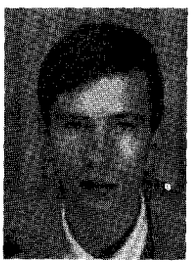
Algorithme de décodage des codes de Reed et Muller

Algorithm intended for decoding Reed-Muller codes



Pierre BONNEAU

INRIA, BP n° 105, 78153 LE CHESNAY CEDEX

Mon activité professionnelle est consacrée à la recherche sur la structure des codes, linéaires ou non. Les problèmes d'équivalence et de groupes, les liens avec la géométrie sont particulièrement étudiés dans ma principale réalisation : ma thèse de troisième cycle : Codes et Combinatoire. J'y classifie un certain type de code parfait 1-correcteur. Actuellement je poursuis ces études en essayant d'en trouver des applications, notamment au décodage.

RÉSUMÉ

Nous décrivons un algorithme de décodage des R (r, m) par un procédé récurrent. Nous utilisons les propriétés géométrico-combinatoires de ces codes et leur invariance par un groupe 2-abélien élémentaire. Nous établissons des liens avec les fonctions booléennes, les codes-idéaux d'algèbre de groupe modulaire, les transformations de Fourier rapides et les configurations combinatoires déduites des géométries finies.

MOTS CLÉS

Algorithme, code de Reed et Muller, transformation de Fourier rapide, groupe abélien élémentaire, fonction booléenne.

SUMMARY

We describe an algorithm intended for decoding the R (r, m) *by mean of an induction process. We use geometrico-combinatorial properties of these code and their stability under the action of a 2-elementary abelian group. We establish connexions with boolean functions, codes which are ideals in a modular group algebra, fast Fourier transforms, and combinatorial designs which are deduced from finite geometries.*

KEY WORDS

Algorithm, Reed-Muller code, fast Fourier transform, elementary abelian group, boolean function.

TABLE DES MATIÈRES

Introduction

Les codes de Reed et Muller sont très utilisés et très connus. Leur grand intérêt provient du fait qu'ils sont faciles à décoder. Le propos du présent article est de généraliser et d'améliorer des algorithmes classiques.

Le paragraphe 1 est consacré à la présentation d'un algorithme de décodage de R (1, *m*). Cet algorithme exploite une définition géométrique de ce code et son invariance sous un groupe 2-abélien élémentaire. Il se termine plus rapidement que celui de [1].

Nous montrons au paragraphe 2 le lien entre les idées du paragraphe 1 et la transformation de Fourier rapide. Cette transformation est présentée sous un nouveau jour.

L'algorithme du paragraphe 1 peut servir d'étape initiale au procédé de récurrence décrit au paragraphe 3 : nous montrons comment déduire d'un algorithme de décodage de R ($r-1$, m) un algorithme de décodage de R (r, m). Nous avons besoin du résultat du paragraphe 4 pour la réalisation pratique de ce procédé. Ce dernier paragraphe est consacré uniquement aux fonctions booléennes.

Enfin le paragraphe 5 synthétise ce qui précède et montre comment agencer les opérations de manière à exploiter des simplifications.

1. Décodage de R (1, *m*)

Soient m un entier naturel, $m \geqq 2$; $\mathbb{F}_2$ un corps à deux éléments; $V = \mathbb{F}_2^m$; $(e_k)_{1 \leqq k \leqq m}$ une base de V pour sa structure naturelle de $\mathbb{F}_2$ espace vectoriel. Nous identifions les éléments de $\mathbb{F}_2^V$ aux sous-ensembles de V : un élément de $\mathbb{F}_2^V$ est une fonction booléenne $f : V \to \mathbb{F}_2$; elle est identifiée à $\{ v \in V, f(v) = 1 \}$. Le nombre d'éléments d'une partie X de V est noté $|X|$. C'est aussi le poids de l'élément de $\mathbb{F}_2^V$ (sa fonction caractéristique) auquel il est identifié. $\mathbb{F}_2^V$ étant muni de sa structure naturelle de $\mathbb{F}_2$ espace vectoriel, soit τ_k ($1 \leqq k \leqq m$) l'automorphisme de $\mathbb{F}_2^V$ défini par $X\tau_k = \{ x + e_k, x \in X \}$. Les τ_k préservent le poids.

R (1, m) est le sous-ensemble de $\mathbb{F}_2^V$ dont les éléments sont :

– le mot 0 dont toutes les composantes sont nulles : $|0| = 0$;

– le mot 1 dont toutes les composantes sont égales à 1. $|1| = 2^m$;

– H parcourant l'ensemble des hyperplans vectoriels de H, les mots encore notés H d'après l'identification dont il a été fait mention ci-dessus. $|H| = 2^{m-1}$;

– les mots de la forme $H' = 1 + H$, H décrivant les mots définis précédemment. H' s'identifie au complémentaire dans V de l'hyperplan vectoriel H. $|H'| = 2^{m-1}$.

Si $C \in R(1, m)$, $C \neq 0$, $C \neq 1$, C est un hyperplan affine de V. Sa direction, notée C_d est C si $0 \in C$, $1 + C$ si $0 \notin C$.

R (1, m) est un sous-espace vectoriel de $\mathbb{F}_2^V$ laissé globalement stable par les τ_k. En effet soit $C \in R(1, m)$, $C \neq 0$ et $C \neq 1$:

$C\tau_k = C$ si $e_k \in C_d$;

$C\tau_k = 1 + C$ si $e_k \notin C_d$.

Comme le lecteur pourra s'en convaincre aisément, nous nous sommes placés dans un cadre très proche de celui du paragraphe 4 du chapitre 13 de [1]. Il transposera en munissant EG (m, 2) d'une structure d'espace vectoriel compatible avec sa structure d'espace affine. Il pourra puiser dans l'ouvrage précité les connaissances nécessitées par la compréhension du présent article.

Nous sommes maintenant en mesure d'aborder le cœur de notre sujet. Les paragraphes 1 et 2 démontrent la validité du troisième. Enfin le paragraphe 4 fait le lien avec la description classique de R (1, m) en termes de fonctions booléennes.

1. Étudions la restriction à R (1, m) des endomorphismes :

$$\varphi_k : \mathbb{F}_2^V \to \mathbb{F}_2^V, \qquad X \mapsto X + X\tau_k.$$

(*a*) $\varphi_k(0) = \varphi_k(1) = 0$ pour tout k.

(*b*) Si $C \neq 0$ et $C \neq 1$, il existe au moins un entier j, $1 \leqq j \leqq m$, tel que $e_j \notin C_d$. $\{ 1, 2, ..., m \}$ est donc réunion de l'ensemble I des i tels que $e_i \in C_d$ et de l'ensemble non vide J des j tels que $e_j \notin C_d$. Si $i \in I$, alors $\varphi_i(C) = 0$; si $j \in J$, alors $\varphi_j(C) = 1$.

2. Soit R un mot reçu. $R = C + E$ où C est le mot de R (1, m) transmis et E un vecteur d'erreur.

Nous supposerons dans toute la suite que $|E| < 2^{m-2}$.

Remarquons que :

$$|\varphi_k(C)|-|\varphi_k(E)|\leqq|\varphi_k(R)|\leqq|\varphi_k(C)|+|\varphi_k(E)|.$$

Puisque :

$$|\varphi_k(E)|\leqq|E|+|E\tau_k|<2^{m-1},$$
$$|\varphi_k(C)|-2^{m-1}<|\varphi_k(R)|<|\varphi_k(C)|+2^{m-1}.$$

D'après 1 deux cas seulement peuvent se produire :

Premier cas :

$$|\varphi_k(C)|=0, \quad \text{alors } |\varphi_k(R)|<2^{m-1}.$$

Second cas :

$$|\varphi_k(C)|=2^m, \quad \text{alors } |\varphi_k(R)|>2^{m-1}.$$

3. La considération de la famille $(|\Phi_k(R)|)_{1\leqq k\leqq m}$ suffit pour trouver C à l'addition du mot constant 1 près :

— si pour tout k, $|\varphi_k(R)|<2^{m-1}$, alors $C=0$ ou 1;

— s'il existe un j tel que $\varphi_j(R)\geqq 2^{m-1}$, alors $C\neq 0$, $C\neq 1$ et $\{1, 2, ..., m\}$ est réunion de l'ensemble I des i tels que $|\varphi_i(R)|<2^{m-1}$ et de l'ensemble non vide J des j tels que $|\varphi_j(R)|>2^{m-1}$. Si $i\in I$, alors $e_i\in C_d$; si $j\in J$, alors $e_j\in C_d$. La forme linéaire f qui a C_d pour noyau est donc déterminée par :

$$\begin{cases} f(e_i)=0 & \text{si } i\in I, \\ f(e_j)=1 & \text{si } j\in J. \end{cases}$$

4. Déterminons explicitement C_d dans le cas, particulièrement intéressant pour les applications, où $(e_k)_{1\leqq k\leqq m}$ est la base canonique de $\mathbb{F}_2^m$. Avec les notations de [1], paragraphe 2 du chapitre 13, C_d est l'ensemble des points où la fonction booléenne $1+\sum_{j\in J} v_j$ prend la valeur 1.

5. Pour terminer le décodage de R il suffit de calculer $|R+C_1|$ et $|R+1+C_1|$, C_1 désignant le mot de $R(1, m)$ autorisé par 3 et qui contient 0. Deux cas seulement peuvent se produire :

Premier cas :

$$|R+C_1|<2^{m-1}, \quad \text{alors } C=C_1.$$

Second cas :

$$|R+C_2|<2^{m-1}, \quad \text{alors } C=C_2.$$

6. Remarquons que pour calculer $\varphi_i(R)$, il suffit de se restreindre à l'hyperplan V_i engendré par les e_j, $j\neq i$. Les éléments de V_i' sont de la forme $v+e_i$ avec $v\in V_i$. La composante de $\varphi_i(R)$ en $v+e_i$ est égale à sa composante en v. Le calcul de $|\varphi_i(R)|$ présente une simplification analogue.

7. Comme me l'a fait remarqué Paul Camion [2], l'algorithme ci-dessus demande m fois 2^{m-1} additions dans $\mathbb{F}_2$ (pour calculer $\varphi_i(R)$), puis m fois 2^{m-1} additions dans $\mathbb{Z}$ (pour calculer $|\varphi_i(R)|$). Dans [1], chap. 14, §4, l'algorithme présenté demande également $m\,2^m$ opérations arithmétiques, mais exige 2^m-1 comparaisons pour trouver le coefficient de Fourier de valeur absolue maximale. Il semble donc, toujours selon [2], que notre gain le plus substantiel provienne du fait que notre algorithme se termine avec seulement m comparaisons [chaque $|\Phi_i(R)|$ est à comparer à 2^{m-1}].

8. Néanmoins l'algorithme présenté dans [1] a le mérite de décoder tous les mots reçus R qui atteignent en un point unique le minimum de leur distance à $R(1, m)$. L'algorithme ci-dessus ressemble beaucoup à celui de Reed ([1], chap. 13, § 6).

2. Un nouveau jour sur la transformée de Fourier rapide

Soit F une application $V\to\mathbb{R}$, V^X le dual de V, i. e. l'ensemble des applications linéaires $\Lambda : V\to\mathbb{F}_2$. Chaque Λ est déterminée de manière unique par son noyau, qui est un hyperplan si elle est non nulle.

La transformée de Fourier de F est l'application $\hat{F}$: $V^X\to\mathbb{R}$ définie par :

$$\hat{F}(\Lambda)=\sum_{v\in \operatorname{Ker}\Lambda} F(v)-\sum_{v\notin \operatorname{Ker}\Lambda} F(v).$$

Évidemment l'application $F\mapsto\hat{F}$ est linéaire. La considération de sa matrice relativement aux bases canoniques de $\mathbb{R}^V$ et $\mathbb{R}^{V^X}$ conduit, par décomposition en produit de Kronecker à l'algorithme de transformation de Fourier rapide qui n'exige que $m\,2^m$ opérations.

Nous présentons de manière différente le même algorithme, en montrant comment ramener le calcul de $\hat{F}$ aux calculs des transformées de deux applications définies sur un espace de dimension $m-1$. Ces deux applications sont des versions réelles des $\varphi_i(X)$ introduites ci-dessus.

Il s'agit de F_1 et F_1' définies sur l'espace V_1 engendré par les e_i, $i>1$, par :

$$F_1(v)=F(v)+F(v+e_1),$$
$$F_1'(v)=F(v)-F(v+e_1).$$

Notons V_1' le complémentaire de V_1 dans V. V_1 et V_1' sont échangés par la translation par e_1. Soit $\Lambda\in V^X$, $H=\operatorname{Ker}\Lambda$, $H'=V\setminus\operatorname{Ker}\Lambda$, Λ_1 la restriction de Λ à V :

$$\hat{F}(\Lambda)=\sum_{v\in H\cap V_1} F(v)$$
$$+\sum_{v\in H\cap V_1'} F(v)-\sum_{v\in H'\cap V_1} F(v)-\sum_{v\in H'\cap V_1'} F(v).$$

Premier cas : H contient e_1. Alors H et H′ sont invariants par la translation par e_1 :

$$\hat{F}(\Lambda)=\sum_{v\in H\cap V_1} F(v)+\sum_{v\in H\cap V_1} F(v+e_1) - \sum_{v\in H'\cap V_1} F(v)-\sum_{v\in H'\cap V_1} F(v+e_1).$$

Donc :

$$\hat{F}(\Lambda)=\hat{F}_1(\Lambda_1).$$

Deuxième cas : H ne contient pas e_1. Alors H et H′ sont échangés par la translation par e_1 :

$$\hat{F}(\Lambda)=\sum_{v\in H\cap V_1} F(v)+\sum_{v\in H'\cap V_1} F(v+e_1) - \sum_{v\in H'\cap V_1} F(v)-\sum_{v\in H'\cap V_1'} F(v).$$

Donc :

$$\hat{F}(\Lambda)=\hat{F}_1'(\Lambda_1).$$

3. Décodage de R (r, m), r quelconque

Au paragraphe 1 nous avons vu comment décoder R $(1, m)$. Par un procédé analogue, nous montrons comment ramener le décodage de R (r, m) à celui de R $(r-1, m)$, $r\geqq 2$. Comme au paragraphe 1 on calcule les $\varphi_i(R)$ pour un mot reçu R. Les $\varphi_i(C)$ sont des éléments de R $(r-1, m)$. Ceci résulte des travaux de Pascale Charpin [3]. $\varphi_i(C)$ correspond à l'élément $(1+X^{e_i})\,\tilde{C}$ de l'algèbre $\mathbb{F}_2[V]$, si $\tilde{C}$ est l'élément de cette algèbre qui correspond à R.

Si $R=C+E$ où $C\in R(r, m)$, $|E|<2^{m-r-1}$, alors $\varphi_i(R)=\varphi_i(C)+\varphi_i(E)$ où $|\varphi_i(E)|<2^{m-r}$. Le décodage de R $(r-1, m)$ nous donne donc les $\varphi_i(C)$.

Il nous reste donc à montrer comment, pour une fonction booléenne f, déduire f à l'addition de 1 près des $\varphi_i(f)$. Ce problème est résolu au paragraphe suivant.

Une fois que C est connu à l'addition de 1 près, la détermination de C s'achève *mutatis mutandis* comme au paragraphe 1, 6.

4. Reconstitution d'une fonction booléenne f à partir des $\varphi_i(f)$

Soit f une fonction booléenne. D'après [1], $f=\sum_{I\in P} v_I$ où P est un ensemble de parties de $\{1, ..., m\}$ qui caractérise f et où nous notons $v_I=\sum_{i\in I} v_i$. On voit facilement que pour chaque entier i, $1\leqq i\leqq m$,

$$\varphi_i(f)=\sum_{I\in P,\ i\in I} v_{I\setminus\{i\}}. \tag{4}$$

Donc l'expression de $\varphi_i(f)$ dans la base v_I nous fournit les éléments $I\in P$ qui contiennent i. Ainsi la donnée des $\varphi_i(f)$ détermine-t-elle f à l'addition de $v_\Phi=1$ près.

5. Synthèse et simplification

Pour réaliser le procédé de récurrence décrit au paragraphe 3, il est nécessaire de calculer des itérés d'ordre r au plus des φ_i en R. On remarque aisément que les φ_i permuttent et que $\varphi_i^2=0$. Donc nous devons calculer les $\varphi_J=\prod_{j\in J}\varphi_j$ pour les parties J de $\{1, ..., m\}$ ayant r éléments au plus. En itérant la formule (4) on obtient :

$$\varphi_J(f)=\sum_{I\in P,\ J\subset I} v_{I\setminus J}. \tag{5}$$

En particulier si l'on sait que $f\in R(r, m)$, P ne contient pas de partie ayant plus de r éléments, donc la connaissance des $\varphi_I(f)$ pour $|I|=r$, détermine les termes de degré r de f. Les termes de degré $r-1$ sont alors connus par l'application de (5) aux J telles que $|J|=r-1$, etc. (*cf.* [1], chap. 13, § 6).

Notons que si $|J|=r$, la valeur de $\varphi_J(f)$ en un point est analogue à la somme envisagée au théorème 14 de [1], chap. 13.

Mais rien ne nous oblige à considérer toutes ces valeurs : il suffit de le faire, d'après une idée analogue à celle du paragraphe 1, 6, seulement sur l'espace V_J engendré par les e_j, $j\notin J$. $\varphi_J(f)$ est alors connue sur tout V grâce à son invariance sous les applications $v\mapsto v+u$, $u\in V_{J'}$, J′ désignant le complémentaire de J. D'ailleurs c'est le poids des $\varphi_J(R)$ qui intervient de manière cruciale, et non pas leurs valeurs en certains points.

Ajoutons que nous évitons les 2^m dépouillements qu'exige ce dernier algorithme. En itérant r fois l'algorithme décrit au paragraphe 3, ils sont remplacés par C_m^r calculs des $|\varphi_J(R)|$, $|J|=r$, et autant de comparaisons de poids, ce qui nous donne C_m^{r-1} mots de R $(1, m)$ sous la forme de I4 et I5. A ces C_m^{r-1} mots exprimés dans la base (v_I), il est pratique d'appliquer la formule (4) pour obtenir des mots de R $(r-2, m)$.

Notons enfin que le procédé de 3 s'applique à n'importe quel algorithme de décodage de R $(r-1, m)$.

Conclusion

Nous avons réalisé le programme que nous nous étions fixé dans l'introduction. Mais les perspectives d'extension du présent travail sont nombreuses.

Les opérateurs φ_i ont joué un rôle fondamental dans le présent article. Ils laissent invariants tous les codes idéaux de l'algèbre du groupe V (*voir* [3]). En élaborant les idées que nous avons développé, on obtient certainement de bons algorithmes de décodage de codes idéaux d'une algèbre de groupe.

Les codes obtenus en prenant pour matrice génératrice une matrice d'incidence d'une géométrie finie peuvent certainement être décodés par des procédés proches de ceux que nous avons décrits, au moins si la géométrie en question admet un groupe approprié.

BIBLIOGRAPHIE

[1] F. J. MacWilliams et N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Third printing, 1981.

[2] P. Camion, Lettre personnelle.

[3] P. Charpin, Codes idéaux de certaines algèbres modulaires, *Thèse de 3e cycle*, Université Paris-VII, 1982.