Codes correcteurs d'erreur

Emmanuel Jeandel (emmanuel.jeandel@lif.univ-mrs.fr)
http://www.lif.univ-mrs.fr/~ejeandel/enseignement.html

22 mars 2011

RAID

RAID est une technologie qui permet de fusionner des disques durs de même capacité pour obtenir un seul disque dur de capacité plus grande, mais qui résiste aux pannes.

Le RAID-1 prend deux disques durs, d'une capacité de 100 Go, et permet d'obtenir un périphérique qui se comporte comme un seul disque dur de 100 Go. Le système est fait de telle façon que si l'un des disques durs ne fonctionne plus, aucune donnée n'est perdue, et on peut reconstituer ce que contenait ce disque dur.

Q 1) Expliquer comment, à votre avis, peut fonctionner RAID-1.

Le RAID-4 prend quatre disques durs, d'une capacité de 100 Go, et permet d'obtenir un périphérique qui se comporte comme un seul disque dur de 300 Go. Le système est fait de telle façon que si l'un des disques durs ne fonctionne plus, aucune donnée n'est perdue, et on peut reconstituer ce que contenait ce disque dur.

Q 2) Expliquer comment, à votre avis, peut fonctionner RAID-4.

Concaténation

On considère un code de paramètre (n, m) et un code de paramètre (p, q). On décide de faire un code de paramètre $(n + p, m + q)$ en concaténant les deux codes.

Q 3) Montrer que si les deux codes détectent une erreur, le code résultant en détecte une.

Q 4) Montrer que si les deux codes corrigent une erreur, le code résultant en corrige une.

Inversion

On dit qu'un code détecte une inversion (resp. une inversion consécutive) si lorsqu'un mot est transmis avec deux chiffres (resp. deux chiffres consécutifs) inversés, on peut s'en rendre compte.

Q 5) Montrer que le code $(9, 15)$ vu en cours (et qui assemble les 9 bits en un carré, et calcule les parités des lignes et des colonnes) détecte une inversion.

Q 6) Montrer que le code ISBN permet de détecter une inversion.

Q 7) Montrer que le code suivant
- détecte une erreur ;
- détecte une inversion consécutive ;
- ne corrige pas une erreur ;
- ne détecte pas une inversion quelconque.

$$00000, 01100, 00110, 11111$$

Q 8) Montrer que le code suivant
- ne détecte pas une erreur ;
- détecte une inversion consécutive ;
- ne détecte pas une inversion quelconque.

$$0000, 0001, 0101, 0100$$

Q 9) Montrer qu'un code qui détecte deux erreurs détecte une inversion

Q 10) Montrer qu'un code linéaire détecte une inversion consécutive si et seulement si il n'y a pas de mots de code ne contenant que des 0 et deux chiffres 1 consécutifs.

Reed-Muller

Si x et y sont deux mots de n lettres sur l'alphabet $\{0,1\}$, on note $x \oplus y$ la somme de x et y bit à bit. Par exemple

$$00010 \oplus 01010 = 01000$$

$$10010 \oplus 11000 = 01010$$

On rappelle quelques propriétés de $\oplus$ et de la distance de Hamming d vue en cours :

Théorème 1

Un code C détecte k erreurs si et seulement s'il n'existe pas deux mots de code à distance inférieure à k, c'est à dire :

$$\forall a \in C, \forall b \in C, d(a,b) > k$$

Proposition 2

$$d(a \oplus c, b \oplus c) = d(a,b)$$

$$d(a,c) \leq d(a,b) + d(b,c)$$

$$a \oplus a = 0$$

Si C_1 et C_2 sont deux codes sur n lettres, on note $C_1 \star C_2$ l'ensemble des mots xz où x est un mot de C_1 et z est la somme ($\oplus$) de x et d'un mot y de C_2.

Par exemple, si $C_1 = \{00, 01, 11\}$ et $C_2 = \{00, 11\}$, alors

$$C_1 \star C_2 = \{0000, 0101, 1111, 0011, 0110, 1100\}$$

1100 s'obtient par exemple en prenant $x = 11$ et $y = 11$ (puisque $x \oplus y = 00$). Le code 0101 s'obtient en prenant $x = 01$ et $y = 00$ (puisque $x \oplus y = 01$).

Q 11) Calculer $C_1 \star C_2$ lorsque

$$C_1 = C_2 = \{00000, 01100, 00110, 11111\}$$

Q 12) Si C_1 contient m_1 mots et C_2 contient m_2 mots, calculer combien de mots contient $C_1 \star C_2$. Justifier.

Q 13) On suppose que C_1 détecte d_1 erreurs et C_2 en détecte d_2. Montrer que $C_1 \star C_2$ en détecte $\min(2d_1 + 1, d_2)$. (Aide : Prendre deux mots de code a et b de $C_1 \star C_2$ différents et montrer qu'on a dans ce cas $d(a,b) > \min(2d_1 + 1, d_2)$. On pourra faire deux cas)

Le code de Reed-Muller $R(n, 0)$ est par définition le code

$$\{\underbrace{0 \cdots 0}_{2^n}, \underbrace{1 \cdots 1}_{2^n}\}$$

il est donc constitué de deux mots, de longueur 2^n.

Q 14) Combien ce code détecte-t-il d'erreur ?

Le code de Reed-Muller $R(n, n)$ est par définition le code contenant l'ensemble des mots de longueur 2^n (il contient donc 2^{2^n} mots)

Q 15) Combien ce code détecte-t-il d'erreur ?

Le code de Reed-Muller $R(n,k)$ est défini de la façon suivante :

$$R(n,k) = R(n-1,k) \star R(n-1,k-1)$$

Q 16) Calculer $R(2,1)$ et $R(3,1)$

Q 17) Montrer que $R(n,k)$ détecte $2^{n-k}-1$ erreurs. On utilisera une récurrence.

Q 18) Montrer par récurrence que le nombre de mots de $R(n,k)$ est

$$2^{C_n^0+C_n^1+\cdots+C_n^k}$$

TP

Afin de pouvoir corriger des erreurs, on va transformer un fichier a.txt en un fichier a.txt.cor de la façon suivante :

- On lit les caractères 9 par 9
- On écrit les caractères dans une grille 3×3 et on calcule les sommes de chaque ligne et de chaque colonne (comme vu en cours avec des bits)
- On écrit les 15 bits résultants dans le nouveau fichier, de haut en bas.

Il n'est pas demandé de gérer le cas où le fichier n'a pas une taille multiple de 9.

Par exemple, si on part du fichier toto.txt suivant :

```
Ceci est un test.
```

Qui contient les caractères ascii suivants (obtenus en utilisant la commande `od -Ad -td1 toto.txt`) :

```
0000000   67  101   99  105   32  101  115  116   32  117  110   32  116  101  115  116
0000016   46   10
```

On doit obtenir le fichier toto.txt.cor suivant :

```
0000000  67 101  99  11 105  32 101 238 115 116  32   7  31 249 232 117
0000016 110  32   3 116 101 115  76 116  46  10 172  93   1 157
```

En effet $11 = 67 + 101 + 99 \mod 256$ et $101 + 32 + 116 = 249 \mod 256$.

Q 19) Ecrire le programme qui transforme `toto.txt` en `toto.txt.corr`.

Q 20) Ecrire le programme qui transforme `toto.txt.corr` en `toto.txt` en corrigeant les erreurs. Tester sur le fichier `b.zip.cor` disponible sur la page web du cours.