Détection et correction d’erreurs

1. Introduction

Exercices

1. Quelle est la distance de Hamming entre deux mots valides dans le code ASCII ?
Quels sont les types d’erreurs que permet de déceler un tel codage ?

2. On choisit un code à trois bits, les mots de code valides sont ceux dont les trois bits sont identiques.
Quelle est la distance de Hamming ?
Si une erreur se produit sur un seul bit, quels sont les mots que l’on peut obtenir, et par quel mot valide chacun peut-il être remplacé sans ambiguïté ?

1. La distance est de 1 ! Il suffit de changer 1 bit pour retomber sur un mot valide
Pour une distance d de Hamming donnée,

On peut détecter avec certitude n bits avec $n+1 \leq d$

On peut corriger avec certitude m bits avec $2m+1 \leq d$

Le bit d’un code détecteur (et correcteur) d’erreur est d’augmenter la distance. Dans notre cas, aucune détection (et donc correction n’est possible → n=0

2. Les mots du code valides sont dont 000 et 111 : la distance est donc de 3
Les mots erronés avec 1 seul bit faux sont : 001, 010, 100, 110, 101, 011
Puisque la distance de Hamming est 3,

- on doit pouvoir détecter au maximum 3-1=2 bits
- et corriger (3-1)/2 = 1 bit.

La correction consiste à revenir au mot valide le plus proche du mot erroné reçu. Ainsi, les 3 premiers mots seront remplacés par 000 (car ils ont une distance de 1 avec 000 et une distance de 2 avec 111, il est bien sur nécessaire de faire la supposition qu'il n'y a pas eu plus d'une erreur) et les trois derniers par 111 (pour des raisons similaires).

2. Codes par bloc

Exercices

Dans l'alphabet ASCII le mot « OSI » se code par les 3 caractères de 7 bits suivants :
'O' = 1001111, 'S' = 1010011 et 'I' = 1000011

1. La LRC (Longitudinal Redundancy Check) consiste à rajouter un bit de parité à la fin d'un bloc de données (octet, caractère, suite de bits, …). La VRC (Vertical Redundancy Check) consiste à calculer les bits de parité entre plusieurs blocs de données en vertical :1 bit de parité pour les bits qui sont à la même position dans les différents blocs considérés.
Donnez la VRC du mot « OSI » en utilisant une parité paire pour calculer le LRC de chaque caractère

2. Combien d'erreurs ce code peut-il détecter ? Combien peut-il en corriger ?

1.

1001111 **1**
1010011 **0**
1000011 **1**
1011111 0 ← VRC
↑ LRC

2.

Il peut détecter jusqu'à 3 erreurs, s'il y en a quatre, on n'est pas certain de détecter car on retombe sur un code valide.
Il peut corriger 1 erreur (la position du bit est donnée par l'intersection de la ligne du bit incorrect dans le VRC avec la colonne du bit incorrect dans le LRC. Lorsqu'il y a plus de 2 erreurs l'ambiguïté empêche la correction de l'erreur. Il faut raisonner par l'exemple : modifier un bit, puis 2, puis 3 puis 4.

Il est également possible de déterminer la distance de Hamming, c'est un moyen simple et efficace. On constate que sur un bit quelconque de données il y a trois bits de contrôle qui interviennent. Il faut donc changer 4 bits pour retomber sur quelque chose de valide. La distance est égale à 4. D'après les formules données en cours on peut détecter n erreurs tel que $n+1 \leq$ dist et corriger m erreur tel que $2m+1 \leq$ dist. Pour nous, on trouve n=3 et m=1 (arrondis à l'entier inférieur)

3. Code de Hamming

Exercices

1. On souhaite utiliser le codage de Hamming pour émettre la suite de bits (1011). Quelle est la séquence à transmettre ?
2. Y a-t-il une erreur dans le mot suivant : 1101101 ?
3. Soit le mot de Hamming suivant : 101101111011011
    a. Quels sont les bits de contrôle de parité ?
    b. Quelles positions contrôle chacun de ces bits ?
    c. Quel est le message reçu ?
    d. Y a-t-il une erreur dans le message ?

1.

m_4	m_3	m_2	**k_3**	m_1	**k_2**	**k_1**
1	*0*	*1*	**0**	*1*	**0**	**1**

- k_1 contrôle m_1, m_2 et m_4 → 1+1+1, k_1 = 1 (parité paire)
- k_2 contrôle m_1, m_3 et m_4 → 1+0+1, k_2 = 0 (parité paire)
- k_3 contrôle m_2, m_3 et m_4 → 1+0+1, k_3 = 0 (parité paire)

Le message envoyé sera donc 101**0**1**01**

2.

m_4	m_3	m_2	**k_3**	m_1	**k_2**	**k_1**
1	*1*	*0*	**1**	*1*	**0**	**1**

- k_1 contrôle m_1, m_2 et m_4 → 1+0+1, or k_1 = 1 (parité paire) **PBLE**
- k_2 contrôle m_1, m_3 et m_4 → 1+0+1, or k_2 = 0 (parité paire) **OK**
- k_3 contrôle m_2, m_3 et m_4 → 0+1+1, or k_3 = 0 (parité paire) **PBLE**

Le bit en erreur est donc en position ($k_3k_2k_1$) soit en position $(101)_2$ soit en position 5 (m_2) → le 0 doit être un **1**.

3.

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**
1	0	1	1	0	1	1	**1**	1	0	1	**1**	0	**1**	**1**

a. les bits de contrôle de parité sont k_1 (position 0001), k_2 (position 0010), k_3 (position 0100) et k_4 (position 1000)

b.

- k_1 contrôle m_1, m_2, m_4, m_5, m_7, m_9,m_{11} → 1
- k_2 contrôle m_1, m_3, m_4, m_6, m_7, m_{10} et m_{11} → 1
- k_3 contrôle m_2, m_3, m_4, m_8, m_9, m_{10} et m_{11} → 1
- k_4 contrôle m_5, m_6, m_7, m_8, m_9, m_{10} et m_{11} → 1

c.

Le message reçu est donc : 10110111010

d.

Tout est donc ok (pas d'erreurs dans les bits de contrôle).

4. Codes CRC (Cyclic Redundancy Check)

Exercices

1. Calculez le CRC du mot « OSI » en utilisant le polynôme générateur $x^8 + 1$ et en supposant que le 8ème bit de chaque caractère est un bit de parité paire et que le mot d'information est composé des bits des 3 caractères à la suite.

2. Calculez avec la méthode de la division polynomiale le bloc de contrôle (CRC) correspondant à la suite de bits 1100101010101011 en utilisant le polynôme générateur $G(x) = x^4 + x^3 + x + 1$

3. On utilisera le polynôme générateur x^4+x^2+x
- On souhaite transmettre le message suivant 1111011101, quel sera le CRC à ajouter ?
- On reçoit les messages : 1111000101010 et 11000101010110, sont-ils corrects ?

4. La détection d'erreurs utilise le CRC $x^6 + x^4 + x + 1$.
Le récepteur reçoit la séquence binaire suivante 101011000110
Le message est-il correct (tester avec la méthode de division polynomiale binaire) ?

1.

I = 100111111010011010000111 les bits en gras représentent la VRC

G = 100000001 et degré = 8

En utilisant la division polynomiale on obtient R = 10111110 (ce qui correspond à la LRC). Donc, la LRC peut être obtenue par un code cyclique

2.

I = 1100101010101011,

degré = 4 ⇒ C0 = 11001010101010110000

⇒ $C0(x) = x^{19}+ x^{18}+ x^{15}+ x^{13}+ x^{11}+ x^9+ x^7+ x^5+ x^4$

G = 11011 ⇒ $G(x) = x^4+ x^3+ x+1$

On effectue la division de C0(x) par G(x) (la division polynomiale symbolique est immédiate).

$R(x) = x^3$ donc R = 1000, on transmet alors C = IR = 11001010101010111000

3.

- I = 1111011101

  degré = 4 ⇒ C0 = 1111011101**0000** à diviser par **10110**

  ➔ CRC = **1100**
- On procède à la division binaire 1111000101010 par 10110 → Reste à 0 **OK**
- On procède à la division binaire 11000101010110 par 10110 → Reste à 1110 **PBLE**

4.

On procède à la division binaire 101011000110 par 1010011 → Reste à 0 **OK**