TP 6 – CODES CORRECTEURS D'ERREURS

► **PRÉREQUIS** : Chapitre 5 du poly.

Nous allons utiliser dans ce TP le modèle suivant :

- On considère des messages sous forme d'une suite de bits;
- Les bits seront regroupés par paquets de k bits avec k une puissance de 2;
- Chaque bit a une probabilité non nulle d'être altéré (et donc inversé).

1 Bit de parité

Pour pouvoir corriger une éventuelle erreur dans un bloc de bits, on doit ajouter une information redondante. En voici un premier exemple.

1. (**Bit de parité**) Coder une fonction qui découpe un message en blocs de 8 bits et qui transforme chaque bloc de 8 bits en un bloc de 9 bits en ajoutant à la fin un bit de sorte que la somme des bits du nouveau bloc de 9 bits soit paire.
2. Tester votre fonction sur 01101011 et 00110101. Que se passe-t-il si un bit est modifié? Et si deux bits sont modifiés? Peut-on corriger lorsque l'on détecte une erreur?

On aimerait que notre algorithme ne rajoute pas trop d'informations au message initial, soit capable de détecter et corriger un nombre raisonnable d'erreurs mais aussi que les algorithmes de codage et de décodage soient efficaces.

2 Paramètres d'un code

Un bloc de k bits sera indifféremment appelé bloc, mot ou vecteur dans la suite. L'ensemble des mots de k bits n'est rien d'autre que $M_k = \mathbb{F}_2^k$. On parlera indifféremment de bits ou de lettres. Un mot de k bits sera noté $b_1b_2 \cdots b_k$ ou $(b_1 \ b_2 \ \cdots \ b_k)$. Un code correcteur de paramètre (k, n) est une application injective $\Phi : M_k \rightarrow M_n$ appelée **encodage**. Le paramètre k est la **dimension** du code et n sa **longueur**. L'image C de Φ est appelée **image** du code et les éléments de C sont les **mots du code**.

3. Pourquoi demander que la fonction d'encodage soit injective? Quelle relation existe entre k et n ?
4. Quelle est la dimension et la longueur de l'ajout du bit de parité?
5. (**Code de répétition pure**) Considérons l'application $\Phi : M_1 \rightarrow M_3$ définie par $\Phi(0) = 000$ et $\Phi(1) = 111$. Expliciter toutes les notions ci-dessus au cas de ce code.
6. Préciser ce que peuvent devenir les mots 000 et 111 après 0,1 et 2 erreurs. Parmi les mots trouvés, quels sont ceux qui sont des mots du code de répétition pure ci-dessus? Combien d'erreurs ce code peut-il détecter? Corriger?

On appelle **poinds de Hamming** de m et on note $\omega(m)$ le nombre de lettres non nulles de m . Pour deux mots m et n du code, on appelle **distance de Hamming** entre m et n , la quantité $\omega(m - n)$.

7. Coder une fonction distance de Hamming et une fonction poinds de Hamming. Tester vos fonctions sur 01100111 et avec les couples (0001001, 0101001) et (0000110, 0001100).

Soit Φ une fonction d'encodage d'image C . On appelle **capacité de détection** de Φ et on note e_d le plus grand nombre d'erreurs que Φ permet de détecter à coup sûr. On appelle **capacité de correction** de Φ et on note e_c le plus grand nombre d'erreurs que Φ permet de corriger à coup sûr. On appelle **distance minimale** de Φ et on note d_Φ la plus petite distance de Hamming non nulle entre deux mots du code.

10. Rappeler les relations entre ces différentes quantités.
11. Déterminer ces paramètres dans le cas du bit de parité et de la répétition pure.

3 Codes linéaires

Un code Φ de paramètres (k, n) est dit **linéaire** si l'injection Φ est une application linéaire. Autrement dit s'il existe une matrice $G \in \mathcal{M}_{k,n}(\mathbb{F}_2)$ de rang k telle que

$$\forall m \in M_k, \quad \Phi(m) = mG.$$

La matrice G est appelée **matrice génératrice** du code Φ .

12. Les codes du bit de parité et de répétition pure sont-ils linéaires? Si oui, préciser leur matrice génératrice.

13. Étudier le code linéaire ayant

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

comme matrice génératrice. Préciser en particulier ses paramètres, C et donner la liste des distances de Hamming, d_Φ ainsi que e_d et e_c .

14. (Bornes de Singleton et de Hamming) Soit Φ un code linéaire de paramètre (k, n) d'image C . Rappeler la borne de Singleton et démontrer la borne de Hamming suivante

$$\sum_{i=0}^{e_c} C_n^i \leq 2^{n-k}.$$

Un code pour lequel on a égalité dans la borne de Singleton est dit **MDS (Maximum Distance Separable)** tandis qu'un code pour lequel on a égalité dans le borne de Hamming est dit **parfait**.

15. Donner un minorant sur la longueur d'un code linéaire de dimension k détectant d erreurs.

16. Les codes bit de parité, répétition pure et de la question 13 sont-ils MDS? Parfaits?

17. Implémenter une fonction qui prend en argument la matrice génératrice d'un code linéaire et renvoie les paramètres du code d_Φ, e_d et e_c et si le code est parfait ou MDS. Tester votre fonction sur la matrice

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

Comparer avec les commandes préexistantes de Sage suivantes

Argument(s)	Commande	Sortie
$G \in \mathcal{M}_{k,n}(\mathbf{F}_q)$	LinearCode(G)	code de matrice génératrice G
code C	<code>C.generator_matrix()</code>	matrice génératrice de C
code C	<code>C.parity_check_matrix()</code>	matrice de contrôle de C
mot c de C	<code>c.hamming_weight()</code>	poids de Hamming de c
code C	<code>C.minimum_distance()</code>	distance minimale de C
$r, q \geq 1$	<code>codes_HammingCode(GF(q), r)</code>	code de Hamming de longueur $\frac{q^r-1}{q-1}$
$n \geq 1, P \in \mathbf{F}_q[X]$	<code>codes_CyclicCode(n, P)</code>	code cyclique $(P) \subseteq (\mathbf{F}_q)^n$

Tester la perfection des codes de répétition de longueur inférieure à 32. Faire une conjecture.

Un code Φ de paramètres (k, n) est dit **systématique** si pour tout $m \in M_k$, le mot m est un préfixe de $\Phi(m)$.

18. Montrer qu'un code linéaire de paramètre (k, n) est systématique si, et seulement si, sa matrice génératrice est de la forme $(I_k \ G')$ où $G' \in \mathcal{M}_{k,n-k}(\mathbf{F}_2)$.

19. Soit Φ un code systématique de paramètres (k, n) et de matrice génératrice $(I_k \ G')$. Rappeler pourquoi la matrice $({}^tG' \ I_{n-k})$ est une matrice de contrôle de Φ .

20. Montrer que la distance d_Φ pour un code linéaire systématique est le nombre minimal de colonnes linéairement dépendantes de H où H est une matrice de contrôle.

21. Implémenter une fonction qui prend en entrée un code linéaire systématique à partir de sa matrice génératrice et renvoie une matrice de contrôle. Donner les matrices de contrôles des codes bits de parité, répétition pure et de la question 13.

Écrire une fonction qui prend en entrée un code linéaire systématique à partir de sa matrice de contrôle et renvoie une matrice génératrice. Composer ces deux programmes renvoie-t-il la matrice d'origine?

Soit Φ un code de paramètres (k, n) et de matrice génératrice G et de contrôle H . On se fixe un mot x de longueur k . Le mot de code correspondant est $y = \Phi(x)$. S'il y a eu des erreurs durant la transmission, on reçoit un mot z . On appelle **mot erreur** associé à z le mot e tel que $z = y + e$. On rappelle que le **syndrome** de z est le mot $z^t H$. On voit alors immédiatement que $z^t H = e^t H$ et que le syndrome ne dépend que de l'erreur. L'ensemble des syndromes de z , noté S_z est appelé **classe littérale** de z . Le principe du décodage est alors le suivant :

- On reçoit le mot z transmis avec d'éventuelles erreurs.
- On calcule la syndrome $s = z^t H$ de z .
- Si $s = 0$, on ne détecte pas d'erreur¹ et on retourne $\Phi^{-1}(z)$.
- Sinon on recherche l'erreur e de plus petit poids possible² telle que $e^t H = s$.
- Si $\omega(e) \leq e_c$, on retourne $\Phi^{-1}(z + e)$.

1. Ce qui ne signifie pas pour autant qu'il n'y en a pas! Pensez au bit de parité!

2. Donc la plus probable!

- Sinon retourner "Impossible de corriger les erreurs".

Il nous reste à expliquer comment déterminer e .

22. Rappeler pourquoi chaque syndrome s est associé à au plus un mot $m \in \mathbb{F}_2^n$ de poids $\omega(m) \leq e_c$.
23. Que se passe-t-il dans le cas d'une seule erreur? Par exemple prendre le code de matrice génératrice

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

et décoder le mot 0101 en sachant qu'il y a eu au plus une erreur.

4 Décodage

Pour trouver e , on va construire une **table de décodage**. Soit Φ un code linéaire de paramètres (k, n) . L'ensemble des syndromes possibles est donc $\mathbb{F}_2^{n-k}$. Pour établir cette table de décodage, on liste tous les syndromes de Φ puis tous les mots z de $\mathbb{F}_2^n$ par poids croissant. Pour chaque z , on calcule alors $z^t H$ et au syndrome s on associe le premier mot z obtenu tel que $z^t H = s$. On arrête alors ce procédé dès qu'on a associé un mot à chaque syndrome et le mot associé à un syndrome est alors l'erreur de poids minimal donnant s .

24. Implémenter une fonction table de décodage et la tester sur le code bit de parité, répétition pure et le code de la question 13. Décoder alors respectivement les mots 101010100, 101 et 1100.

5 Codes de Hamming

Un **code de Hamming** est un code parfait de paramètres $(2^r - r - 1, 2^r - 1)$ pour $r \geq 2$ dont une matrice de contrôle est obtenu par n'importe quelle énumération en colonne de tous les mots de m bits non nuls³.

25. Étudier et implémenter le code de Hamming capable de corriger une erreur et de paramètres $r = 2$.
26. Étudier et implémenter le code de Hamming de matrice génératrice

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

et décoder le mot 1011101.

Les codes parfaits sont rares. Parmi eux on compte entre autres les codes de répétition de paramètre $(1, 2e_c + 1)$ et les codes de Hamming.

6 Codes cycliques

27. Rappeler pourquoi se donner un code cyclique C de longueur n revient à se donner un polynôme unitaire P divisant $X^n - 1$, que l'on appelle **polynôme générateur** de C , et tel que

$$C = \{ \pi(PQ) : \deg(Q) \leq n - \deg(P) - 1 \}$$

avec $\pi : \mathbb{F}_2[X] \rightarrow \mathbb{F}_2[X]/(X^n - 1)$ la surjection canonique. Donner une matrice génératrice du code.

28. Écrire un programme qui détermine si un code est cyclique. Quels sont les codes présentés plus haut qui sont cycliques? Donner le cas échéant les polynômes générateurs. Déterminer les codes cycliques de longueur 7 et en donner tous les paramètres.
29. Tester `codes_CyclicCode` avec $1, X - 1, \frac{X^n - 1}{X - 1}$. Que reconnaît-on? Que se passe-t-il avec un polynôme qui ne divise pas $X^n - 1$?
30. Trouver tous les polynômes irréductibles de degré 3 sur $\mathbb{F}_2$ et vérifier que les codes cycliques de longueur 7 qu'ils induisent sont équivalents. Comparer avec les codes cycliques induits par des polynômes réductibles de degré 3.

L'opération de codage est alors encore plus simple! Elle consiste à utiliser la bijection

$$\begin{cases} V & \longrightarrow C \\ Q & \longmapsto \pi(PQ) \end{cases}$$

avec V l'espace vectoriel de dimension $n - \deg(P)$ des polynômes de $\mathbb{F}_2[X]$ de degré inférieur à $n - \deg(P) - 1$. L'opération de codage consiste donc en pratique à une multiplication polynomiale modulo $X^n - 1$.

Le décodage se fait de façon analogue en déterminant le syndrome. Le syndrome d'un mot est alors défini comme le reste de la division euclidienne de ce mot par le polynôme générateur.

3. Le minitel codait notamment ses données avec un code de Hamming de paramètre $(120, 128)$, soit $r = 7$.

31. Donner une matrice de contrôle.

On a alors l'algorithme suivant lorsqu'on suppose que $z = m + e$ avec une erreur e qui comporte k (où k est la dimension du code) zéros d'affilée. On initialise $i = 0$.

- On détermine le syndrome s_i de $X^i z$ par division euclidienne.
- Si $\omega(s_i) \leq e_c$, alors l'erreur est donnée par $e_i = X^{n-i}(s_i, \mathbf{0})$ et on retourne $m = z - e_i$.
- Sinon, on pose $i = i + 1$ et on recommence jusqu'à ce que $i = n$.
- Si $i = n$, l'erreur ne peut pas être corrigée.

32. Justifier que l'algorithme ci-dessus fonctionne.

33. Implémenter et étudier un code cyclique de paramètres $(7, 15)$ et de polynôme générateur $X^8 + X^7 + X^6 + X^4 + 1$ et décoder le mot correspondant à $X^{11} + X^{10} + X^7 + X^6 + X^5 + X^4 + X^3 + X + 1$.

Les codes cycliques permettent de corriger e_c erreurs et d'en détecter $\deg(P)$ sous réserve qu'elles n'apparaissent pas n'importe où. Ces codes sont notamment utilisés dans certains codes barres, disques durs ou lors de vols spatiaux. Je vous renvoie aux codes BCH notamment.

7 Bonus : un tour de magie

On demande à un interlocuteur de penser à un nombre entre 0 et 15 et on lui pose les questions suivantes en autorisant au plus un mensonge :

- Est-ce que le nombre est supérieur ou égal à 8?
- Est-ce que le nombre est dans $\{4, 5, 6, 7, 12, 13, 14, 15\}$?
- Est-ce que le nombre est dans $\{2, 3, 6, 7, 10, 11, 14, 15\}$?
- Est-ce que le nombre est impair?
- Est-ce que le nombre est dans $\{1, 2, 4, 7, 9, 10, 12, 15\}$?
- Est-ce que le nombre est dans $\{1, 2, 5, 6, 8, 11, 12, 15\}$?
- Est-ce que le nombre est dans $\{1, 3, 4, 6, 8, 10, 13, 15\}$?

Le prestidigitateur note alors $m_i = 1$ si la réponse à la i -ème question est positive et $m_i = 0$ sinon et pose $m = (m_1, \dots, m_7) \in (\mathbf{F}_2)^7$. Il calcule alors les quantités suivantes

$$a = m_1 + m_3 + m_5 + m_7, \quad b = m_2 + m_3 + m_6 + m_7, \quad c = m_4 + m_5 + m_6 + m_7.$$

Soit k l'entier qui s'écrit $\overline{cba}$ en base 2. Si $k = 0$, alors il n'y a pas de mensonge. Sinon, k indique la question où l'interrogé a menti. On corrige alors m en changeant m_k et $\overline{m_1 m_2 m_3 m_4}$ est le nombre à deviner écrit en base 2.

Expliquer ce tour! On pourra méditer sur le code correcteur de matrice de contrôle

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

Pour $k = 2^r - 1 - r$, créer un tour de magie similaire devinant un nombre entre 0 et $2^k - 1$ permettant de corriger au plus un mensonge. Programmer ces tours de magie pour $r \in \{2, 3, 4\}$.

8 Complexité

Que dire des complexités des différents algorithmes mis en œuvre dans ce TP?