



DIRO
IFT 1215

INTRODUCTION AUX SYSTÈMES INFORMATIQUES

CODES CORRECTEURS

Max Mignotte

Département d'Informatique et de Recherche Opérationnelle
Http : [//www.iro.umontreal.ca/~mignotte/](http://www.iro.umontreal.ca/~mignotte/)
E-mail : mignotte@iro.umontreal.ca

CODES CORRECTEURS

SOMMAIRE

Introduction -Contrôle de parité	2
Double parité	3
Code de Hamming	4
Détection d'erreurs groupés : Code CRC	11
Code CRC -Exemple	13
Code CRC -Polynômes générateurs	15
Exercices	16

CODES CORRECTEURS CODE DE HAMMING

Code de Hamming

Code auto-correcteur

m Bits d'info + k bits de contrôle de parité
 ► $m + k = n$ Bits transférés

m	0	0	1	1	2	3	4	4	5	6	7	8	9	10	...	120	...
k	1	2	2	3	3	3	3	4	4	4	4	4	4	4	4	8	...
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	128	...

Exemple

Si $m = 4$ (nb de Bits d'information), on peut construire un code de Hamming sur $n = 7$ Bits en ajoutant $k = 3$ Bits de contrôle

7	6	5	4	3	2	1
m_4	m_3	m_2	k_3	m_1	k_2	k_1

Les 3 bits de contrôle k_3, k_2, k_1 sont placés sur les puissances de 2

- k_1 en position 1
- k_2 en position 2
- k_3 en position 4

CODES CORRECTEURS CODE DE HAMMING

Si $m = 4, k = 3$

Pour chaque Bit du message ► Bit contrôlant sa parité ?

Bit n° 7	(0111)	Bit n° 7 est contrôlé par k_3, k_2, k_1
Bit n° 6	(0110)	Bit n° 6 est contrôlé par k_3, k_2
Bit n° 5	(0101)	Bit n° 5 est contrôlé par k_3, k_1
Bit n° 4	(0100)	Bit de contrôle k_3
Bit n° 3	(0011)	Bit n° 3 est contrôlé par k_2, k_1
Bit n° 2	(0010)	Bit de contrôle k_2
Bit n° 1	(0001)	Bit de contrôle k_1

Inversement, qui contrôle qui ?

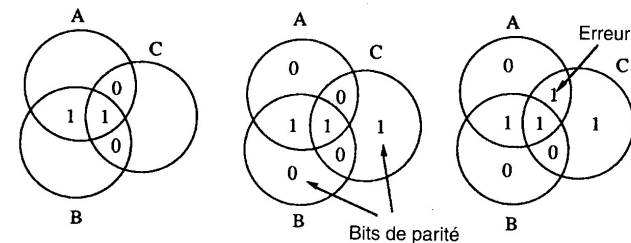
k_1	contrôle les bits	1, 3, 5, 7
k_2	contrôle les bits	2, 3, 6, 7
k_3	contrôle les bits	4, 5, 6, 7

Exemple

Avec une parité paire, k_1 doit être tel que le nombre de bits à 1, compté sur les bits 1, 3, 5, 7 soit pair

Compréhension graphique

Codage du mot 1100



CODES CORRECTEURS CODE DE HAMMING

k_1	contrôle les bits	1, 3, 5, 7
k_2	contrôle les bits	2, 3, 6, 7
k_3	contrôle les bits	4, 5, 6, 7

Quand on reçoit l'information, on effectue à nouveau le contrôle de parité et pour chaque bit de contrôle, on compare la valeur reçue à celle recalculée

Si elles sont identiques, On assigne la valeur 0 à la variable binaire A_i au bit de contrôle k_i , sinon on lui assigne la valeur 1

La valeur des A_i nous donne la position de l'erreur

- $A_3A_2A_1 = 000$ indique une absence d'erreur
- $A_3A_2A_1 = 001$ indique une erreur sur le bit n° 1
- ...

Exemple

$m = 4, k = 3, \text{parité paire, réception de } 1011100$

numéro	7	6	5	4	3	2	1
type	m_4	m_3	m_2	k_3	m_1	k_2	k_1
valeur	1	0	1	1	1	0	0

$k_1 = 0$, (bits 1, 3, 5, 7) est donc faux $\rightarrow A_1 = 1$

$k_2 = 0$, (bits 2, 3, 6, 7) est juste $\rightarrow A_2 = 0$

$k_3 = 1$, (bits 4, 5, 6, 7) est donc faux $\rightarrow A_3 = 1$

L'adresse binaire de l'erreur est $A_3A_2A_1 = 101_2 = 5_{10}$

Le bit 5 est faux. Le message corrigé est 1001100 sans les bits de contrôle, le message corrigé est 1001

CODES CORRECTEURS CODE DE HAMMING

Exemple 8 Bits d'info + 4 Bits de contrôle, parité paire

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1

$$\begin{aligned}
 C1 &= D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7 \\
 C2 &= D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7 \\
 C4 &= D2 \oplus D3 \oplus D4 \oplus D8 \\
 C8 &= D5 \oplus D6 \oplus D7 \oplus D8
 \end{aligned}$$

Word stored as	0	0	1	1	0	1	0	0	1	1	1	1
Word fetched as	0	0	1	1	0	1	1	0	1	1	1	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Check bit					0				0		0	1

$$\begin{aligned}
 C1 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 & C1 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 \\
 C2 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 & C2 &= 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0 \\
 C4 &= 0 \oplus 0 \oplus 1 \oplus 0 = 1 & C4 &= 0 \oplus 1 \oplus 1 \oplus 0 = 0 \\
 C8 &= 1 \oplus 1 \oplus 0 \oplus 0 = 0 & C8 &= 1 \oplus 1 \oplus 0 \oplus 0 = 0
 \end{aligned}$$

Calcul

Lu

Erreur sur le 6ième Bit, cad D_3

CODES CORRECTEURS CODE DE HAMMING

Calcul Simplifié du code de Hamming

1 Transmission d'un message

Coder 10101011001 avec une parité paire
 $m = 11$ donc $k = 4$ ($n = 15$)

numéro	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
type	m_{11}	m_{10}	m_9	m_8	m_7	m_6	m_5	k_4	m_4	m_3	m_2	k_3	m_1	k_2	k_1
valeur	1	0	1	0	1	0	1	?	1	0	0	?	1	?	?

Dans le message à transmettre, on a des bits à 1 dans les positions : 15, 13, 11, 9, 7, 3

On transforme ces positions en valeur binaire et on les additionne modulo 2 : on met 1 lorsque l'on a un nombre impair de 1 et 0 pour un nombre pair de 1

$$\begin{array}{r}
 15 = 1 \ 1 \ 1 \ 1 \\
 13 = 1 \ 1 \ 0 \ 1 \\
 11 = 1 \ 0 \ 1 \ 1 \\
 9 = 1 \ 0 \ 0 \ 1 \\
 7 = 0 \ 1 \ 1 \ 1 \\
 3 = 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 0 \ 0 \rightarrow \text{bits de parité} \\
 k_4 \ k_3 \ k_2 \ k_1
 \end{array}$$

Message codé 1010101001100

Nota

Parité impaire, Nb impair de 1 \triangleright 0 (Nb pair de 1 \triangleright 1)

CODES CORRECTEURS CODE DE HAMMING

2 Réception d'un message

Réception de 101000101001100 avec une parité impaire
 $m = 11$ donc $k = 4$ ($n = 15$)

On a des Bits à 1 dans les positions : 15, 13, 9, 7, 4, 3

$$\begin{array}{r}
 15 = 1 \ 1 \ 1 \ 1 \\
 13 = 1 \ 1 \ 0 \ 1 \\
 9 = 1 \ 0 \ 0 \ 1 \\
 7 = 0 \ 1 \ 1 \ 1 \\
 4 = 0 \ 1 \ 0 \ 0 \\
 3 = 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 0 \ 0 \\
 A_4 \ A_3 \ A_2 \ A_1 \rightarrow \text{erreur à la position 4}
 \end{array}$$

addition modulo 2 inversée

Après correction du Bit en position 4, on a le message
 101000101000100

Après correction, ce message à des à 1 dans les positions

$$\begin{array}{r}
 15 = 1 \ 1 \ 1 \ 1 \\
 13 = 1 \ 1 \ 0 \ 1 \\
 9 = 1 \ 0 \ 0 \ 1 \\
 7 = 0 \ 1 \ 1 \ 1 \\
 3 = 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 0 \ 0 \\
 \rightarrow \text{aucune erreur détectée}
 \end{array}$$

addition modulo 2 inversée

Après élimination des bits redondant, le message est
 10100011001

CODES CORRECTEURS CODE DE HAMMING

Code de Hamming et erreurs groupés

Arrangement matriciel du code de Hamming :

ASCII	code de Hamming (pour chaque lettre)											
	11	10	9	8	7	6	5	4	3	2	1	(numéros)
H -> 1001000	1	0	0	1	1	0	0	1	0	0	0	
a -> 1100001	1	1	0	0	0	0	0	0	1	1	0	
m -> 1101101	1	1	0	0	1	1	0	0	1	1	1	
m -> 1101101	1	1	0	0	1	1	0	0	1	1	1	
i -> 1101001	1	1	0	0	1	0	0	1	1	0	1	
n -> 1101110	1	1	0	0	1	1	1	1	0	0	1	
g -> 1100111	1	1	0	0	0	1	1	0	1	0	1	

On transmet les Bits colonnes par colonne (on aura un bit maximal erroné par ligne)

CODES CORRECTEURS DÉTECTION D'ERREURS GROUPÉS : CODE CRC

Code CRC

[Cyclic Redundant Coding] ou codes polynomiaux

Information de n Bits \triangleright polynôme de degré $n - 1$

$$1101 \rightarrow x^3 + x^2 + 1$$

$$110001 \rightarrow x^5 + x^4 + 1$$

Pour calculer les bits de contrôle, on utilisera l'addition et la soustraction modulo 2

Somme modulo 2

$$\begin{array}{r} 10011011 \\ + 11001010 \\ \hline 01010001 \end{array}$$

Soustraction modulo 2

$$\begin{array}{r} 11110000 \\ - 10100110 \\ \hline 01010110 \end{array}$$

Algorithme

$M(x)$ \triangleright Polynôme associé au message original M

$G(x)$ \triangleright Polynôme générateur de degré r choisi

— Envoie —

- 1 $M(x) \leftarrow M(x)x^r$ (\triangleright ajout de r "0" à la fin de M)
- 2 Effectuer la division (modulo 2)

$$\frac{M(x)x^r}{G(x)} = Q(x) + R(x)$$

CODES CORRECTEURS

DÉTECTION D'ERREURS GROUPÉS : CODE CRC

- 3 Le Quotient $Q(x)$ est ignoré
Le reste $R(x)$ [*checksum*] contient r bits (degré $r - 1$)

On effectue la soustraction modulo 2

$$M(x)x^r - R(x) = T(x)$$

$T(x)$ ▷ Polynôme cyclique & message à envoyé

— Réception —

- 1 On effectue la division

$$\frac{T(x)}{G(x)}$$

- Si le reste = 0 il n'y a pas d'erreur
- Si le reste $\neq 0$, il y a erreur, on doit retransmettre

Nota

En choisissant judicieusement $G(x)$, on peut détecter toute erreur sur 1 Bit, 2 Bits, une séquence de n bits et au delà la détection est possible avec une très grande probabilité

CODES CORRECTEURS

CODE CRC -EXEMPLE

Exemple 1 : — Envoie —

Message $M = 101101$ ▷ $M(x) = x^5 + x^3 + x^2 + 1$

Poly. générateur $G = 1011$ ▷ $G(x) = x^3 + x + 1$

- 1 $M(x) \leftarrow M(x) x^r$

$$M(x)x^3 = 101101000$$

- 2 Division modulo 2 de $\frac{M(x)x^r}{G(x)}$

$$\begin{array}{r}
 101101000 \mid 1011 \\
 \underline{1011} \\
 000001000 \\
 \underline{1011} \\
 0011 \rightarrow R(x) = 011
 \end{array}$$

- 3 $Q(x)$ ignoré, Soustraction modulo 2

$$M(x)x^r - R(x) = T(x)$$

$$T(x) = 101101011$$

CODES CORRECTEURS CODE CRC -EXEMPLE

Exemple 2 : — Réception d'un message —

Message $M = 11010101 = T(x)$

Poly. générateur $G = 1011 \triangleright G(x) = x^3 + x + 1$

- 1 On effectue la division (modulo 2) $\frac{T(x)}{G(x)}$

$$\begin{array}{r|l}
 11010101 & 1011 \\
 \hline
 1011 & \\
 \hline
 01100 & \\
 1011 & \\
 \hline
 01111 & \\
 1011 & \\
 \hline
 01000 & \\
 1011 & \\
 \hline
 00111 & \rightarrow R(x) = 111
 \end{array}$$

Des erreurs de transmission ont été détecté
Il faut retransmettre

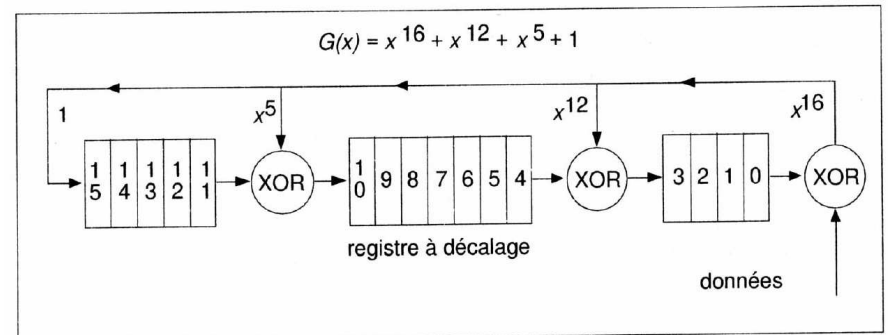
CODES CORRECTEURS CODE CRC -POLYNÔMES GÉNÉRATEUR

Polynôme Générateur

CRC-12 $= x^{12} + x^{11} + x^3 + x^2 + x + 1;$

CRC-16 $= x^{16} + x^{15} + x^2 + 1;$

CRC-CCITT $= x^{16} + x^{12} + x^5 + 1.$



Circuits permettant de calculer $R(x)$

- Au départ le registre à décalage contient tous ses (16) bits à 0, à la fin, il contient le CRC
- Pour chaque bit en entrée, on effectue un décalage de gauche à droite en tenant compte des sorties des différents XOR

CODES CORRECTEURS EXERCICES

Exo -1-

Transmission par codage de Hamming du message 116570_8 ,
parité impaire

Message $\triangleright 116570_8 = 1\ 001\ 110\ 101\ 111\ 000_2$

Nb de Bits de contrôle : $k = 5$

k_1	contrôle les bits	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21	\triangleright	0
k_2	contrôle les bits	2, 3, 6, 7, 10, 11, 14, 15, 18, 19	\triangleright	0
k_3	contrôle les bits	4, 5, 6, 7, 12, 13, 14, 15, 20, 21	\triangleright	1
k_4	contrôle les bits	8, 9, 10, 11, 12, 13, 14, 15	\triangleright	0
k_5	contrôle les bits	16, 17, 18, 19, 20, 21	\triangleright	0

Le message codé à transmettre est
10011010101101001000

Avec la méthode de Hamming simplifiée

$$\begin{array}{r}
 21 \rightarrow \sim 1\ 0\ \bar{1}\ 0\ 1 \\
 18 \rightarrow 1\ 0\ 0\ 1\ 0 \\
 17 \rightarrow 1\ 0\ 0\ 0\ 1 \\
 15 \rightarrow 0\ 1\ 1\ 1\ 1 \\
 13 \rightarrow 0\ 1\ 1\ 0\ 1 \\
 11 \rightarrow 0\ 1\ 0\ 1\ 1 \\
 10 \rightarrow 0\ 1\ 0\ 1\ 0 \\
 9 \rightarrow 0\ 1\ 0\ 0\ 1 \\
 7 \rightarrow 0\ 0\ 1\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 0
 \end{array}$$

CODES CORRECTEURS EXERCICES

Exo -2-

Réception du message 6130014_8 , codage de Hamming,
parité impaire

Nb de Bits de contrôle : $k = 5$, $n = 21$

$k_1 = 0$, il contrôle les bits 1,3,5,7,9,11,13,15,17,19,21; il est juste	$\rightarrow A_1 = 0$
$k_2 = 0$, il contrôle les bits 2,3,6,7,10,11,14,15,18,19; il est donc faux	$\rightarrow A_2 = 1$
$k_3 = 1$, il contrôle les bits 4,5,6,7,12,13,14,15,20,21; il est juste	$\rightarrow A_3 = 0$
$k_4 = 0$, il contrôle les bits 8,9,10,11,12,13,14,15; il est donc faux	$\rightarrow A_4 = 1$
$k_5 = 1$, il contrôle les bits 16,17,18,19,20,21; il est juste	$\rightarrow A_5 = 0$

L'adresse binaire de l'erreur est : $A_5 A_4 A_3 A_2 A_1 = 0\ 1\ 0\ 1\ 0 = 10$.

Le Bit 10 du message transmis est donc faux

Le message corrigé est
1 100 001 100 100 001

Avec la méthode de Hamming simplifiée

$$\begin{array}{r}
 21 \rightarrow 1\ 0\ 1\ 0\ 1 \\
 20 \rightarrow 1\ 0\ 1\ 0\ 0 \\
 16 \rightarrow 1\ 0\ 0\ 0\ 0 \\
 14 \rightarrow 0\ 1\ 1\ 1\ 0 \\
 13 \rightarrow 0\ 1\ 1\ 0\ 1 \\
 4 \rightarrow 0\ 0\ 1\ 0\ 0 \\
 3 \rightarrow 0\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 1\ 0\ 1\ 0
 \end{array}$$

CODES CORRECTEURS EXERCICES

Exo -3-

Méthode CRC, Message à transm. $M = 456_8 = 100101110_2$

Poly. générateur $G = 100011 \triangleright G(x) = x^5 + x + 1$

Message à envoyer ?

- $M(x) \leftarrow M(x) x^r \quad M = 10010111000000$

•

$$\begin{array}{r}
 100101110000000 \mid 100011 \\
 \underline{1000011} \\
 000110110 \\
 \underline{100011} \\
 0101010 \\
 \underline{100011} \\
 00100100 \\
 \underline{100011} \\
 00011100
 \end{array}$$

- $R(x) = 11100 \triangleright$ Message à envoyer

$$T(x) = 10010111011100 = 22734_8$$

CODES CORRECTEURS EXERCICES

Exo -4-

CRC, Recep. du message $M = 76543_8 = 111110101100011_2$

Poly. générateur $G = 100011 \triangleright G(x) = x^5 + x + 1$

Message à re-envoyer ?

•

$$\begin{array}{r}
 111110101100011 \mid 100011 \\
 \underline{100011} \\
 0111011 \\
 \underline{100011} \\
 0110000 \\
 \underline{100011} \\
 0100111 \\
 \underline{100011} \\
 000100100 \\
 \underline{100011} \\
 000111011 \\
 \underline{100011} \\
 011000
 \end{array}$$

Le reste est $\neq 0$

Il faut retransmettre le message