & & & & & . & . & . & . \\ . & & & & . & & & & & h_1 & h_0 & 0 \\ 0 & & & & & h_k & . & . & . & . & h_1 & h_0 \end{pmatrix}$$

On a trouvé une matrice de contrôle H de C de taille $n \times n$.

Exemples 6.5

a) Supposons que les messages à transmettre sont les éléments de F_2^3.

Soit $g(x) = x^3 - 1 = 1 + x^3$ il divise $x^3 - 1$, $g(x)$ engendre un code cyclique sur F_2.

Tous les messages sont encodés de la manière suivante :

$$000 \mapsto 000000 \qquad 100 \mapsto 100100$$
$$001 \mapsto 001001 \qquad 101 \mapsto 101101$$
$$010 \mapsto 010010 \qquad 110 \mapsto 110110$$
$$011 \mapsto 011011 \qquad 111 \mapsto 111111$$

La matrice génératrice correspondante est la matrice G avec :

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

b) soit $g(x) = x^3 + x + 1 \in F_2[x]$ *et* $n = 7$ alors $h(x) = \dfrac{x^7 - 1}{g(x)}$

c'est-à-dire $h(x) = x^4 + x^2 + x + 1$

pour le code cyclique de polynôme générateur $g(x)$ on a :

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

7. Dualité

Notons $x = (x_1, x_2, ..., x_n)$ *et* $y = (y_1, y_2, ..., y_n)$ deux mots *de* F_q^n avec $n \geq 1$.

Définition 7.1

le produit scalaire euclidien sur F_q^n est la forme bilinéaire symétrique qui à tout *x* et *y* de F_q^n associé l'élément $< x, y >= \sum_{i=1}^{n} x_i y_i$ de F_q.

Par exemple, le produit scalaire euclidien sur F_2^4 de $x = 1101$ et $y = 1111$ est :

$$< x,y > = 1.1 \ + \ 1.1 \ + 0.1 \ + \ 1.1 \ = 1$$

Définitions 7.2

1) Deux mots x, y de F_q^n sont dits orthogonaux si $< x, y > = 0$.

2) Soit *C* un code linéaire de longueur *n*, le dual ou l'orthogonal de *C* est l'ensemble :

$$C^{\perp} = \{ y \in F_q^n \ ; \forall \ x \in C \ ; \ < x,y > = 0 \}$$

Pour un code linéaire *C*, le dual $C^{\perp}$ est aussi un code linéaire sur F_q. Si de plus *C* est de matrice génératrice *G* et de matrice de contrôle *H*, alors $C^{\perp}$ serait de matrice génératrice *H* et de matrice de contrôle *G*, et cela provient de la définition de l'orthogonalité.

L'orthogonalité permet de déduire que : $G\ {}^tH = H\ {}^tG = 0$.

Enfin remarquons que si *C* est un code linéaire $[n,k]$ alors $C^{\perp}$ est un $[n, n-k]$ car l'égalité suivante : $dim C + dim C^{\perp} = n$ est vraie. nous citons quelques propriétés de dualité sans démonstration car cette dernière est facile.

Propriétés 7.3

Soit C_1 et C_2 *deux codes , alors :*

i) $(C_1^{\perp})^{\perp} = C_1$

ii) $(C_1 + C_2)^{\perp} = C^{\perp}{}_1 \cap C^{\perp}{}_2$ avec $C_1 + C_2 = \{c_1 + c_2 / c_1 \in C_1 et\, c_2 \in C_2\}$

il peut arriver que pour un code *C* , le dual $C^{\perp}$ contient *C*.

Si $C \subset C^{\perp}$, alors *C* est appelé un code auto-orthogonal (en anglais *weakly self dual*) et si $C^{\perp} = C$ le code *C* est appelé un code auto dual (en anglais *stricly self dual*). Pour ces deux cas nous pouvons démontrer la proposition suivante :

Proposition 7.4

i) *Un code C est auto-orthogonal si, et seulement si,* $< x, y >= 0$. *pour tout* $x, y \in C$.

ii) *Un code C est auto-dual si, et seulement si, il est auto-orthogonal et de dimension* $k = \frac{n}{2}$ *(donc n doit être pair).*

Exemple 7.5

Le code binaire [7,3] de matrice génératrice

$$G = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

est un code auto orthogonal, il suffit de voir que les vecteurs lignes de *G* sont de poids pairs et sont orthogonaux deux à deux.

Lors du déroulement de ce travail, nous avons pu montrer la proposition suivante en utilisant un argument d'espace quotient, qui calcule le dual d'un code auto orthogonal particulier :

Proposition 7.6

Si n est impair, C est un code $\left[n, \frac{1}{2}(n-1)\right]$ *auto orthogonal sur* F_2 *; alors*

$C^{\perp} = C \cup (\mathbf{1} + C)$ *où* $\mathbf{1}$ désigne le mot 11...1.

Preuve

définissons la relation $\Re$ sur $C^{\perp}$ par :

$$\forall\, x,y \in C^{\perp}\ ;\ x \,\Re\, y \ \Leftrightarrow (x\text{-}y) \in C.$$

La relation $\Re$ est une relation d'équivalence sur $C^{\perp}$, qui n'est autre que la relation modulo C ; c'est-à-dire :

$$C^{\perp} / \Re = C^{\perp} / C = \{x + C / x \in C^{\perp}\}.$$

C et $C^{\perp}$ sont des espaces vectoriels sur F_2, donc des groupes pour l'addition des vecteurs de F_2^n où C est un sous-groupe de $C^{\perp}$. D'après le théorème de Lagrange I.2.8. nous avons :

$$[C^{\perp}:C] = |C^{\perp}/C| = \frac{|C^{\perp}|}{|C|}$$

Or

$$|C| = 2^{\frac{1}{2}(n-1)}, \quad \text{et} \qquad |C^{\perp}| = 2^{n-\frac{1}{2}(n-1)}$$

Alors

$$[C^{\perp}:C] = \frac{2^{n-\frac{1}{2}(n-1)}}{2^{\frac{1}{2}(n-1)}} = 2$$

C'est-à-dire que

$$C^{\perp}/C = \{C, a + C\} \qquad \text{avec} \quad a \notin C.$$

Montrons à présent que $\mathbf{1} \notin C$ où $\mathbf{1}$ est comme dans la proposition citée ci-dessus :

Nous avons : $\mathbf{1}.1 = 1 + 1 + ... + 1 = n \cdot 1$ n'est pas divisible par 2, car n est impair par hypothèse. Donc $\mathbf{1} \notin C$.

Nous obtenons donc $C^{\perp}/C = \{C, \mathbf{1} + C\}$ qui forme une partition de $C^{\perp}$ ce qui entraîne que $C^{\perp} = C \cup (\mathbf{1} + C)$. *c.q.f.d.*

8. Polynôme énumérateur des poids

Le polynôme énumérateur des poids d'un code représente un invariant de ce code,qui permet d'étudier des propriétés concernant les poids des mots du code. Il sera présent pour l'étude du chapitre quatre.

Définitions 8.1

1) La distribution de poids d'un code C de longueur n sur un corps fini F_q est la suite $A_0, A_1, ..., A_n$ où chaque A_i ; $0 \le i \le n$, est le nombre de mots dans C de poids i .

2) Le polynôme énumérateur $W_C(x,y)$ de C est le polynômede $Z[x,y]$ défini par *:*

$$W_C(x,y) = \sum_{i=0}^{n} A_i x^{n-1} y^i.$$

Remarques et exemple 8.2

a) Le polynôme énumérateur $W_C(x,y)$ est un polynôme homogène de degré *n* ;

En effet, pour $t \in Z$; $W_C(tx,ty) = t^n \; W_C(x,y)$.

b) Le polynôme énumérateur $W_C(x,y)$ peut être défini d'une autre manière ; à savoir

$$W_C(x,y) = \sum_{u \in C} x^{n-\omega(u)} \cdot y^{\omega(u)}.$$

b) Si nous remplaçons *x* par *1*, nous obtiendrons aussi un polynôme énumérateurs $W_C(y)$ avec :

$$W_C(y) = \sum_{i=0}^{n} A_i y^i.$$

qui est le plus utilisé.

c) Considérons le code $C = \{000,001,101,110\}$ de longueur *3* sur F_2

le dual $C^\perp$ de C est $C^\perp = \{000,111\}$ et les polynômes énumérateurs sont respectivement :

$$W_C(x,y) = x^3 + 3xy^2$$
$$W_{C^\perp}(x,y) = x^3 + y^3.$$

e) Le code $\{00,11\}$ de longueur *2* sur F_2 , noté *D*, est auto-dual, $D^\perp = D$ et

$$W_D(x,y) = W_{D^\perp}(x,y) = x^2 + y^2.$$

Comment déduire $W_{C^\perp}$ à partir de W_C ? le théorème de MAC WILLIAMS apporte la réponse. Afin de démontrer ce théorème, nous avons besoin de définir certaines notions et d'énoncer quelques lemmes.

Définition 8.3

Soit *p* la caractéristique de F_q et soit α un élément primitif de F_q telle que $\{1,\alpha,\alpha^2,...,\alpha^{s-1}\}$ soit une base de F_q vu comme espace vectoriel sur F_p *(avec* $q = p^s$ *).*

On définit alors, pour tout $\{\beta_0,\beta_1,...,\beta_{s-1}\}$ *de* F_q ; l'application χ_β allant de F_q dans le corps des nombres complexes C par :

$\forall \; \gamma = (\gamma_0,\gamma_1,...,\gamma_{s-1}) \in F_q$

$$\chi_\beta(\gamma) = \omega^{\beta_0\gamma_0+...+\beta_{s-1}\gamma_{s-1}} \quad \text{où :} \; \omega = e^{\frac{2i\pi}{p}}.$$

On remarque que χ_β est un morphisme du groupe additif F_q dans le groupe multiplicatif ***U*** des nombres complexes de module un.

Lemme 8.4

Pour tout β non nul du corps F_q de cardinal $q = p^s$, le morphisme χ_β vérifie :

$$\sum_{\gamma \in F_q} \chi_\beta(\gamma) = 0.$$

Preuve

On a ,en effet, les équations suivantes :

$$\begin{aligned}\sum_{\gamma \in F_q} \chi_\beta(\gamma) &= \sum_{\gamma \in F_q} \prod_{i=0}^{s-1} \omega^{\beta_i \gamma_i} \\ &= \sum_{\gamma_0=1}^{p-1} \cdots \sum_{\gamma_{s-1}=0}^{p-1} \sum_{i=0}^{s-1} \omega^{\beta_i \gamma_i} \\ &= \prod_{i=0}^{s-1} \sum_{\gamma_i=0}^{p-1} \omega^{\beta_i \gamma_i}.\end{aligned}$$

Comme β est non nul, il existe un entier r tel que β_r est non nul

On a alors :

$$\sum_{\gamma_r=0}^{p-1} \omega^{\beta_r \gamma_r} = \sum_{j=0}^{p-1} \omega^j = \frac{1-\omega^p}{1-\omega} = 0. \qquad c.q.f.d$$

Soit un entier $n \geq 1$, posons pour tout u et v de F_q^n

$\chi_u(v) = \chi_1(< u, v >)$ où *1* désigne l'élément neutre de F_q et $< u, v >$ est le produit scalaire considéré sur F_q^n.

Remarquons que pour tout $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_{s-1}) \in F_q$: $\chi_1(\gamma) = \omega^{\gamma_0}$.

Lemme 8.5

*Soit f une fonction définie sur F_q^n ; $n \geq 1$ à valeurs dans un anneau commutatif contenant **U**. la transformée de Hadamard $\hat{f}$ de la fonction f donnée par :*

$$\forall\, u \in F_q^{\,n}\ ;\ \hat{f}(u) = \sum_{v \in F_q^n} \chi_u(v) f(v)$$

Si C est un code linéaire de longueur n sur F_q *on a :*

$$\sum_{u \in C^\perp} f(x) = \frac{1}{|C|} \sum_{u \in C} \hat{f}(u).$$

Preuve

$$\begin{aligned}
\sum_{u \in C} \hat{f}(u) &= \sum_{u \in C} \sum_{v \in F_q^n} \chi_u(v) f(v) \\
&= \sum_{u \in C} \sum_{v \in F_q^n} \chi_1(<u,v>) f(v) \\
&= \sum_{v \in F_q^n} \sum_{u \in C} \chi_1(<u,v>) f(v) \\
&= \sum_{v \in C^\perp} f(v) \left(\sum_{u \in C} \chi_1(<u,v>) + \sum_{v \notin C^\perp} f(v) \left(\sum_{u \in C} \chi_1(<u,v>) \right) \right) \\
&= |C| \cdot \sum_{v \in C^\perp} f(v) + \sum_{v \notin C^\perp} f(v) \sum_{u \in C} \chi_1(<u,v>).
\end{aligned}$$

Si $v \notin C^\perp$, cela signifie qu'il existe un $u_1 \in C$ tel que : $<v,u_1> = 1$, il existe donc une partition $C_0, C_1, ..., C_{q-1}$ du code C telle que : pour tout z dans C_0 on a :

$<v,z> = 0$ et pour tout $i \geq 1$ et pour tout z dans C_i on a : $<v,z> = \alpha^i$ (α étant un élément primitif de F_q). Il est aisé de voir que ces ensembles sont en bijection avec C_0,

On en déduit donc que :

$$\sum_{u \in C} \hat{f}(u) = |C| \cdot \sum_{v \in C^\perp} f(v) + |C_0| \sum_{v \notin C^\perp} f(v) \sum_{\gamma \in F_q^n} \chi_1(\gamma).$$

Or d'après le lemme 8.4 ; cette dernière équation se réduit simplement à :

$$\sum_{u \in C} \hat{f}(u) = |C| \cdot \sum_{v \in C^\perp} f(v).$$ *c.q.f.d*

Théorème 8.6 : (*Identité de* MAC WILLIAMS) **:**

Soit C un code de longueur n sur un corps F_q *de cardinal q.*

On a alors l'équation suivante :

$$W_{C^\perp}(x,y) = \frac{1}{|C|} W_C(x+(q-1)y, x-y).$$

Preuve

Soit l'application f définie pour tout u de F_q^n par :

$$f(u) = x^{n-\omega(u)} y^{\omega(u)}$$

La transformée de Hadamard de *f* s'écrit donc :

$$\forall\, u \in F_q^n \;\; ; \;\; \hat{f}(u) = \sum_{v \in F_q^n} \chi_u(v) f(v).$$

$$= \sum_{v \in F_q^n} \chi_1(< u,v >)\, x^{n-\omega(u)}\, y^{\omega(u)}$$

On définit l'application δ de F_q dans $\{0,1\}$ qui pour tout $\gamma \in F_q$ associe *1* si γ est non nul et *0* sinon.

On obtient alors, en appliquant le lemme 8.4 que

$$\forall\, u \in F_q^n \;\; ; \hat{f}(u) = (x+(q-1)y)^{n-\omega(u)} \cdot (x-y)^{\omega(u)}$$

Et en appliquant le lemme 8.5. on en déduit les égalités suivantes :

$$W_{C^\perp}(x,y) = \sum_{c \in C^\perp} x^{n-\omega(c)} \cdot y^{\omega(c)}$$

$$= \frac{1}{|C|} \sum_{c \in C} (x+(q-1)y)^{n-\omega(c)} \cdot (x-y)^{\omega(c)}$$

$$= \frac{1}{|C|} W_C(x+(q-1)y, x-y).$$

c.q.f.d

Exemples 8.7

Appliquons le théorème 8.6. aux exemples : 9.2.*d*) et 9.2.*e)* :

Pour l'exemple 9.2. *d)* :

$$W_C(x,y) = x^3 + 3xy^2$$

$$W_{C^\perp}(x,y) = \frac{1}{4} W_C(x+y, x-y)$$

$$= \frac{1}{4}\left[(x+y)^3 + 3(x+y)(x-y)^2\right]$$

$$= x^3 + y^3.$$

dont il permet de calculer $W_C(x,y)$ une autre fois :

$$W_C(x,y) = W_{(C^\perp)^\perp}(x,y)$$

$$= \frac{1}{2} W_{C^\perp}(x+y, x-y)$$

$$= \frac{1}{2}\left[(x+y)^3 + (x-y)^3\right]$$

$$= x^3 + 3xy^2.$$

Pour l'exemple 9.2.*e)* :

$$W_D(x,y) = x^2 + y^2$$

Donc

$$W_{D^\perp}(x,y) = \frac{1}{2} W_D(x+y, x-y)$$

$$= \frac{1}{2}\left[(x+y)^2 + (x-y)^2\right]$$

$$= x^2 + y^2.$$

Qui est correcte puisque *D* est auto-dual.

On déduit facilement du théorème 9.6. le corollaire suivant :

Corollaire 8.8

Soit C un code auto-dual sur le corps fini F_q *. Son énumérateur des poids* $W_C(x,y)$ *vérifie l'équation suivante :*

$$W_C(x,y) = \frac{1}{|C|} W_C(x + (q-1)y, x-y).$$

GROUPES DE PERMUTAIONS ET EQUIVALENCE DES CODES

1. Introduction

La notion d'équivalence entre deux codes est liée à la notion de groupe de permutation .elle peut être définie en plusieurs niveaux. Supposons que nous ayons deux codes. Il s'agit de trouver une permutation (ou une matrice monomiale...) telle que l'image du premier code par cette permutation est le deuxième code. S'il s'agit de permutations nous dirons que les code sont équivalents par permutation, sinon nous dirons simplement équivalents ou isomorphes.

Toute une classe de cryptosystèmes fondé sur la théorie des codes correcteurs d'erreurs à émergé, dès l'apparition, des premiers systèmes à clef publique les systèmes de chiffrement proposés par R.J MCELIECE et H .NIERDERREITER. (voir pour plus de détail [PL01] et [HC95]) utilisent comme clef secrète un code linéaire très structuré, par lequel on dispose d'un algorithme de décodage rapide, et fournissent comme clef publique un code équivalent.
D'ailleurs l'apparition d'un algorithme efficace pour trouver une permutation entre deux codes équivalents par permutation a permis de «casser» dans certains cas les systèmes de chiffrement de MCELIECE et NIERDERREITER.

Outre son importance en cryptographie, l'équivalence aide à classifier les codes , en particulier auto-duaux.

Deux codes équivalents ont la même distance minimale, même distribution des poids, leurs groupes de permutations (ou d'automorphisme) sont isomorphes.
Tout cela nous a incité à nous intéresser à la détermination de l'équivalence de code .

2. Groupe de permutations et d'automorphismes des codes

Dans cette section nous définissons le groupe de permutations et le groupe d'automorphismes d'un code . Ces groupe, ainsi définis, ont la propriété de préserver la distance de Hamming : ce sont des groupes d'isométrie .

Groupe de permutations d'un code 2.1

Soient n un entier positif non nul et q une puissance d'un nombre premier soit $\boldsymbol{I}$ un ensemble ordonné de cardinal n utilisé pour indexer les cordonnées des mots de F_q^n (dans la suite nous prenons $\boldsymbol{I} = \{1,2,...n\}$)

une permutation $\sigma \in S_n$ agit sur les mots de F_q^n comme suit :

si $c=(c_i)_{i\in I}$ est un mot de F_q^n, alors :

$$\sigma(c)=(c_{\sigma(i)})_{i\in I}=(c_{\sigma(1)},c_{\sigma(2)},...,c_{\sigma(n)})$$

Nous voyons facilement que :

si $\sigma_1,\sigma_2\in S_n$; alors $\sigma_1\sigma_2(c)=\sigma_1(\sigma_2(c))$ pour tout $c\in F_q^n$.

En effet :

$$\begin{aligned}\sigma_1(\sigma_2(c))&=\sigma_1((c_{\sigma_2(i)})_{i\in I})\\&=(c_{\sigma_1(\sigma_2(i))})_{i\in I}\\&=(c_{\sigma_1\sigma_2(i)})_{i\in I}\\&=\sigma_1\sigma_2(c)\end{aligned}$$

D'autre part , si *id* désigne l'identité dans S_n, nous avons pour tout $c\in F_q^n$:

$$id(c)=(c_{id(i)})_{i\in I}=(c_i)_{i\in I}=c$$

donc on peut conclure de ce qui précède, que le groupe symétrique S_n opère sur F_q^n par l'opération :

$$(\sigma,c)\mapsto\sigma(c).$$

Soit maintenant *C* un code de longueur n sur F_q . La permutation σ de S_n définit une action sur le code *C* comme suit :

$$(\sigma,C)\mapsto\sigma(C)$$

avec $\sigma(C)=\{\sigma(c)/\,c\in C\}$.

Comme ce qui précède, nous avons :

$$\forall\sigma_1,\sigma_2\in S_n\ :\ \sigma_1(\sigma_2(C))=\sigma_1\sigma_2(C)$$

et $id(C)=C$

car :

$$\begin{aligned}\sigma_1(\sigma_2(C))&=\sigma_1(\{\sigma_2(c)/\,c\in C\})\\&=\{\sigma_1(\sigma_2(c))/\,c\in C\}\\&=\{\,\sigma_1\sigma_2(c)/\,c\in C\,\}\\&=\sigma_1\sigma_2(C)\ .\end{aligned}$$

et

$$\begin{aligned} id(C) &= \{id(c) / c \in C\} \\ &= \{c / c \in C\} \\ &= C . \end{aligned}$$

Ce qui permet de dire que le group S_n opère aussi sur l'ensemble des codes de longueur *n* sur F_q par :

$$(\sigma, C) \mapsto \sigma(C).$$

Notation 2.1.1

Soit C un code de longueur *n* sur F_q **.** Notons $perm(C)$ le sous ensemble de tous les éléments σ de S_n ;tels que $\sigma(C) = C$. i .e :

$$perm(C) = \{\sigma \in S_n / \sigma(C) = C\}.$$

Proposition 2.1.2

L'ensemble $perm(C)$*, muni du produit usuel des permutations, est un sous groupe de* S_n *.*

Preuve

D'après la définition d'un sous groupe (I.2.6), il suffit de montrer que $\sigma_1\sigma_2 \in perm(C)$ et $\sigma_1^{-1} \in perm(C)$ si $\sigma_1, \sigma_2 \in perm(C)$.

Comme $\sigma_1, \sigma_2 \in perm(C)$, nous avons :

$$\begin{aligned} \sigma_1\sigma_2(C) &= \sigma_1(\sigma_2(C)) \\ &= \sigma_1(C) && \text{car } \sigma_2 \in perm(C) \\ &= C && \text{car } \sigma_1 \in perm(C) \end{aligned}$$

Donc, $\sigma_1\sigma_2 \in perm(C)$.

Comme $\sigma_1 \in perm(C)$, alors $\sigma_1(C) = C$, ce qui entraîne que :

$$\begin{aligned} \sigma_1^{-1}(\sigma_1(C)) = \sigma_1^{-1}(C) &\Leftrightarrow (\sigma_1^{-1}\sigma)(C) = \sigma^{-1}(C) \\ &\Leftrightarrow id(C) = \sigma^{-1}(C) \\ &\Leftrightarrow C = \sigma^{-1}(C) \end{aligned}$$

Ce qui veut dire que $\sigma_1^{-1} \in perm(C)$.

Définition 2.1.3

Le sous groupe $perm(C)$ de S_n est appelé le groupe de permutations du code C .

Exemples 2.1.4

a) Le groupe de permutations d' un code de répétition est S_n tout entier .

En effet si $c = (x, x,\ldots x)$ est un mot de ce code avec $x \in F_q$, pour toute permutations $\sigma \in s_n$, $\sigma(c) = c$.

b) Le groupe de permutation du code $\{0000,0011,1100,1111\}$ sur F_2 est composé des *8* permutations suivantes :

id , (12), (34), $(12)(34)$, $(13)(24)$, $(14)(23)$, (1324), (1423)

où la notation (ij) désigne une transposition, et $(ijkl)$ désigne un cycle d'ordre *4* .

c) Soit C le code $[3,2]$défini sur F_2 par sa matrice génératrice G avec :

$$G = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

c'est a dire que :

$$C = \{\ x(0,1,1) + y(1,1,0)\ \} \text{ où } x, y \in F_2 .$$

$$C = \{000,011,110,101\}.$$

S_3 désigne le groupe symétrique de degrés *3*, il est donc d'ordre *3! = 6* .

à savoir :

$$S_3 = \{id,(12),(13),(23),(123),(132)\}.$$

Pour tout permutation $\sigma \in S_3$, déterminons $\sigma(C)$:

$$id\,(C) = C .$$

$$(12)(C) = \{000,101,110,011\} = C$$

$$(13)(C) = \{000,110,011,101\} = C.$$

$$(23)(C) = \{000,011,101,110\} = C$$

$$(123)(C) = \{000,101,011,110\} = C$$

$$(132)(C) = C.$$

Cela permet de conclure que $perm(C) = S_3$.

d) soit L le code $[3,1]$ sur F_2, de matrice génératrice G'

avec :

$$G' = (101)$$

Alors

$$L = \{000,101\}$$

Déterminons $\sigma(L)$, pour tout $\sigma \in S_3$:

$$id(L) = L$$
$$(12)(L) = \{000,011\} = L_1$$
$$(13)(L) = \{000,101\} = L$$
$$(23)(L) = \{000,110\} = L_2$$
$$(123)(L) = \{000,011\} = L_1$$
$$(132)(L) = \{000,110\} = L_2$$

Donc $perm(L) = \{id,(13)\}$.

e) Par définition d'un code cyclique, un code cyclique est invariant par permutation à droite σ où :

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & . & . & . & n-1 & n \\ n & 1 & 2 & . & . & . & n-2 & n-1 \end{pmatrix}$$

donc le groupe de permutations d'un code cyclique est non trivial, c'est à dire non réduit au groupe trivial $\{id\}$, car il contient le sous groupe $gp(\sigma)$engendré par la permutation à droite σ.

Proposition 2.1.5

Un code linéaire et son dual ont le même groupe de permutations .

Preuve

Soit *C* un code linéaire de longueur *n*.

Nous désirons prouver que $perm(C) = perm(C^{\perp})$,

c'est à dire :

$$\sigma \in perm(C) \Leftrightarrow \sigma \in perm(C^{\perp})$$

où encore :

$$\forall \sigma \in S_n, \sigma(C) = C \Leftrightarrow \sigma(C^{\perp}) = C^{\perp}$$

Pour montrer l'équivalence , il suffit de montrer l'implication :

$$\forall \sigma \in S_n \; ; \; \sigma(C) = C \Rightarrow \sigma(C^{\perp}) = C^{\perp}$$

car l'implication réciproque en résulte , en effet :

$$\sigma(C^{\perp}) = C^{\perp} \Rightarrow \sigma\left((C^{\perp})^{\perp}\right) = (C^{\perp})^{\perp}$$
$$\Rightarrow \sigma(C) = C$$

Soit $\sigma \in S_n$ telle que $\sigma(C) = C$, pour $x \in C^{\perp}$ nous avons :

$$< x,c >= \sum_{i=1}^{n} x_i c_i = 0 \quad \text{pour tout } c = (c_i)_{i \in I} \text{ de } C .$$

posons pour tout $i \in I$, $\sigma(i) = j_i$ cela revient à écrire $i = \sigma^{-1}(j_i)$

$$< \sigma(x),c > = \sum_{i=1}^{n} x_{\sigma(i)} c_i$$
$$= \sum_{i=1}^{n} x_{j_i} c_{\sigma^{-1}(j_i)}$$
$$=< x,\sigma^{-1}(c) >= 0$$

car $\sigma^{-1} \in perm(C)$ *et* $x \in C^{\perp}$

ce qui vient d'être montré prouve que si $x \in C^{\perp}$ alors $\sigma(x) \in C^{\perp}$ pour tout $x \in C^{\perp}$. C'est à dire

$$\sigma(C^{\perp}) \subset C^{\perp} \text{ pour tout } \sigma \in perm(C) \qquad (1)$$

comme $\sigma^{-1} \in perm(C)$ et d'après ce qui vient d'être montré on a :

$$\sigma^{-1}(C^{\perp}) \subset C^{\perp}$$

cela entraîne que :

$$\sigma(\sigma^{-1}(C^{\perp})) \subset \sigma(C^{\perp})$$

ou encore

$$C^{\perp} \subset \sigma(C^{\perp}) \qquad (2)$$

D'après (1) et (2) nous pouvons conclure que $\sigma(C^{\perp}) = C^{\perp}$ *c.q.f.d*

Il est possible de définir des notions concernant d'autre groupe opérant sur F_q^n et définir ainsi autres notions concernant les codes sur F_q , par exemple ,la notion de groupe d'automorphismes d'un code citée par plusieurs auteurs [par exemple dans MS 77] .

Dans cette section nous définissons le groupe d'automorphismes d'un code comme groupe de matrices monomiales et nous montrons qu'il est possible de remplacer le calcul du groupe d'automorphismes d'un code par celui du groupe de permutations de son code étalé, et de la même manière ; il est possible de remplacer la détermination de l'équivalence de deux codes par celle de l'équivalence par permutations de leurs codes étalés .

2.2 Groupe d'automorphismes d'un code

Permutations monomiales 2.2.1

Définition (*matrice de permutations*) **2.2.1.1**

Une *matrice de permutation* est une matrice $n \times n$ inversible à coefficients dans $\{0,1\} \subset F_q$ ayant un et un seul élément non nul par ligne et par colonne .

Exemples 2.2.1.2

a) La matrice :

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

est une matrice de permutation sur F_3 .

b) la matrice :

$$\begin{pmatrix} 1 & 0 & . & . & 0 & 0 \\ 0 & . & & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & 0 \\ 0 & . & . & . & 0 & 1 \end{pmatrix}$$

est une matrice de permutation sur F_q

Si $c \in F_q^n$ et P est une matrice de permutation , alors le produit cP donne un mot de F_q^n qui est égal en fait à c avec des coordonnées permutées , c'est la raison pour laquelle on appelle les matrices de cette sorte matrices de permutation .

Par exemple , $si = (1,2,0)$ de F_3^3 et P la matrice de permutation de l'exemple *a*) nous aurons :

$$c.P = (1,2,0)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = (1,0,2)$$

Définition (*matrice monomiale*) **2.2.1.3**

Une matrice monomiale est une matrice $n \times n$ inversible à coefficients dans F_q ayant un et un seul élément non nul par ligne et par colonne .

Exemple 2.2.1.4

a) la matrice :

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

est une matrice monomiale sur F_7 .

b) la matrice :

$$\begin{pmatrix} 0 & \alpha \\ 1 & 0 \end{pmatrix}$$

est une matrice monomiale sur F_4 .

Si $q = 2$, toutes les matrices monomiales sont des matrices de permutation , c'est plus intéressant dons le cas $q \neq 2$, on montre facilement qu'une matrice monomiale *M* est le produit d'une matrice diagonale inversible *D* et une matrice de permutation *P* , où *P* est obtenue à partir de *M*, en remplaçant les scalaires non nuls par des *1* et le $i-ème$ terme sur le diagonale de *D* est égal au terme non nul de la $i-ème$ ligne de *M* . cela se voit tout de suite si on remarque que si $D = (d_{ij}),\ P = (P_{ij})$ alors $DP = \left(\sum_{k=1}^{n} d_{ik} P_{kj} \right)_{ij} = (d_{ii} P_{ij})_{ij}$ car $d_{ij} = 0$, si $i \neq j$.

Soit $c \in F_q^n$. Multiplier *c* par *M* (à droite) revient à multiplier *c* par *D* et par *P*. i.e. multiplier chaque composant c_i de *c* par un élément de F_q^* (plus précisément , par le $i-ème$ terme d_{ii} sur la diagonale de *D*, et permuter les composantes des mots de F_q^n résultant par la permutation correspondante à *P* .

Il n'est pas difficile de montrer que le produit de deux matrices monomiales est aussi une matrice monomiale , et que l'ensemble des matrices monomiales muni du produit de matrices forme un groupe appelé groupe monomial.

Nous pouvons alors définir le groupe d'automorphismes d'un code *C* comme étant le sous groupe de tout les éléments M du groupe monomial laissant *C* globalement invariant, i.e. tel que $CM = C$ où M agit sur *C* comme suit : $CM = \{cM / c \in C\}$.

Mais traiter les matrices est plus difficile que les permutations, c'est pourquoi nous définirons le groupe d'automorphismes de *C* de façon différente, comme le sous groupe d'un certain groupe de permutations, même si au fond c'est la même chose exprimée en langage mathématique de deux manières différentes .

Comment d'écrire l'action de *M* sur $c \in F_q^n$ comme l'action d'une permutation ?

L'idée est la suivante. D'abord nous définirons les permutations monomiales, nous montrerons qu'elles forment un groupe, nous les « décomposerons »d'une certaines manière, nous donnerons quelques propriétés et nous montrerons comment on peut défini leur action sur les mots de F_q^n.

Définition 2.2.1.5

Notons $\Omega_q = F_q^* \times I$ ou $I = \{ 1,2,...,n \}$.

Une permutation monomiale de *I* est une permutation σ de Ω_q (qui agit donc sur les couples $(a,\gamma), a \in F_q^*, \gamma \in I_n$ telle que :

$$\sigma(ab,\gamma) = a(\sigma(b,\gamma)) \text{ pour tout les } a,b \in F_q^* \text{ et } \gamma \in I$$

ou le produit $c(d,\alpha)$ est de fini par :

$$c(d,\alpha) = (cd,\alpha) \text{ pour tout les } c,d \in F_q^*, \alpha \in I$$

Notons $S_q(I_n)$ l'ensemble des permutations monomiale de *I* (c'est un sous ensemble de $S(\Omega_q)$)

Proposition 2.2.1.6

$S_q(I)$ avec le produit usuel des permutations est un groupe

Preuve

Comme $S_q(I)$ contient l'identité (il n'est pas vide) , il suffit de montrer que

$$\sigma_1\sigma_2 \in S_q(I) \text{ et } \sigma_1^{-1} \in S_q(I) \text{ si } \sigma_1 \text{ et } \sigma_2 \in S_q(I)$$

nous avons .

$$\sigma_1\sigma_2(ab,\gamma) = \sigma_1(\sigma_2(ab,\gamma)) = \sigma_1(a(\sigma_2(b,\gamma)))$$

Soit $(d,\delta) = \sigma_2(b,\gamma)$, alors :

$$\sigma_1\sigma_2(ab,\gamma) = \sigma_1(a(d,\delta)) = \sigma_1(ad,\delta) = a\sigma_1(d,\delta) = a\sigma_1(\sigma_2(b,\gamma))$$
$$= a\sigma_1\sigma_2(b,\gamma)$$

et donc $\sigma_1\sigma_2 \in S_q(I)$.

continuons, soit $\sigma_1^{-1}(ab,\gamma) = (d,\delta)$, alors :

$$(ab,\gamma) = \sigma_1(d,\delta) \text{ et } \sigma_1(d,\delta) = \sigma_1(aa^{-1}d,\delta) = a\sigma_1(a^{-1}d,\delta)$$

d'où

$$(ab,\gamma) = a\sigma_1(a^{-1}d,\delta) \text{et } a^{-1}(ab,\gamma) = \sigma_1(a^{-1}d,\delta)$$

comme

$$a^{-1}(ab,\gamma) = a^{-1}a(b,\gamma) = (b,\gamma)$$

nous obtenons

$$(b,\gamma) = \sigma_1(a^{-1}d,\delta)$$

et donc

$$\sigma_1^{-1}(ab,\gamma) = a\sigma_1^{-1}(b,\gamma).$$

Ce qui montre que $\sigma_1^{-1} \in S_q(I)$. *c.q.f.d*

Proposition 2.2.1.7

Soit σ une permutation monomiale de I. Alors il existe une fonction $j:\ I \to F_q^*$ et une permutation τ de I *telles que $\sigma(1,\gamma) = (j(\gamma),\tau(\gamma))$ pour tout $\gamma \in I$.*

Preuve

Comme σ est une permutation de $\Omega_q = F_q^* \times I$, à chaque couple $(1,\gamma),\sigma$ fait correspondre une autre couple (d,δ), i.e. $\sigma(1,\gamma) = (d,\delta)$.

c'est comme ça nous définissons *j* : l'image de $\gamma \in I$ par j est le d correspondant .

Soient maintenant $(d_1,\delta_1) = \sigma(1,\gamma_1)$ et $(d_2,\delta_2) = \sigma(1,\gamma_2)$ ou $\gamma_1 \neq \gamma_2$.

Supposons que $\delta_1 = \delta_2$, alors

$$\sigma(d_1^{-1},\gamma_1) = d_1^{-1}\sigma(1,\gamma_1) = d_1^{-1}(d_1,\delta_1) = (1,\delta_1)$$

et

$$\sigma(d_2^{-1},\gamma_2) = d_2^{-1}\sigma(1,\gamma_2) = d_2^{-1}(d_2,\delta_2) = (1,\delta_2)$$

mais $\delta_1 = \delta_2$, donc :

$$(1,\delta_1) = (1,\delta_2) \text{ et } \sigma(d_1^{-1},\gamma_1) = \sigma(d_2^{-1},\gamma_2)$$

Comme σ est une permutation de Ω_q, elle est injective , d'ou $(d_1^{-1},\gamma_1) = (d_2^{-1},\gamma_2)$ce qui contredit $\gamma_1 \neq \gamma_2$. Donc pour tous les γ_1 et γ_2 différents nous obtenons δ_1 et δ_2 différents.

L'application $\tau : I \to I$ qui à γ associe δ comme ci dessous est donc injective, donc c'est une permutation de I . *c.q.f.d*

Corollaire 2.2.1.8

Soit σ une permutation monomiale de I , (a_1,α_1) et (a_2,α_2) deux points de Ω_q. Notons $(b_1,\beta_1) = \sigma(a_1,\alpha_1)$ et $(b_2,\beta_2) = \sigma(a_2,\alpha_2)$. Alors $\alpha_1 = \alpha_2$ si et seulement si $\beta_1 = \beta_2$,de plus, si $\alpha_1 = \alpha_2$, alors $b_2 = a_2 a_1^{-1} b_1$.

Preuve

Nous avons que $\sigma(1,\alpha_1) = (a_1^{-1}b_1,\beta_1)$ et $\sigma(1,\alpha_2) = (a_2^{-1}b_2,\beta_2)$.

Si $\alpha_1 = \alpha_2$, alors $\sigma(1,\alpha_1) = \sigma(1,\alpha_2)$

d'où

$$(a_1^{-1}b_1,\beta_1) = (a_2^{-1}b_2,\beta_2),$$

donc

$$\beta_1 = \beta_2 \text{ et } b_2 = a_2 a_1^{-1} b_1.$$

Si $\beta_1 = \beta_2$, alors comme dans la démonstration de la proposition 2.2.1.7 , nous montrons que $\alpha_1 = \alpha_2$. *c.q.f.d*

Corollaire 2.2.1.9

une permutation monomiale σ de I est définie complètement par les image de $(1,\gamma)$, $\gamma \in I$ c'est à dire par j et τ de la proposition 2.2.1.7.

Preuve

Supposons que nous connaissions les images de $(1,\gamma)$ pour $\gamma \in I$.

Il faut montrer que nous pouvons déduire les images des autres points par σ .

Comme $\sigma(d,\gamma) = d\sigma(1,\gamma)$ et $\sigma(1,\gamma) = (j(\gamma),\tau(\gamma))$ sont connues

$\sigma(d,\gamma) = d(j(\gamma),\tau(\gamma)) = (dj(\gamma),\tau(\gamma))$ est déterminé de manière unique . *c.q.f.d*

lemme 2.2.1.10

Soient σ une permutation monomiale , j et τ comme dans la proposition 2.2.1.7 alors l'action de l'inverse de σ est donnée par :

$$\sigma^{-1}(1,\gamma) = \left(j\left(\tau^{-1}(\gamma)\right)^{-1},\tau^{-1}(\gamma)\right) \text{ pour tout } \gamma \in I .$$

Preuve

Nous savons que $\sigma(1,\delta) = (j(\delta),\tau(\delta))$ pour tout $\delta \in I$

Alors pour tout $\delta \in I$

$$\begin{aligned}\sigma^{-1}(j(\delta),\tau(\delta)) = (1,\delta) &\Leftrightarrow j(\delta)\sigma^{-1}(1,\tau(\delta)) = (1,\delta)\\ &\Leftrightarrow \sigma^{-1}(1,\tau(\delta)) = j(\delta)^{-1}(1,\delta)\\ &\Leftrightarrow \sigma^{-1}(1,\tau(\delta)) = \left(j(\delta)^{-1},\delta\right)\\ &\Leftrightarrow \sigma^{-1}(1,\tau(\delta)) = \left(j\left(\tau^{-1}(\tau(\delta))\right)^{-1},\tau^{-1}(\tau(\delta))\right)\end{aligned}$$

Notons $\gamma = \tau(\delta)$ .Quand δ parcourt *I*, γ parcourt aussi *I*.

Donc

$$\sigma^{-1}(1,\gamma) = \left(j\left(\tau^{-1}(\gamma)\right)^{-1},\tau^{-1}(\gamma)\right) \text{ pour tout } \gamma \in I \qquad \textit{c.q.f.d}$$

Groupe d'automorphismes 2.2.2

Notation 2.2.2.1

Nous définissons l'action d'une permutation monomiale σ de *I* sur un mot $c = (c_\gamma)_{\gamma\in I}$ de F_q^n comme suit :

$$\sigma(c) = \left(j(\gamma)^{-1}c_\delta\right)_{\gamma\in I} \quad \text{où} \quad \delta = \tau(\gamma)$$

Concrètement , nous multiplions chaque coordonnée c_γ de *c* par $j(\gamma)^{-1}$ en obtenant ainsi un vecteur $j(\gamma)^{-1}c_{\gamma\in I}$ et nous permutons les coordonnées de ce vecteurs par τ .

Nous voyons que l'action d'une permutation monomiale coïncide totalement avec l'action d'une matrice monomiale , d'abord nous multiplions les composantes de *c* par les

éléments de F_q^*,et ensuite nous permutons les coordonnées. D'ailleurs, à toute fonction $j : I \to F_q^*$ on peut associer une matrice diagonale inversible et inversement. Et à chaque permutation τ de I on peut associer une matrice de permutation et inversement. Donc les ensembles des matrice monomiale et des permutation monomiale correspondent.

Définition 2.2.2.2

Le groupe d'automorphisme $Aut(C)$ d'un code C est le sous-ensemble de toutes les permutations monomiales σ de I telles que $\sigma(C) = C$. i.e :

$$Aut(C) = \{\ \sigma \in S_q(I) / \sigma(C) = C\ \}.$$

La démonstration du fait que $Aut(C)$ forme un groupe est analogue à celle de la proposition III.2.3.

Notons que la propriété du proposition III.2.6 reste valable pour le groupe d'automorphisme.

3. Représentation d'un groupe d'automorphisme

L'algorithme de séparation des supports utilisé pour déterminer la permutation qui définie l'équivalence de deux codes équivalents par permutation s'appui sur les groupes de permutations des codes, par contre, Etudier l'équivalence des codes en s'appuyant sur les groupes d'automorphismes paraît difficile.

Dans cette section nous allons construire un code C^* à partir d'un code donné C tel que le groupe d'automorphisme de C s'exprime comme groupe de permutation de C^*.

Définition 3.1

Soit n un entier positif non nul, q une puissance d'un nombre premier, I un ensemble de cardinal utilisé pour indexer les coordonnées des mots de F_q^n, $\Omega_q = I \times F_q^*$ et C un code de longueur n sur F_q. Le code étalé C^* de C est le code définie à partir de C comme suit :

$$C^* = \left\{ \left(a^{-1} c_\gamma\right)_{(a,\gamma) \in \Omega_q} / c = \left(c_\gamma\right)_{\gamma \in I} \in C \right\}$$

alors nous avons le résultat suivant :

Proposition 3.2

Soit n *un entier positif non nul ,* q *une puissance d'un nombre premier,* I *un ensemble de cardinal* n *utilisé pour indexer les coordonnées des mots de* F_q^n *,* $\Omega_q = I \times F_q^*$ *,* C *un code de longueur* n *sur* F_q *et* C^* *le code étalé de* C *,alors*

$$Aut(C) = Perm(C^*) \cap S_q(I)$$

Preuve

Montrons d'abord que : $Aut(C) \subset Perm(C^*) \cap S_q(I)$. Nous avons par définition de $Aut(C)$ que $Aut(C) \subset S_q(I)$, il faut montrer que :

$$Aut(C) \subset Perm(C^*)$$

Soit $\sigma \in Aut(C)$, c'est-à-dire

Pour tout $c = (c_\gamma)_{\gamma \in I} \in C$, $\sigma(c) = \left(j(\gamma)^{-1} c_{\tau(\gamma)}\right)_{\gamma \in I} \in C$.

Il faut montrer que $\sigma \in Perm(C^*)$, c'est à dire pour tout $c^* \in C^*$, $\sigma(c^*) \in C^*$

où σ agit sur C^* comme une permutation et non comme une permutation monomiale .

Soit $c^* \in C^*$. Par construction de c^*, il existe $c \in C$ tel que :

$$c^* = \left(a^{-1} c_\gamma\right)_{(a,\gamma) \in \Omega_q}$$

Notons $c^*{}_{(a,\gamma)}$ la composante de c^* indexée par $(a,\gamma) \in \Omega_q$, par définition $c^*{}_{(a,\gamma)} = a^{-1} c_\gamma$

alors :

$$\begin{aligned}\sigma(c^*) &= \sigma\left(\left(c^*{}_{(a,\gamma)}\right)_{(a,\gamma)\in\Omega_q}\right) = \left(c^*{}_{\sigma(a,\gamma)}\right)_{(a,\gamma)\in\Omega_q} = \left(c^*{}_{a\sigma(1,\gamma)}\right)_{(a,\gamma)\in\Omega_q} \\ &= \left(c^*{}_{a(j(\gamma),\tau(\gamma))}\right)_{(a,\gamma)\in\Omega_q} = \left(c^*{}_{(aj(\gamma),\tau(\gamma))}\right)_{(a,\gamma)\in\Omega_q} = \left((aj(\gamma))^{-1} c_{\tau(\gamma)}\right)_{(a,\gamma)\in\Omega_q} \\ &= \left(a^{-1}\left(j(\gamma)^{-1} c_{\tau(\gamma)}\right)\right)_{(a,\gamma)\in\Omega_q} = \left(\ a^{-1}\sigma(c)\ \right)_{a\in F_q^*}\end{aligned}$$

Notons $\sigma(c) = d = (d_\gamma)_{\gamma \in I}$. nous avons donc :

$$\sigma(c^*) = \left(a^{-1}\sigma(c)\right)_{a\in F_q^*} = \left(a^{-1}(d\gamma)_{\gamma\in I}\right)_{a\in F_q^*} = \left(a^{-1} d_\gamma\right)_{(a,\gamma)\in\Omega_q} \in C^*$$

Car $d \in C$.

Donc $\sigma \in perm(C^*)$et $Aut(C) \subset perm(C^*)$.

Maintenant nous allons montrer que $perm(C^*) \cap S_q(I) \subset Aut(C)$.

Supposons que $\sigma \in perm(C^*) \cap S_q(I)$. Il faut montrer que $\sigma \in Aut(C)$, c'est à dire que pour tout $c \in C$, $\sigma(c) \in C$.

Soit $c \in C$. Montrons que $\sigma(c) \in C$. Soit c^* le mot de C^* associé, c'est à dire

$$c^* = \left(a^{-1} c_\gamma\right)_{(a,\gamma) \in \Omega_q} .$$

Exactement comme dans la première partie de la démonstration nous montrons que

$$\sigma(c^*) = \left(a^{-1}\sigma(c)\right)_{a \in F_q^*} .$$

D'ailleurs, $\sigma(c^*) \in C^*$, car $\sigma \in perm(C^*)$.

Donc

$$\left(a^* \sigma(c)\right)_{a \in F_q^*} = \left(a^{-1} d_\gamma\right)_{(a,\gamma) \in \Omega_q} \in C^*$$

Par construction de C^* nous avons que $\sigma(c) = \left(d_\gamma\right)_{\gamma \in I} \in C$. *c.q.f.d*

Commentaires 3.3

Le théorème de CAYLEY permet de représenter tout groupe comme un sous groupe du groupe symétrique $S(\Omega)$, c'est à dire il permet de voir tout groupe comme un groupe de permutation agissant sur un ensemble ; donc nous pouvons considérer le groupe $Aut(C)$ d'un code C comme un groupe de permutation de $S(Aut(C))$ mais le théorème de CAYLEY n'affirme pas que ce groupe de permutations isomorphe à $Aut(C)$, est un groupe de permutations d'un code ce qui nous permet pas d'utiliser l'algorithme de séparation des supports car ce dernier s'appuie sur les permutations qui agissent sur les mots d'un code .

D'autre part ; le théorème de CAYLEY permet de représenter chaque élément $\sigma \in Aut(C)$ comme une permutation de $Aut(C)$ c'est à dire une permutation qui agit sur $|Aut(C)|$ indice, et en général le groupe $Aut(C)$ est difficile à déterminer à fortiori son $|Aut(C)|$,mais la proposition 3.2 représente le groupe $Aut(C)$ comme un sous groupe de $Perm(C^*)$ ou nous pouvons manipuler l'algorithme de séparation des support d'une part , d'autre part elle représente tout élément de $Aut(C)$ comme une permutation agissant sur les l'ensemble $F_q^* \times I$ qui est de cardinale bien déterminer $(q-1)n$. Enfin la proposition 3.2 permet de voir les code isomorphes comme des code équivalents par permutation ce qui fait appel à l'utilisation d'un algorithme qui détermine cette permutation qui définie l'isomorphisme. (voir section prochaine)

4.1 Equivalences des code

Les notions d'équivalence et d'isomorphisme interviennent fréquemment en Algèbre . en fait pour étudier les structures d'objets mathématiques l'algorithme utilise un principe puissant : il classifie les objets qui possèdent une même structure d'une façon bien défini (appelée équivalence ou isomorphisme) en suite il étudie la structure d'un seul objet qui représente tout les autre objets de même classe que lui.

Par ce principe pour étudier les propriétés d'une structure définie sur des objets différents en nature d'éléments, il suffit d'étudier les propriétés de la mémé structure défini sur un représentant de chaque classe déterminée.

Par exemple , pour étudier les propriétés d'un groupe cyclique fini d'ordre n d'un corps fin de cardinal q , il suffit d'étudier celle de Z/nZ, F_q ,…

En abordant ce principe aux codes sur un corps fini F_q nous pouvons étudier les propriétés des codes en étudiant celle d'un pour chaque classe définie par ce principe .

Dans cette section en utilisant ce principe , nous définissons l'équivalences des codes , nous montrons quelques propriétés concernent les codes équivalents , et en fin nous montrons que l'équivalence peut être vu comme équivalence par permutations ce qui permet d'utiliser l'algorithme de Séparation des supports dû à NICOLAS SENDRIER.

Equivalence par permutation 4.1

Soient n un entier naturel non nul , et $I=\{1,2,..,n\}$ ensemble pour indexer les coordonnées des mots de F_q^n .

Pour tout $x\in F_q^n$;notons $x=(x_1,x_2,..,x_n)$ ou simplement $x=x_1x_2....x_n$.

Rappelons l'action du groupe symétrique S_n sur F_q^n .

Pour toute permutation $\sigma\in S_n$, et pour $x\in F_q^n$: σ agit sur x comme suit :

$$\sigma(x)=(x_{\sigma(1)},x_{\sigma(2)},....,x_{\sigma(n)})$$

Une permutation $\sigma\in S_n$ sera notée :

$$\sigma=\begin{pmatrix} 1 & 2 & . & . & . & . & n \\ \sigma(1) & \sigma(2) & . & . & . & . & \sigma(n) \end{pmatrix}.$$

Soit $\mathbb{C}_n$ l'ensemble des codes de longueur *n* sur F_q .donc $\mathbb{C}_n$ est une partie de $P(F_q^n)$ l'ensemble de toutes les parties de F_q^n .

Soient $\sigma \in S_n$ et C un code de $\mathbb{C}_n$, définissons l'application φ de $S_n \times \mathbb{C}_n$ dans $\mathbb{C}_n$ par : $\varphi(\sigma, C) = \sigma(C)$ avec :

$$\sigma(C) = \{\sigma(x) / \ x \in C\}.$$

Proposition 4.2

L'application φ définie une opération de S_n sur $\mathbb{C}_n$. (c'est-à-dire S_n opère sur $\mathbb{C}_n$.)

Preuve

soient $\sigma, \tau \in S_n$ et C un code de $\mathbb{C}_n$.. $\sigma(C)$ est un sous ensemble de F_q^n, donc un élément de $\mathbb{C}_n$..

$$\sigma\tau(C) = \{\ \sigma\tau(x) / \ \ x \in C\}$$

Puisque

$$\begin{aligned}\sigma\tau(x) = (x_{\sigma\tau(1)}, ..., x_{\sigma(\tau(n))}) &= (x_{\sigma(\tau(1)),}, ..., x_{\sigma(\tau(n))}) \\ &= \sigma(x_{\tau(1)}, ..., x_{\tau(n)}) \\ &= \sigma(\tau(x))\end{aligned}$$

Donc

$$\sigma\tau(C) = \{\sigma(\tau(x)) / \ x \in C\} = \sigma(\tau(C)).$$

Ce qui prouve que $\varphi(\sigma\tau, C) = \varphi(\sigma, \tau(C))$

soit *id* l'identité de S_n .

$$\begin{aligned}\varphi(id, C) = id(C) &= \{id\ \ (x) / \ x \in C\} \\ &= \{x / \ x \in C\} \\ &= C\end{aligned}$$

En résumant , le groupe S_n opère sur $\mathbb{C}_n$. par :

$$(\sigma, C) \to \sigma\, C.$$ *c.q.f.d*

L' opération φ permet de définir une relation d'équivalence sur $\mathbb{C}_n$

En effet ; soit ~ la relation sur $\mathbb{C}_n$ définie par :

Pour deux codes C et C' de $\mathbb{F}_n$;

$$C \sim C' \Leftrightarrow \exists \sigma \in S_n ; C' = \sigma(C) \text{ .}$$

Proposition 4.3

La relation $\sim$ définie sur $\mathbb{F}_n$ est une relation d'équivalence .

Preuve

1) $\sim$ est réflexive car :pour tout code C de $\mathbb{F}_n$;

$$C = id(C)$$

Ce qui entraîne que : $C \sim C$.

2) $\sim$ est symétrique : soient C et C' de $\mathbb{F}_n$ tels que $C \sim C'$

nous avons :

$$C \sim C' \Leftrightarrow \exists \sigma \in S_n ; C' = \sigma(C)$$

Or

$$\sigma \in S_n \Leftrightarrow \overset{-1}{\sigma} \in S_n .$$

donc

$$C' = \sigma(C) \Leftrightarrow \overset{-1}{\sigma}(C') = \overset{-1}{\sigma}(\sigma(C)) \Leftrightarrow \overset{-1}{\sigma}(C') = \left(\overset{-1}{\sigma}\sigma\right)(C)$$

$$\Leftrightarrow \overset{-1}{\sigma}(C') = id(C) \Leftrightarrow \overset{-1}{\sigma}(C') = C \Leftrightarrow C' \sim C$$

3) $\sim$ est transitive : soient C, C' et C'' de $\mathbb{F}_n$ tels que : $C \sim C'$ et $C' \sim C''$

$$C \sim C' \text{ et } C' \sim C'' \Leftrightarrow \exists \sigma \in S_n , \exists \tau \in S_n : C' = \sigma(C) \text{ et } C'' = \tau(C')$$

$$\Rightarrow \exists \sigma, \tau \in S_n : C'' = \tau\sigma(C)$$

or $\tau, \sigma \in S_n$ *et* S_n est un groupe , alors $\tau\sigma \in S_n$

Donc $C'' = \tau\sigma(C)$ce qui prouve que : $C \sim C''$ *c.q.f.d*

Définition 4.4

Deux codes de même longueur n sur F_q ,sont équivalents par permutations s'ils sont équivalents au sens de la relation $\approx$ définie ci – dessus .

Cela revient à dire que de code C et C' de même longueur sont équivalents par permutation s'il existe une permutation $\sigma \in S_n$ telle que : $C' = \sigma(C)$.

Exemple

En se référant aux exemple 3.2.5 nous voyons que :

1) Le code C défini dans l'exemple *c)* est équivalent par permutation à lui même et il n'existe pas un autre code de longueur *3* équivalent par permutation à C autre que C.
2) Le code L défini dans l'exemple *d)* possède trois code équivalent par permutation, à savoir :
   - ✓ Le code L lui même déduit par les permutations id et (*13*).
   - ✓ Le code L_1 déduit de L par (*12*) et (*123*) .
   - ✓ Le code L_2 déduit de L par (*23*) et (*132*) .

Définition 4.6

Soient C un code de longueur n sur F_q.

La classe de C selon la relation ~ est appelée orbite de C et sera noté $\overline{C}$. L'ensemble $S_C = \{\sigma \in S_n / \sigma(C) = C\}$ est appelé le stabilisateur de C par S_n.

Pour l'exemple *1)* de 4.3, nous avons $\overline{C} = \{C\}$, tandis que pour l'exemple *2)* nous avons $\overline{L} = \{L, L_1, L_2\}$.

Proposition 4.7

Le stabilisateur de C par S_n n'est autre que son groupe de permutation.

Preuve

Immédiate en voyons que $perm(C) = \{\sigma \in S_n / \sigma(C) = C\}$. *c.q.f.d*

L'exemple *2)* de 4.3 montre qu'il y a *3* codes équivalents au code *L*, et que le code L_1 est obtenu de L en agissant les deux permutations (*12*) et (*123*) sur L, il est logique donc de se demander que si *C* est un code donné combien de codes équivalents par permutation à *C* existe-ils ? et si C' est l'un d'entre eux C' est-il déduit de C par une permutation unique ou par plusieurs ?

Les propositions suivantes donnent une réponse à cette question .

Proposition 4.8

Le nombre des codes équivalents par permutation à un code C de longueur n est

$$\frac{n!}{|perm(C)|} .$$

Preuve

Soit $\overline{C}$ l'orbite de *C* selon l'action de S_n sur $\mathbb{F}_2^n$

Soit encore $(S_n / perm(C))_g$ l'ensemble des classes à gauche de S_n modulo $perm(C)$

Définissons l'application $h : \overline{C} \to (S_n / perm(C))_g$ définie comme suit :

pour tout $C_1 = \pi(C)$ de $\overline{C}$:

$$h(C_1) = \pi perm(C).$$

Montrons que *h* est une bijection :

Soient C_1, C_2 deux codes de $\overline{C}$ tels que $C_1 = C_2$, alors il existent deux permutations $\pi_1, \pi_2 \in S_n$ telles que : $C_1 = \pi_1(C)$ et $C_2 = \pi_2(C)$,

$$C_1 = C_2 \Leftrightarrow \pi_1(C) = \pi_2(C)$$

$$\Leftrightarrow \pi_2^{-1}\pi_1(C) = C$$

$$\Leftrightarrow \pi_2^{-1}\pi_1 \in perm(C)$$

$$\Leftrightarrow \pi_1 perm(C) = \pi_2 perm(C)$$

Ce que prouve que *h* est une application injective , il reste à montrer qu'il est surjective .

Soit $\sigma\ perm(C) \in (S_n / perm(C))_g$;alors $\sigma(C)$ est un code équivalent par permutation à *C* et $h(\sigma(C)) = \sigma\ perm(C)$. Ce qui entraîne avec la première partie de la démonstration que *h* est bijective,

Donc les deux ensembles $\overline{C}$ et $(S_n / perm(C))_g$ ont même cardinal.

$$\left|\overline{C}\right| = \left|(S_n / perm(C))_g\right| = [S_n : perm(C)]$$

$$= \frac{|S_n|}{|perm(C)|} \qquad (\textit{Théorème de Lagrange})$$

$$= \frac{n!}{|perm(C)|} \qquad \textit{c.q.f.d.}$$

Exemples 4.9

En se référant à l' exemple 2.1.4

a) Un code de répétition possède un seul équivalent par permutation car

$$\frac{n!}{|perm(C)|} = \frac{n!}{n!} = 1$$

ce code équivalent n'est autre que le code de répétition lui même.

b) Le code de cet exemple possède $\frac{4!}{8} = 3$ codes équivalent par permutation .

c) Le code C de cet exemple possède un seul code équivalent par permutation qui n'est autre que C.

d) Le code L de cet exemple possède $\frac{3!}{2} = 3$ codes équivalents par permutation à savoir les codes L, L_1, L_2.

Passons maintenant à répondre à la deuxième partie de la question posée au dessous 4.5 c'est à dire la question suivante :

Si C et C' sont deux codes équivalents par permutation, combien de permutation $\sigma \in S_n$ établissent $C' = \sigma(C)$ et dans quel cas σ est unique ?

Afin de répondre à cette question , nous définissons une relation sur S_n notée $\equiv$ par ; pour toutes permutations $\sigma_1, \sigma_2 \in S_n$;

$$\sigma_1 \equiv \sigma_2 \Leftrightarrow \sigma_1(C) = \sigma_2(C)$$

Il est évident, que la relation $\equiv$ ainsi définie, est une relation d'équivalence sur S_n.

Cette relation qui veut dire que $\sigma_1 \equiv \sigma_2$, si et seulement si σ_1 et σ_2 définissent le même code équivalent par permutation à C.

Analysons de plus cette relation :

$$\begin{aligned} \sigma_1 \equiv \sigma_2 &\Leftrightarrow \sigma_1(C) = \sigma_2(C) \\ &\Leftrightarrow \sigma_2^{-1}\sigma_1(C) = C \\ &\Leftrightarrow \sigma_2^{-1}\sigma_1 \in perm(C) \\ &\Leftrightarrow \sigma_1 perm(C) = \sigma_2 perm(C) \end{aligned}$$

Cette dernière égalité veut dire que les classe à droite modulo $perm(C)$ de σ_1 et σ_2 sont égaux.

Cela permet de définir une application bijective T de $S_n / \equiv$ ensemble des classes modulo $\equiv$ sur $(S_n / perm(C))_d$ ensemble des classes à droite modulo $perm(C)$ comme suit :

$$T : S_n / \equiv \to (S_n / perm(C))_d$$

$$\overline{\sigma} \mapsto \sigma\ perm(C)$$

Puisque T est bijective , alors

$$|S_n / \equiv| = |(S_n / perm(C))_d|$$

$$= [S_n : perm(C)]$$

$$= \frac{n!}{\lfloor perm(C) \rfloor}$$

C'est à dire que le nombre des permutations prises comme représentant selon la relation $\equiv$ est le même nombre des codes équivalents à C.

Calculons maintenant le cardinal de $\overline{\sigma}$, c'est à dire que le nombre des permutations que lors ses actions sur C produisent le même code équivalent à C.

Pour cela étudions l'application $\varphi : perm(C) \to \overline{\sigma}$ définie par :

pour tout $\pi \in perm(C)$, $\varphi(\pi) = \sigma\pi^{-1}$.

Au premier lieu φ est bien définie et de plus elle est injective car,

si $\pi_1, \pi_2 \in perm(C)$, alors $\pi_1(C) = C$ et $\pi_2(C) = C$.

Donc nous avons

$$\varphi(\pi_1) = \sigma\pi_1^{-1} \text{ et } \varphi(\pi_2) = \sigma\pi_2^{-1}$$

$$\pi_1 = \pi_2 \Leftrightarrow \sigma\pi_1^{-1} = \sigma\pi_2^{-1} \Leftrightarrow \varphi(\pi_1) = \varphi(\pi_2)$$

Montrons que φ est surjective, soit $\tau \in \overline{\sigma}$, alors la permutation τ vérifie

$$\tau(C) = \sigma(C) \Leftrightarrow \tau^{-1}\sigma(C) = C$$

$$\Leftrightarrow \tau^{-1}\sigma \in perm(C)$$

et de plus

$$\varphi(\tau^{-1}\sigma) = \sigma(\tau^{-1}\sigma)^{-1} = \sigma\sigma^{-1}(\tau^{-1})^{-1} = id.\tau = \tau .$$

Ce qui montre que φ est surjective. En résumant φ est bijection de $perm(C)$ dans $\overline{\sigma}$,ce qui permet de conclure que les ensembles $perm(C)$ et $\overline{\sigma}$ ont même cardinal , nous pouvons donc annoncer la proposition suivante :

Proposition 4.10

Soit C un code de longueur n sur F_q. alors :

1) *Le nombre des permutations de S_n qui produisent un même code équivalent par Permutation à C , est $|perm(C)|$.*
2) *Si C' est un code équivalent à C par une permutation σ , alors σ est unique si et seulement si le groupe de permutation de C est trivial (c'est à dire réduit à l'identité).*

La proposition précédente affirme que si C' est équivalent à C par une permutation σ ,il suffit de déterminer l'ensemble $\{\sigma\pi^{-1} / \pi \in perm(C)\}$ pour déterminer toutes les permutations qui produisent C' à partir de C.

La deuxième affirmation de la proposition suivante donne une condition nécessaire et suffisante pour que la permutation σ soit unique, à savoir le groupe de permutation de C est trivial. Mais que dire du groupe de permutation de C' ?

La réponse est apportée par la proposition suivante :

Proposition 4.11

Deux codes de même longueur équivalents par permutation, ont des groupes de permutation isomorphes.

Preuve

L'idée de la démonstration est la suivante :

Soient C et C' deux codes équivalents par permutation $\sigma \in S_n$ tel que $\sigma(C) = C'$ et soit $\pi \in perm(C)$.

$$\sigma(C) = C' \Leftrightarrow \sigma\pi(C) = \sigma(C) = C'$$

or

$$\sigma(C) = C' \Leftrightarrow C = \sigma^{-1}(C')$$

Alors :

$$\sigma(C) = C' \Leftrightarrow \sigma\pi\sigma^{-1}(C') = C'$$

C'est-à-dire que si $\pi \in perm(C)$, alors $\sigma\pi\sigma^{-1} \in perm(C')$.

Cette remarque, nous amène à définir l'application ψ de $perm(C)$ dans $perm(C')$ par :

$$\psi : perm(C) \to perm(C')$$

$$\pi \mapsto \sigma\pi\sigma^{-1}$$

Soient $\pi_1, \pi_2 \in perm(C), \tau \in perm(C')$

$$\psi(\pi_1) = \psi(\pi_2) \Leftrightarrow \sigma\pi_1\sigma^{-1} = \sigma\pi_2\sigma^{-1}$$

$$\Leftrightarrow \pi_1 = \pi_2$$

ce qui permet de déduire que ψ est injective

$$\psi(\sigma^{-1}\tau\sigma) = \sigma(\sigma^{-1}\tau\sigma)\sigma^{-1} = (\sigma\sigma^{-1})\tau(\sigma\sigma^{-1}) = \tau$$

ce qui affirme que ψ est surjective

En résumant ψ est bijective

$$\psi(\pi_1 .\pi_2) = \sigma\pi_1\pi_2\sigma^{-1} = (\sigma\pi_1\sigma^{-1})(\sigma\pi_2\sigma^{-1}) = \psi(\pi_1)\psi(\pi_2)$$

c'est à dire que ψ est un homomorphisme de groupe .

En résumant tous les résultats obtenus lors de cette démonstration, nous pouvons affirmer que ψ est un isomorphisme de groupes.

L'isomorphisme ψ définie ci-dessous nous permet d'obtenir un groupe de permutation d'un code C',équivalent à un code C par une permutation σ, à partir du groupe de permutation de *C*.

En effet,

$$perm(C') = \sigma\ perm(C).\sigma^{-1} = \{\sigma\pi\sigma^{-1} / \pi \in perm(C)\}.$$

En se référant au chapitre *I*, nous pouvons conclure que $perm(C')$ est le conjugué de $perm(C)$ par σ.

Revenons à notre question posée au dessous de la proposition 4.8 , nous pouvons donc répondre facilement que $perm(C')$ est aussi trivial si la permutation σ est unique .

Proposition 4.12

Deux codes équivalents par permutation Ont :

1) même longueur .

2) même distribution des poids .

3) même distance minimale .

4) même polynôme énumérateur des poids.

Preuve

Soient C et C' deux codes équivalents par permutation, et soit σ une permutation qui réalise cette équivalence, c'est à dire $C' = \sigma(C)$.

1) Il résulte de la définition de l'équivalente, car C et C' sont parties de F_q^n.

2) Pour $x \in C$ de poids $\omega(x) = k$,l'élément $\sigma(x) \in C'$ est aussi de poids k car σ permute les positions des composantes x_i, $1 \le i \le n$ de x sans changer leurs valeurs dans F_q. Donc $\omega(x) = \omega(\sigma(x))$ pour tout $x \in C$.

L'application :

$$\{x \in C / \omega(x) = k\} \to \{\sigma(x) \in C' / \omega(\sigma(x)) = k\}$$

$$x \mapsto \sigma(x)$$

est une bijection (car cette application n'est autre que σ restreinte sur le premier ensemble sur son image par σ). Alors les ensembles précédents ont même cardinal pour tout k avec $0 \le k \le n$:

$$\left|\{x \in C / \omega(x) = k\}\right| = \left|\{\sigma(x) \in C' / \omega(\sigma(x)) = k\}\right| \qquad \forall k \in \{0,1,...,n\}.$$

Donc nous avons $A_k = B_k$ pour tout $k \in \{0,1,.,.,.,n\}$ où $(A_k)_{0 \le k \le n}$, $(B_k)_{0 \le k \le n}$ sont, respectivement, les distributions des poids de C et de C'. En concluant $(A_k)_{0 \le k \le n} = (B_k)_{0 \le k \le n}$.

4) il résulte de *2),* puisque pour tout $k \in \{0,1,.,.,.,n\}$;nous avons $A_k = B_k$.

et comme $(A_k)_{0 \le k \le n}$, $(B_k)_{0 \le k \le n}$ sont respectivement les coefficients du $W_C(x,y)$ et $W_{C'}(x,y)$ nous en déduirons que $W_C(x,y) = W_{C'}(x,y)$.

3) il résulte du fait que la distance minimale d d'un code est le plus petit indice non nul i dans la suite du distribution des poids $(A_k)_{1 \le k \le n}$ tel que $A_i \neq \phi$. *c.q.f.d*

Remarque :4.13

La proposition 4.10 peut être utilisé dans un sens inverse. Si deux codes C et C' n'ont pas même longueur ou même distance minimale , ou même distribution des poids, ou même polynôme énumérateur des poids , nous pouvons affirmer que les deux codes ne sont pas équivalent permutation, par exemple :

1) les deux codes de matrices génératrices $G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ et $G' = (1101)$ sur F_2 ne sont pas équivalents par permutation.

2) Les codes $C=\{000,101,110,111\}$ et $C'=\{000,100,001,101,111\}$ ne sont pas équivalents par permutation .

3) Les deux codes C et $C^{\perp}$ cités pour l'exemple *8.2.d)* ne sont pas équivalents par permutation .

La réciproque de la proposition 4.10 est fausse ; deux codes C et C' possédants même longueur, même polynôme énumérateur des poids, n'impliquent pas que ces codes sont équivalent par permutation.

Pour finir cette section, montrons la proposition suivante :

Proposition 4.11

L'équivalence par permutation conserve la linéarité c'est-à-dire, si C est un code linéaire $[n,k,d]$ *alors tout code équivalent à* C *est aussi un code linéaire* $[n,k,d]$.

Preuve

Supposons que C est un code linéaire de paramètres $[n,k,d]$ sur F_q . et soit C' un code équivalent à C par une permutation σ .

Nous avons : $\sigma(C)=C'$, montrons que C' est aussi linéaire de paramètres $[n,k,d]$.

Soient $x',y'\in C'$ et $\lambda\in F_q$ alors $x'=\sigma(x)$ et $y'=\sigma(y)$ avec : $x,y\in C$

Posons $x=(x_i)_{1\le i\le n}$, $y=(y_i)_{1\le i\le n}$, $x'=(x'_i)_{1\le i\le n}$ et $y'=(y'_i)_{1\le i\le n}$

Alors

$$x'-y'=(x'_i-y'_i)_{1\le i\le n}=(x_{\sigma(i)}-y_{\sigma(i)})_{1\le i\le n}=\sigma(x-y)\in C'$$

et

$$\lambda x'=\lambda(x'_i)_{1\le i\le n}=(\lambda x'_i)_{1\le i\le n}=(\lambda x_{\sigma(i)})_{1\le i\le n}=\sigma(\lambda x_i)_{1\le i\le n}=\sigma(\lambda x)\in C'$$

ce qui prouve que C' est linéaire.

or

$$C'=\sigma(C) \text{ alors } |C'|=|\sigma(C)|=|C|$$

Donc $q^k=|C'|=q^{k'}$ avec $k=dim\,C$ et $k'=dim\,C'$

Enfin nous avons $k=k'$. Nous pouvons conclure que C' est linéaire de paramètres $[n,k,d]$

Equivalence ou isomorphisme de codes 4.15

Comme dans 3.2 nous définissons l'équivalence ou l' isomorphisme de deux codes C et C' de même longueur *n* sur un corps fini F_q.

Définition 4.15.1

Soit C et C' deux codes de longueur *n* sur F_q. Nous disons qu'ils sont *équivalents* (ou isomorphes) s'il existe une permutation monomiale σ de *I* telle que :

$$C' = \sigma(C) \text{ .où } \sigma(C) = \{\sigma(c) / c \in C\} .$$

D'ailleurs , il est possible d'exprimer l'équivalence entre deux codes *C* et C' par l'équivalence par permutation entre C^* et C'^*. Ce qui permet d'affirmer que toutes les propriétés étudiées en section 4 sont vraies aussi pour des codes équivalents, de plus il est possible d'appliquer l'algorithme de SENDRIER pour déterminer la permutation monomiale qui produit C' à partir de *C* car elle n'est que la permutation qui produit C'^* à partir de C^*.

Donc nous pouvons annoncer la proposition suivante :

Proposition 4.15.2

Soient C et C' deux codes de longueur n sur F_q. Les deux codes C et C' sont équivalents si et seulement si leurs codes étalés C^* et C'^* *sont équivalents par permutation de* $S_q(I)$ *.Et de plus* $\sigma(C) = C'$ *pour* $\sigma \in S_q(I)$ *si et seulement* $\sigma(C^*) = C'^*$.

Preuve

La démonstration est analogue , à la démonstration de la proposition 3.2.

Remarque 4.15.3

Quand on implante les permutations monomiales de *I* on les représente habituellement par les permutations de $\{1,2,...,|\Omega_q|\}$ où $\Omega_q = F_q^* \times I$.

Soit $F_q^* = \{a_1,....,a_{q-1}\}$ avec $a_1 < a_2 < ... < a_{q-1}$ et soit $I = \{1,2,....n\}$.Ordonnons par exemple Ω_q selon l'ordre lexicographique c'est à dire si (a,γ) et $(\alpha,\delta) \in \Omega_q$,

$$(a,\gamma) < (d,\delta) \text{ si et seulement si } a < d \text{ ou } a = d \text{ et } \gamma < \delta .$$

Plus précisément

$$(a_1,1) < (a_1,2) < .... < (a_1,n) < (a_2,1) < .... < (a_2,n) .... < (a_{q-1},1) < .... < (a_{q-1},n)$$

alors le mot $c^* = (c_{(a,\gamma)})_{(a,\gamma) \in \Omega_q}$ de C^* correspondant à un mot $c = (c_\delta)_{\delta \in I}$ de *C* est

$$c^* = (c^*_{(a_1,1)},...,c^*_{(a_1,n)},c^*_{(a_2,1)},...c^*_{(a_2,n)},...c^*_{(a_{q-1},1)},...c^*_{(a_{q-1},n)})$$

$$= (a_1^{-1}c_1, a_1^{-1}c_n, a_2^{-1}c_1,...,a_2^{-1}c_n,...,a_{q-1}^{-1}c_1,...,a_{q-1}^{-1}c_n)$$

$$= \left(a_1^{-1}(c_1, c_2, ...c_n) \middle| a_2^{-1}(c_1, ..., c_n) \middle| ... \middle| a_{q-1}^{-1}(c_1, ..., c_n)\right)$$

$$= \left(a_1^{-1}c \middle| a_2^{-1}c ... \middle| a_{q-1}^{-1}c\right)$$

ce qui signifie que :

$$C^* = \left(a_1^{-1}C \middle| a_2^{-1}C \middle| ... a_{q-1}^{-1}C\right)$$

DETERMINATION DE L'EQUIVALENCE ENTRE DEUX CODES

Il s'agit ici d'essayer de répondre aux deux questions suivantes :

1. Etant donné deux code C et C', décider si C et C' sont équivalents par permutation ?
2. Si C et C' sont équivalents par permutation, retrouver la permutation σ telle que $\sigma(C)=C'$?.

PETRANK et ROTH ont prouvé que résoudre le problème de la question (1) est au moins difficile que le problème de l'isomorphisme des graphes [voir pour plus de détail PL 01] , et pour la question (2) , la permutation σ à retrouver est unique , si est seulement si le groupe de permutation de C (ou de C') est trivial d'après la proposition III.4.8.

1. Notations et définitions

I désigne toujours l'ensemble $\{1,2,\ldots,n\} \subset N$.

Pour tout $x \in F_q^n$, $\operatorname{supp}(x)=\{i \in I : x_i \neq 0\}$ désigne le support de x .

Nous rappelons que deux codes C et C' de longueur n sur F_q sont équivalents par permutation, s'il existe une permutation $\sigma \in S_n$ telle que : $C'=\sigma(C)$, et nous noterons donc $C \sim C'$ ou $C \sim \sigma(C)$.

Perm (C) désigne le groupe de permutations de C .

2. Codes poinçonnés

Soit C un code de longueur n sur F_q .

Si J est une partie non vide de I, nous noterons E_J l'ensemble des mots de F_q^n du supports inclus dans J c'est à dire :

$$E_J=\{x \in F_q^n : \operatorname{supp}(x) \subset J\}$$

et si J est le singleton $\{i\}$, nous noterons simplement E_i :

$$E_i = \{x \in F_q^n : \operatorname{supp}(x)=\{i\}\} = \{ (0, 0,\ldots, 0, \alpha, 0,.. ,0) \in F_q^n \quad / \alpha \in F_q^* \}.$$

Définition 2.1

Le code C poinçonné en i est par définition :

$$C_i = (C + E_i) \cap E_{I \setminus \{ i \}}$$

Remarque 2.2

1) L'ensemble C_i est bien un code sur F_q , car il est une partie de F_q^n. Les éléments du code C_i sont tous les éléments de C où les coordonnées indexées par i sont remplacées par zéro (zéro de F_q) .
2) Si le code C est linéaire, il en est de même pour le code poinçonné C_i puisque il est intersection de deux sous espaces vectoriels de F_q .
3) Dans la définition usuelle d'un code poinçonné en i , la position indexée par i est supprimée [par exemple MS77,ch1] ce qui produit un code de longueur $n - 1$. La définition que nous adoptons permet d'indexer les coordonnées des mots des deux codes C et C_i par le même ensemble I.

Propriétés des codes poinçonnes 2.3

a) La commutativité : Pour tout code C de longueur n et pour i, j∈I

nous avons :

$$(C_i)_j = (C_j)_i$$

Preuve

Elle découle de la remarque 2.2

Nous noterons alors chacun de $(C_i)_j$, et $(C_j)_i$ tout simplement par $C_{\{i,j\}}$.

b) La somme : Pour deux codes B et C de langueur n et pour tout i∈I

nous avons : $(B + C)_i = B_i + C_i$.

Preuve

Soit $x \in (B + C)_i$, alors : $x = (x_1,x_2,\ldots,x_{i-1}, 0 , x_{i+1} , \ldots,x_n)$ tell qu'il existe un certain $x_i \in F_q$ où le mot

$$\bar{x} = (x_1,x_2,\ldots,x_{i-1}, x_i , x_{i+1} , \ldots,x_n) \in B + C .$$

$\bar{x}$ s'écrit donc comme suit :

$\bar{x} = b+c = (b_1 , b_2 ,\ldots, b_n) + (c_1 , c_2 ,\ldots, c_n)$ avec $b \in B$ et $c \in C$.

ce qui entraîne que :

$$x = (b_1 , b_2 ,\ldots, b_{i-1} , 0 , b_{i+1} , b_n) + (c_1 , c_2 ,\ldots, c_{i-1} , 0 , c_{i+1} , c_n)$$

où le premier mot est obtenu à partir de b en remplaçant b_i par 0 , et le deuxième est obtenu à partie de c en remplaçant c_i par 0 .ce qui proue que $x \in B_i + C_i$.

Réciproquement, on montre que si $x \in B_i + C_i$ alors $x \in (B+C)_i$ par la même manière que la précédente. *c.q.f.d*

c) Equivalence : Pour tout code C de longueur n , pour tout $i \in I$, et pour toute permutation $\sigma \in S_n$., nous avons :

$$\sigma(C_i) = \sigma(C)_{\sigma(i)} .$$

Preuve

la permutation σ change les positions des coordonnées d'un mot d'un code sans changer leurs valeurs . Alors si $b_i = 0$ pour $b = (b_1 , b_2 ,\ldots, b_n)$ de C , il est évident que $b_{\sigma(i)} = 0$ pour $\sigma(b) \in \sigma(C)$.

donc :

$$\begin{aligned}\sigma(C_i) &= \{ \sigma(b) / b \in C_i \} \\ &= \{ \sigma(b) / b \in C \text{ en remplaçant } b_i \text{ par } 0 \} \\ &= \{(b_{\sigma(1)} ,\ldots , b_{\sigma(n)}) \in \sigma(C) \text{ en remplaçant } b_{\sigma(i)} \text{ par } 0 \} \\ &= \sigma(C)_{\sigma(i)} . \end{aligned}$$ *c.q.f.d*

la propriété 2.3.*c).* affirme que si C et C' sont deux codes équivalents par permutation avec $\sigma (C) = (C')$, alors il en est de même pour C_i et $C'_{\sigma(i)}$ et de plus $\sigma(C_i) = C'_{\sigma(i)}$.

d) Dimension : pour tout code linéaire C et pour tout $i \in I$, nous avons :

$$\dim C_i = \dim C - \dim (C \cap E_i)$$

Preuve

Pour tout $x = (x_1,x_2,\ldots,x_{i-1}, x_i , x_{i+1} , \ldots,x_n)$ nous avons :

$$x = (x_1,x_2,\ldots,x_{i-1}, 0 , x_{i+1} , \ldots,x_n) + (0,0,0,\ldots,0, x_i , 0,0) = y + z$$

Il est clair que $y \in C_i$ et $z \in C \cap E_i$, ce qui entraîne que : $C = C_i + (C \cap E_i)$

En remarquant que $C_i \cap (C \cap E_i) = \{ 0 \}$, nous pouvons conclure que :

$$C = C_i \oplus (C \cap E_i) ,$$ *c.q.f.d*

3. Invariants

Deux codes équivalents par permutations possèdent des propriétés communes, par exemple, ils ont même distance minimale, même polynôme énumérateurs des poids … etc.

Ces propriétés sont invariantes sur chaque classe d'équivalence de codes, tout ça nous ramènent à défini la notion d'invariant liée à celle d'équivalence.

Notations 3.1

Notons $\mathbb{C}_n$ l'ensemble de tous les codes de longueur n sur F_q. l'ensemble $\mathbb{C} = \bigcup_{n \geq 1} \mathbb{C}_n$ est l'ensemble de tous les codes sur F_q .

Soit E un ensemble non vide sur lequel la notion d'égalité est définie.

Définition 3.2

Un invariant sur E est une application $\nu: \mathbb{C} \to E$ telle que deux codes équivalents prennent la même valeur par ν, c'est a dire :

$$\forall C \in \mathbb{C}_n , \ \forall \sigma \in S_n : \nu(\sigma(C)) = \nu(C).$$

Exemples 3.3

a. La longueur d'un code est un invariant sur N .

b. La distance minimale est un invariant sur N.

c. Le polynôme énumérateur $W(C) = \sum_{c \in C} X^{\omega(c)}$ dans sa forme simple, est un invariant sur $N[X]$.

d. L'application $\nu : \mathbb{C} \to 2^{\mathbb{C}}$

$$C \to \{ (\sigma(C) / \sigma \in S_n , \ C \in \mathbb{C}_n \}$$

est un invariant où $2^{\mathbb{C}}$ désigne l'ensemble des parties de $\mathbb{C}$.

Par définition $\nu(C) = \nu(C')$ si et seulement si , C et C' sont équivalents par permutations.

Bien entendu, la complexité du calcul de ν(C) rend cet invariant inutilisable en pratique même pour des petites valeurs de n.

Corollaire 3.4

Soit ν un invariant et soit C un code de longueur n, pour tout $i \in I$ et pour toute permutation $\sigma \in S_n$; nous avons :

$$\nu(C_i) = \nu(\sigma(C)_{\sigma(i)}).$$

Preuve

puisque ν est un invariant, nous avons :

$$\nu(C_i) = \nu(\sigma(C_i)) = \nu(\sigma(C)_{\sigma(i)}). \qquad c.q.f.d$$

Cela revient à dire que si C et C' sont deux codes équivalents par permutation tels que $\sigma(C) = C'$. alors $\nu(C_i) = \nu(C'_{\sigma(i)})$, par tout $i \in I$.

Définition 3.5

Le *Hull* d'un code linéaire C est l'intersection de C et son dual $C^{\perp}$ que nous le noterons $H(C)$. c'est à dire $H(C) = C \cap C^{\perp}$.

il est à remarquer que le Hull d'un code linéaire C est aussi un code linéaire.

Proposition 3.6

Soit C un code linéaire de longueur n et $\sigma \in S_n$.

Alors :

1. $H(\sigma(C)) = \sigma(H(C))$.
2. *Si ν est invariant, l'application $C \to \nu(H(C))$ est aussi un invariant.*

Preuve

Pour *1)* il suffit de remarquer que :

$$\sigma(C^{\perp}) = \sigma(C)^{\perp} \text{ et } \sigma(A \cap B) = \sigma(A) \cap \sigma(B).$$

Pour *2)* il suffit d'appliquer la définition d'un invariant. *c.q.f.d*

L'invariant est une propriété globale d'un code, il peut nous aider à décider si deux codes sont équivalents où non dans certains cas, par exemple, deux codes de valeurs différentes par un invariant ne sont pas équivalents. Mais il peut arriver que deux codes non équivalents ont la même valeur par un invariant, ce qui est le cas par exemple pour le polynôme énumérateur, la longueur … etc. Pour ces raisons nous allons définir une propriété locale d'un code et une de ses positions.

4. Signatures

Définition 4.1

Une signature S sur un ensemble E, est une application qui à tout code C de longueur n et à tout élément i de I, associe un élément $S(C, i)$ de E et telle que pour toute permutation $\sigma \in S_n$ et pour tout i de I :

$$S(\sigma(C), \sigma(i)) = S(C, i).$$

Exemple 4.2

On peut construire une signature à partir de tout invariant. Soit ν un invariant, pour tout code C de longueur n, et pour tout $i \in I$, l'application définie par :

$$S(C, i) = \nu(C_i)$$

est une signature.

Cette exemple donne une approche à la réponse de la première question au début de ce chapitre. En effet, si nous avons un invariant v et nous voulons savoir si deux codes C et C' sont équivalents, nous pouvons remarquer que si $C' = \sigma(C)$ alors nous avons $v(C_i) = v(C'_{\sigma(i)})$, pour tout $i \in I$.

Construction de signatures 4.3

Soient S et T deux signatures sur deux ensembles E et E' respectivement.

1- La signature produit de S et T est la signature, notée $S \times T$, définie par :

$$S \times T : (C, i) \rightarrow (S(C, i), T(C, i)).$$

2- Le dual de S est la signature sur E, notée $S^{\perp}$, définie par :

$$S^{\perp} : (C, i) \rightarrow S(C^{\perp}, i)$$

Comparaison des signatures 4.4

Soient S et T deux signatures et soit C un code de longueur n.

1)- la signature T est plus discriminante que la signature S pour C, et nous noterons $S \leq_C T$ si :

$$\forall i, j \in I, T(C, i) = T(C, j) \Rightarrow S(C, i) = S(C, j).$$

2)- S et T sont équivalentes pour C, et nous noteront $S \equiv_C T$ si

$$S \leq_C T \text{ et } T \leq_C S.$$

3)- La signature S est auto-duale si $S \equiv S^{\perp}$.

Nous avons toujours $S \leq_C S \times T$ et le produit $S \times S^{\perp}$ est toujours auto-dual.

Signature totalement discriminante 4.5

Commençons maintenant pour donner une approche à la réponse de la deuxième question posée au début de ce chapitre ; c'est à dire déterminer la permutation qui définie l'équivalence de codes donnés sachant qu'il sont équivalents.

Définition 4.5.1

Une signature S est *totalement discriminante* pour un code C de longueur n si

$$S(C, i) \neq S(C, j) \text{ pour tout } i, j \text{ distincts de } I.$$

Cela veut dire que l'application $S_C : i \rightarrow S(C, i)$ est injective, pour C donné. Nous dirons alors que les positions $i \in I$ sont discriminées.

Il est nature de poser les questions suivantes : Si S est une signature totalement discriminante pour un code C, quelle *(s)* propriété *(s)* possède t-il le code C ? et si C' est un code équivalent à C, S est-elle totalement discriminante pour C' ? .

La réponse à ces question est apportée par la proposition suivante :

Proposition 4.5.2

Soit C un code de longueur n, S est une signature totalement discriminante pour C ; alors :

1) le groupe de permutation de C est trivial.

2) la signature S est totalement discriminante pour tout code C' équivalent à C.

Preuve

Soit *S* une signature totalement discriminante pour *C*.

1) soit σ une permutation de *Perm (C)* , alors :

$$\forall i \in I ;\ S (C , i) = S (\sigma(C) , \sigma(i))$$
$$= S (C , \sigma(i))$$

cela entraîne que $i = \sigma(i)$, pour tout $i \in I$.

2) Soit C' un code équivalent à *C* tel que $C' = \pi(C)$; alors pour *i* et *j* de *I* nous avons :

$$S (C' , i) = S (C' , j) \Leftrightarrow S (\pi^{-1} (C') , \pi^{-1} (i)) = S (\pi^{-1} (C'), \pi^{-1} (j))$$
$$\Leftrightarrow S (C , \pi^{-1} (i)) = S (C , \pi^{-1} (j))$$
$$\Leftrightarrow \pi^{-1} (i) = \pi^{-1} (j)$$

Car *S* est totalement discriminante pour *C* .

Donc : $S (C', i) = S (C', j) \Leftrightarrow i = j$ et *S* est totalement discriminante pour C' *c.q.f.d.*

Etant donnés deux codes équivalents *C* et C' tels que $C' = \sigma (C)$, et une signature totalement discriminante *S*, nous pouvons facilement calculer la permutation σ

Proposition 4.5.3

Etant donnés deux codes équivalents C et C' tels que $C' = \sigma (C)$, et une signature S totalement discriminante pour C , alors σ est bien déterminée et elle est unique.

Preuve

Soient *C* et C' deux codes équivalents de longueur *n*, tels que $C' = \sigma (C)$, et soit *S* une signature totalement discriminante pour *C*, donc elle l'est aussi pour C'.

Soient *A*, *B* respectivement les ensembles des valeurs $S (C , i)$ et $S (C', j)$, pour tout $(i , j) \in I \times I$;

C'est à dire :

$$A = \{ S (C , i) / i \in I \}$$
$$B = \{ S (C', j) / j \in I \}.$$

Comme S est totalement discriminante pour C et pour C', alors chaque ensemble A ou B est de cardinal égal à n, et comme $C' = \sigma(C)$, nous avons d'une part :

$$\forall i \in I ; S(C, i) = S(\sigma(C), \sigma(i)) = S(C', \sigma(i)) \quad (1)$$

et comme C et C' sont équivalents, A et B sont égaux :

$$\forall i \in I ; \exists j \in I \text{ tels que : } S(C, i) = S(C', j) \quad (2)$$

nous pouvons conclure de (1) et de (2) que :

$$\forall i \in I ; \exists j \in I \text{ tels que : } S(C', \sigma(i)) = S(C', j).$$

Du fait que S est totalement discriminantes pour C', nous pouvons affirmer que $\sigma(i) = j$. D'après la proposition 4.5.2, le groupe de permutation de C est trivial, ce qui implique l'unicité de σ.

Remarque importante

La méthode ainsi présentée dans la preuve est appelée algorithme de séparation des supports dû à NICOLAS SENDRIER, il permet de calculer la permutation σ sous les conditions de la proposition 4.5.3.

Pour décider si C et C' sont équivalents, l'algorithme apporte la réponse si $|A| = |B| \leq n$ (pour plus de détail voir [NS 96] et [NS 02]).

Exemple 4.5.4

Considérons les deux codes suivants sur F_2 :

$$C = \{ 1110 , 0111 , 1010 \}$$

$$C' = \{ 0011 , 1011 , 1101 \}$$

Comme invariant, nous prenons le polynôme énumérateur des poids, c'est à dire

$$v(C) = W_C(x) = \sum_{c \in C} x^{\omega(c)} = W(C)$$

Et comme signature, nous prenons la signature suivante :

$$S(C, i) = v(C) = W(C_i)$$

$$C_1 = \{ 0110 , 0111 , 0010 \} \rightarrow W(C_1) = x + x^2 + x^3$$

$$C_2 = \{ 1010 , 0011 \} \rightarrow W(C_2) = 2x^2$$

$$C_3 = \{ 1100 , 0101 , 1000 \} \rightarrow W(C_3) = x + 2x^2$$

$$C_4 = \{ 1110 , 0110 , 1010 \} \rightarrow W(C_4) = 2x^2 + x^3$$

Nous voyons que la signature *S* et totalement discriminante pour *C*. Pour le code C' nous avons :

$$C'_1 = \{\ 0011\ ,\ 0101\ \} \rightarrow W(C'_1) = 2x^2$$

$$C'_2 = \{\ 0011\ ,\ 1011\ ,\ 1001\ \} \rightarrow W(C'_2) = 2x^2 + x^3$$

$$C'_3 = \{\ 0001\ ,\ 1001\ ,\ 1101\ \} \rightarrow W(C'_3) = x + x^2 + x^3$$

$$C'_4 = \{\ 0010\ ,\ 1010\ ,\ 1100\ \} \rightarrow W(C'_4) = x + 2x^2\ .$$

Remarquons que :

$$W(C_1) = W(C'_3)$$

$$W(C_2) = W(C'_1)$$

$$W(C_3) = W(C'_4)$$

$$W(C_4) = W(C'_2)$$

Nous pouvons donc obtenir immédiatement la permutation σ telle que $C' = \sigma(C)$.

$$\sigma(1) = 3\ ,\ \sigma(2) = 1\ ,\ \sigma(3) = 4\ ,\ \sigma(4) = 2\ .$$

C'est à dire que $\sigma = (1342)$

Remarque 4.5.5

la proposition 4.5.3. nous permet de calculer immédiatement σ si la signature considérée est totalement discriminante. Mais en général une telle signature n'est disponible à un seul coup d'œil. Notons encore que si le groupe de permutation du code *C* (et bien sûr C') n'est pas trivial, une telle signature n'existe pas, ces deux remarques nous ont poussé à étudier un cas particulier où nous pouvons calculer la permutation σ si cette signature qui n'est pas totalement discriminante vérifie une certaine condition que nous avons supposée.

Rappels et notations 4.5.6

Dans tout ce qui suit *C* et C' représentent deux codes équivalents de longueur *n* tels que $\sigma(C) = C'$, et *S* une signature définie comme suite :

$$S(C, i) = v(C_i)$$ où v est un invariant.

Rappelons que *S* est totalement discriminante pour *C*, veut dire que l'application $i \rightarrow S(C, i)$, avec *C* fixé , est injective. Supposons dans la suite que *S* n'est pas discriminante en *s* positions seulement, c'est à dire que :

$$S(C, i_1) = S(C, i_2) = \ldots = S(C, i_s) \text{ avec } i_1, i_2, \ldots I_s \in I .$$

et $2 \leq s \leq n-2$.

Etudions si cette supposition, nous apporte des informations sur les indices i_1 , i_2,,…, i_s .

Partition associée à une signature 4.5.7

Si S est une signature et C est un code de longueur n. Soit la relation R_C définie sur I par :

$$\forall i, j \in I : i \; R_C \; j \Leftrightarrow S(C, i) = S(C, j)$$

Lemme 4.5.8

la relation R_C ainsi définie est une relation d'équivalence sur I et elle permet de définir une partition de I selon S et C .

Preuve

Il est évident que R_C est une relation d'équivalence sur I .

Pour $j \in I$; $\overline{j}$ désigne la classe de j selon R_C .

C'est à dire ; $\overline{j} = \{ i \in I / \; i \; R_C \; j \}$.

Soient : $\overline{j}_1$, $\overline{j}_2$, …… , $\overline{j}_d$ les classes d'équivalence module R_C , alors :

$\{ \overline{j}_1 , \overline{j}_2 , \ldots\ldots , \overline{j}_d \}$ est une partition de I qui n'est autre que I / R_C .

Alors ; si S n'est pas discriminante en s positions pour C, les ensembles

$\{ i_1 , i_2 , . , i_s \}$ et $(\{\alpha\})_{\alpha \in I - \{i_1, \ldots, i_s\}}$ forment une partition de I selon S et C . *c.q.f.d*

Si $C' = \sigma(C)$, que peut être la partition de I selon S et C'? la réponse à cette question est donnée par la proposition suivante :

Proposition 4.5.9

Si $\{ \overline{j}_1 , \overline{j}_2 , \ldots\ldots , \overline{j}_d \}$ est une partition de I selon C et S et si $C' = \sigma(C)$, alors la partition de I selon C' et S est :

$$\{ \overline{\sigma(j_1)}, \overline{\sigma(j_2)}, \ldots\ldots, \overline{\sigma(j_d)} \} .$$

Preuve

La proposition vient du fait que si $i , j \in I$, alors :

$$i \; R_C \; j \Leftrightarrow S (C , i) = S (C , j)$$

$$\Leftrightarrow S (\sigma(C) , \sigma(i)) = S (\sigma(C) , \sigma(j))$$

$$\Leftrightarrow S (C' , \sigma(i)) = S (C' , \sigma(j))$$

$$\Leftrightarrow \sigma(i) \; R_C \; \sigma(j)$$

Donc nous pouvons conclure que si $\bar{i}$ est une classe suivant R_C ; alors $\overline{\sigma(i)}$ est une classe selon $R_{C'}$, et réciproquement. Ce qui permet de dire que la partition $\{ \bar{j}_1 , \bar{j}_2 , \ldots\ldots , \bar{j}_d \}$ selon C et S, est en bijection avec la partition $\{ \overline{\sigma(j_1)}, \overline{\sigma(j_2)}, \ldots\ldots, \overline{\sigma(j_d)} \}$ selon C' et S .

Proposition 4.5.10

Si $C' = \sigma(C)$ et S n'est pas discriminante en s positions $i_1 , i_2 , \ldots , i_s$ de I pour C, alors S n'est pas discriminante en s positions $\sigma(i_1), \sigma(i_2) , \ldots , \sigma(i_s)$ de I pour C'.

Preuve

Par hypothèse

$$S (C , i_1) = S (C , i_2) = \ldots\ldots = C (C , i_s) \qquad (1)$$

pour $i_1 , i_2 , \ldots , i_s$ differents de I et $2 \leq i \leq n-2$, et pour $i , j \in I \setminus \{ i_1 , i_2 , \ldots , i_s \}$ et $i \neq j$;

$$S (C, i) \neq S (C, j) \qquad (2)$$

Comme S est une signature et $\sigma(C) = C'$.

$$S (C , i_1) = S (\sigma(C) , \sigma(i_1)) = S (C' , \sigma(i_1))$$

$$S (C , i_2) = S (\sigma(C) , \sigma(i_2)) = S (C' , \sigma(i_2))$$

……..

…….

$$S (C , i_s) = S (\sigma(C) , \sigma(i_s)) = S (C' , \sigma(i_s))$$

En se referant à (1) nous concluons que :

$S (C', \sigma(i_1)) = S (C', \sigma(i_2)) = \ldots\ldots = S (C', \sigma(i_s))$ pour $i_1, i_2 , \ldots, i_s$, different de I et $2 \leq s \leq n - 2$.

Il est facile de remarquer que pour tout l, $t \in \{ 2, \ldots , n - 2 \}$ différents ; $\sigma(i_t) \neq \sigma(i_l)$ car σ est une bijection .

Soit $k, m \in I \setminus \{ \sigma(i_1), \sigma(i_2), \ldots, \sigma(i_s) \}$ avec $k \neq m$.

Comme σ est une bijection, il existe i, j unique de $I \setminus \{ i_1, i_2, \ldots, i_s \}$ avec $i \neq j$, tels que $\sigma(i) = k$ et $\sigma(j) = m$.

Alors :

$$S(C', k) = S(\sigma(C), \sigma(i)) = S(C, i).$$

$$S(C', m) = S(\sigma(C), \sigma(j)) = S(C, j).$$

Par hypothèse (2) nous pouvons conclure $S(C', k) \neq S(C', m)$. *c.q.f.d*

Tirons plus d'information sur les éléments $i_1, i_2, .., i_s$; si $\pi \in perm(C)$, alors :

$$S(C, i) = S(C, \pi(i))$$

Cela équivaut à dire $i \; R_C \; \pi(i)$. pour tout $i \in I$.

Si de plus $S(C, i) = S(C, j)$ nous avons :

$$S(C, \pi(i)) = S(C, \pi(j)).$$

C'est à dire si $i \in \{ i_1, i_2, \ldots, i_s \}$ alors $\pi(i) \in \{ i_1, i_2, \ldots, i_s \}$ pour tout $\pi \in Perm(C)$. Donc nous pouvons annoncer la proposition suivante :

Proposition 4.5.11

Si S est une signature non discriminante en $i_1, i_2, \ldots, i_s$ de I pour C ; alors $\{ i_1, i_2, \ldots, i_s \}$ est stable par l'action du groupe de permutations de C.

Passons maintenant à résoudre le problème de détermination de σ telle que $C' = \sigma(C)$ en se donnant une signature S qui n'est pas discriminante

en $i_1, i_2, \ldots, i_s$ de I pour C, avec $2 \leq s \leq n-2$, il y a donc aux moins deux éléments i, j de I qui sont discriminés par S; c'est à dire $S(C, i) \neq S(C, j)$

avec $i \neq j$.

Puisque $C' = \sigma(C)$ nous avons :

$$S(C, i_1) = S(C, i_2) = \ldots\ldots = S(C, i_s)$$
$$= S(C', \sigma(i_1)) = S(C', \sigma(i_2)) = \ldots\ldots = S(C', \sigma(i_s))$$

et d'après la proposition 4.5.10 :

$$S(C, j_1) = S(C, j_2) = \ldots\ldots = S(C, j_s) \text{ où}$$
$$\sigma\{ i_1, i_2, \ldots, i_s \} = \{ \sigma(i_1), \sigma(i_2), \ldots, \sigma(i_s) \}$$
$$= \{ j_1, j_2, \ldots, j_s \}.$$

Mais pour tout i_t, $1 \leq t \leq s$, quel est l'image $\sigma(i_t)$ correspondant parmi les éléments de $\{ j_1, j_2, \ldots, j_s \}$.

Nous rappelons que pour cette étude la signature utilisée est définie par :

$$S(C, i) = \nu(C_i) \text{ où } \nu \text{ est un invariant.}$$

Choisissons $i \in I \setminus \{ i_1, i_2, \ldots, i_s \}$ une position discriminée par S.

de l'égalité $S(C, i) = S(C', k)$ on peut tirer $\sigma(i) = k$.

Soit pour i fixé ; l'application φ_C de $\{ i_1, i_2, \ldots, i_s \}$ dans E définie par:

$$\varphi_C(\alpha) = \nu(C_{\{i,\alpha\}})$$

rappelons que puisque ν est un invariant, alors pour toute $\pi \in S_n$:

$$\nu(\pi(C_{\{i,\alpha\}})) = \nu(C_{\{i,\alpha\}}) \quad \text{d'une part}$$

$$= \nu(\pi(C)_{\{\pi(i),\pi(\alpha)\}}) \quad \text{d'autre part}$$

Nous pouvons donc annoncer la proposition suivante :

Proposition 4.5.12

Si l'application φ_C est injective, alors les images de $i_1, i_2, \ldots, i_s$ par σ peuvent être déterminés immédiatement avec $C' = \sigma(C)$.

Preuve

Soit $\sigma \in S_n$ telle que $C' = \sigma(C)$.

φ est injective, alors :

$$\forall \alpha, \beta \in \{ i_1, i_2, \ldots, i_s \}, \alpha \neq \beta \Leftrightarrow \varphi_C(\alpha) \neq \varphi_C(\beta).$$

Montrons d'abord que l'application $\varphi_{C'}$ construite aussi comme φ_C, pour $k = \sigma(i)$ fixé, est aussi injective.

Soient $a, b \in \{ j_1, j_2, \ldots, j_s \}$, avec $a \neq b$.

$$\varphi_{C'}(a) = \nu(C'_{\{k,a\}}) = \nu(\sigma(C)_{\{\sigma(i),\sigma(il)\}})$$

$$= \nu(\sigma(C_{\{i,l\}})) = \nu(C_{\{i,l\}})$$

$$= \varphi_C(i_l).$$

de même $\varphi_{C'}(b) = \varphi_C(i_h)$; avec $i_l, i_h \in \{ i_1, i_2, \ldots, i_s \}$.

$$a \neq b \Leftrightarrow i_l \neq i_h$$

$$\Leftrightarrow \varphi_C(i_l) \neq \varphi_C(i_l)$$

$$\Leftrightarrow \varphi_{C'}(a) \neq \varphi_{C'}(b)$$

Calculons successivement :

$\varphi_C(i_1)$, $\varphi_C(i_2)$, … , $\varphi_C(i_s)$ d'une part et $\varphi_{C'}(j_1)$, $\varphi_{C'}(j_2)$, … , $\varphi_{C'}(j_s)$ d'autre part.

les ensembles images $\varphi_C\{ i_1 , i_2 , \ldots , i_s \}$ et $\varphi_{C'}\{ j_1 , j_2 , \ldots , j_s \}$ sont égaux , car comme

$C' = \sigma(C)$ nous avons :

$$v(C_{\{i,\alpha\}}) = v(C'_{\{\sigma(i),\sigma(\alpha)\}})$$

$$= v(C'_{\{k,\sigma(\alpha)\}}) .$$

et cela pour tout $\alpha \in \{ i_1 , i_2 , \ldots , i_s \}$ et $C' = \sigma(C)$.

pour chaque $i_t \in \{ i_1 , i_2 , \ldots , i_s \}$; il existe un seul j_h de $\{ j_1 , j_2 , \ldots , j_s \}$ tel que :

$\varphi_C(i_t) = \varphi_{C'}(j_h)$ d'où $\sigma\{ i , i_t \} = \{ k , j_h \}$.

puisque $\sigma(i) = k$, on pourra conclure que $\sigma(i_t) = j_h$. *c.q.f.d*

Exemple 4.5.13

Considérons les deux codes C et C' définis par :

$$C = \{ 01101 , 01011, 01110 , 10101 , 11110 \}$$

$$C' = \{ 10101 , 00111 , 10011 , 11100, 11011 \}$$

Prenons comme signature :

$S(C , i) = v(C_i) = W(C_i)$, le polynôme énumérateur des poids de C_i pour $i = 1 , 2 , \ldots ,5$.

Alors :

$$C_1 = \{ 01101 , 01011, 01110 , 00101\} \rightarrow x^2 + 3x^3$$

$$C_2 = \{ 00101 , 00011, 00110 , 10101 , 10110 \} \rightarrow 3x^2 + 2x^3$$

$$C_3 = \{ 01001 , 01011, 01010 , 10001 , 11010 \} \rightarrow 3x^2 + 2x^3$$

$$C_4 = \{ 01101 , 01001, 01100 , 10101 , 11100 \} \rightarrow 2x^2 + 3x^3$$

$$C_5 = \{ 01100 , 01010, 01110 , 10100 , 11110 \} \rightarrow 3x^2 + x^3 + x^4$$

Remarquons que les positions 2 et 3 ne peut être discriminées.

Pour C' :

$$C'_1 = \{ 00101 , 00111 , 00011 , 01100, 01011 \} \rightarrow 3x^2 + 2x^3$$

$$C'_2 = \{ 10101 , 00111 , 10011 , 10100 \} \rightarrow x^2 + 3x^3$$

$$C'_3 = \{ 10001 , 00011 , 10011 , 11000, 11011 \} \rightarrow 3x^2 + x^3 + x^4$$

$$C'_4 = \{ 10101 , 00101 , 10001 , 11100, 11001 \} \rightarrow 2x^2 + 3x^3$$

$$C'_5 = \{\ 10100\ ,\ 00110\ ,\ 10010\ ,\ 11100,\ 11010\ \} \rightarrow 3x^2 + 2x^3$$

Remarquons que pour C' , les positions 1 et 5 , ne peuvent être discriminées .

D'après:

$S\ (\ C\ ,\ 1\ ) = S\ (C'\ ,\ 2\ )$ *nous tirons* $\sigma(\ 1\ ) = 2$.

$S\ (\ C\ ,\ 4\ ) = S\ (C'\ ,\ 4\ )$ *nous tirons* $\sigma(\ 4\ ) = 4$.

$S\ (\ C\ ,\ 5\ ) = S\ (C'\ ,\ 3\ )$ *nous tirons* $\sigma(\ 5\ ) = 3$.

Soient les applications :

$\varphi_C\ (\ \alpha) = W\ (\ C_{\{\ 1,\ \alpha\ \}}\ )$, *pour* $\alpha \in \{\ 2\ ,\ 3\ \}$.

$\varphi_{C'}\ (\ \beta) = W\ (\ C'_{\{\ 2,\ \beta\ \}}\ )$, *pour* $\beta \in \{\ 1\ ,\ 5\ \}$.

alors :

$C_{\{\ 1,\ 2\ \}} = \{\ 00101\ ,\ 00011\ ,\ 00110\ \} \rightarrow \varphi_C\ (2\ ) = 3x^2$

$C_{\{\ 1,\ 3\ \}} = \{\ 01001\ ,\ 01011\ ,\ 01010\ ,\ 00001\ \} \rightarrow \varphi_C\ (3\ ) = x + 2x^2 + x^3$

$C'_{\{\ 2\ ,\ 1\ \}} = \{\ 00101\ ,\ 00111\ ,\ 00011,00100\ \} \rightarrow \varphi_{C'}(1\ ) = x + 2x^2 + x^3$

$C'_{\{\ 2\ ,\ 5\ \}} = \{\ 10100\ ,\ 00110\ ,\ 10010\ \} \rightarrow \varphi_{C'}(5\ ) = 3x^2$

donc φ_C et $\varphi_{C'}$ sont injectives et de $\varphi_C\ (2\ ) = \varphi_{C'}(5\ )$, nous tirons $\sigma(2\ ) = 5$,

et de $\varphi_C\ (3\ ) = \varphi_{C'}(1)$, nous tirons $\sigma(3\ ) = 1$..

ainsi la permutation σ est :

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 1 & 4 & 3 \end{pmatrix}$$

CONCLUSIONS & PERSPECTIVES

Nous avons présenté dans ce travail une étude sur l'équivalence par permutation de deux codes de même longueur sur un corps fini , elle est basée sur les notions d'invariant et de signature introduites par NICOLAS Sendrier dans son article[NS 96].

Notre travail était d'étudier les propriétés de deux codes équivalents au point de vue algébrique : isomorphisme de groupes de permutation…etc, d'une part, et d'autre part de déterminer la permutation qui définie l'équivalence si la signature n'est pas totalement discriminante sous une certaine condition .

L'idée de ce travail fait suite à l'article de NICOLAS Sendrier [NS 96] et à sa mémoire d'habilitation à diriger des recherches [NS 2002] ou il détermine la permutation lorsque la signature est totalement discriminante et propose un raffinement d'une signature non totalement discriminante en utilisant comme invariant le polynôme énumérateur des poids du Hull des codes . Il est à signaler que le problème d'équivalence des codes demeure encore ouvert.

Questions :

- L'invariant proposé par Sendrier est le polynôme énumérateur des poids est-il possible de trouver un autre invariant qui serait mieux efficace pour les codes ou au moins pour certaines classes de codes ?
- Est il possible de construire une signature qui serait totalement discriminante pour les codes ou au moins pour certaines classes de codes?
- Si le groupe de permutations d'un code C n'est pas trivial, comment rendre ce travail de détermination de la permutation de la permutation $\sigma \in S_n$; telle que $\sigma(C) = C'$ applicable ?

Enfin , nous pensons qu'on peut commencer par construire un code $L(L')$ à partir de $C(C')$ dont le groupe de permutations *perm (L)* est le groupe quotient $perm(C)/perm(C)$ qui est trivial à condition que les codes L et L' ainsi construits sont équivalents par permutation unique π ssi C et C' sont équivalents par permutation σ ou on peut déduire σ à partir de π.

BIBLIOGRAPHIE

[AD.79] : Alain Bouvier, Denis Richard. *Groupes : observation ,théorie , pratique* : Hermann, deuxième édition 1979.

[AO.02] : Ayoub Otmani. *Codes cortex et construction de codes auto –duaux optimaux,* Thèse de doctorat 2002 .

[CF.GA91] : Cherbonnier F et Fermigier S. *Codes correcteurs d'erreurs* Gazette des mathématiques N° 48, avril 1991.

[CPR02] : Costle M, Paugam A, Quarez R. *Codes correcteurs. P*réparation à l'agrégation Mathématiques. Université de Rennes 1. juin 2002

[DJM01] : Dany-jack Mercier. *Utilisation de l'algèbre dans les systèmes d'informations.* IUFM des Antilles et de GUYANE mai 2001.

[GSR95] : G. Lachaud et S. Vladut. *Les codes correcteurs d'erreurs*, La recherche 278 juillet, août 1995, volume 26.

[HCR95] : Henri Cohen. *Les nombres premiers,* La recherche 278 juillet, août 1995 volume 26.

[JV95] : J. Velu. *Méthodes mathématiques pour l'informatique.* Dunod1995.

[LJG73] : Larry Joel Goldstein. *Abstarct algebra* : *A first course* , Prentice Hall 1973.

[MS 77]: FJ. Macwilliams and NJA Sloane. *The théory of error-correting codes*. North. Holland 1977.

[NS96] : Nicolas Sendrier. *Un algorithme pour trouver la permutation entre deux codes binaires équivalents*. INRIA-Rocquencout, Rapport de recherche N° 2853. Avril 1996.

[NS02] : Nicolas Sendrier. *Cryptosystèmes à clé publique basés sur les codes correcteurs d'erreurs.* INRIA–Rocquencourt. Mémoire d'habilitation à diriger des recherches, mars 2002.

[PL01] : Pierre Loindreau. *Etude et optimisation de cryptosystèmes à clé publique fondés sur la théorie des codes correcteurs*. Thèse de doctorat en sciences. Ecole polytechnique, 2001.

[PS71] : Pierre Samuel, *Théorie algébrique des nombres*. Hermann , deuxième édition 1971.