4 – Codes correcteurs : codes linéaires – codes de Hamming

Des codes hyper utilisés

- on les rencontre partout dès lors que le canal est bruité
  du flash-code au satellite (DVB), *en passant par le* GSM, *les* MODEM, *les* CD, *les* DVD, *l'*ADSL, *le* WI-FI ...
- on distingue différentes familles de codes correcteurs :
  - les codes de répétition
  - les codes linéaires (Hamming, Golay)
  - les codes cycliques (BCH, Reed-Solomon dont CIRC)
  - les codes convolutifs
  - les turbo-codes
- la plupart de ces codes utilisent des *polynômes* ou de l'*algèbre linéaire*
- des techniques permettent d'améliorer encore et toujours ces codes : *extension*, *entrelacement*, *croisement* et *concaténation*.

Codage par blocs

- le codage par blocs commence par découper un message binaire en blocs de $\{0,1\}^k$
- on code ensuite bloc par bloc et on transmet dans un canal bruité
- si le bloc reçu diffère du bloc émis, il y a eu erreur
- le codage va consister en un ajout de redondance de taille r
- on parle alors de (n,k)-code avec $n = k + r$
- les images de $\{0,1\}^k$ par le code sont appelées les mots de code
  il s'agit de codes à longueur fixe donc de codes préfixes (i.e. instantannés)
  l'injectivité de la fonction de codage suffit à garantir la non-ambiguïté du code
- le rendement R d'un (n,k)-code est égal à

$$R = \frac{k}{n}$$

- la redondance vise à permettre la détection d'erreurs et éventuellement la correction

Détection et correction d'erreurs

- un message $m = m_1 ... m_k$ est codé en $c = c_1 ... c_n$ puis est envoyé par le canal
- le récepteur reçoit $c' = c'_1 ... c'_n$
- l'ensemble C regroupe tous les mots de code possibles
- si c' appartient à C, on estime que c' est sans erreur et on le décode naturellement en m
- sinon, il y a détection d'une erreur et l'ensemble V des mots de code *les plus proches* de c' est considéré :
  si $V = \{v\}$ est singleton, on corrige c' en v puis on décode
  sinon, on peut corriger de façon plus hasardeuse par rapport à un des mots de code de V avant de décoder
- si la correction n'est pas possible, la retransmission (ARQ) est effective

Exemple
L'ajout d'un bit de parité est un code détecteur d'erreurs.

Deuxième théorème de Shannon

X est la v.a. d'entrée du canal et Y celle de sortie

- $H(X)$: entropie de la source X à l'entrée du canal
- l'entropie conditionnelle $H(X/Y)$ traduit la perte d'information entre X à l'entrée et Y à la sortie du canal
  c'est ce qu'il reste encore à découvrir sur X alors qu'on a reçu Y
- la capacité du canal est la quantité maximale d'information sur l'entrée effectivement transmise par le canal

$$\mathcal{C} = \max_{p} \left(H(X) - H(X/Y) \right)$$

p étant une distribution de probabilités

si $H(X/Y) = 0$: la source n'apprend rien de plus que la sortie du canal, c'est que le canal est fiable
si l'entropie $H(X/Y)$ est maximale : la capacité devient nulle et la transmission impossible

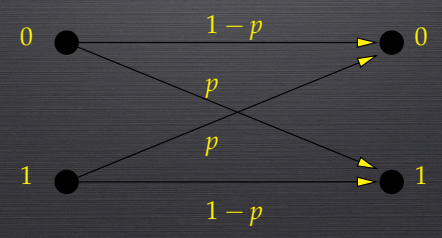
- Deuxième théorème de Shannon
  Soit un canal de capacité $\mathcal{C}$. Pour tout $\varepsilon > 0$, il existe au moins un (n,k)-code de rendement $R = \frac{k}{n}$ et de probabilité d'erreur $p < \varepsilon$ si et seulement si :

$$0 \leq R < \mathcal{C}$$

Canal BSC

Canal Binaire Symétrique *(Binary Symmetric Channel)*
p correspond à la probabilité de recevoir 1 alors que 0 a été émis ou, symétriquement, de recevoir 0 alors que 1 a été émis

- si $p = 1/2$: la capacité du canal devient nulle et un tel canal est inutilisable
- si $p < 1/2$: le théorème assure qu'il existe un (n,k)-code permettant de transmettre sans erreur sur un canal de capacité non nulle
- cela dit, le rendement du code est limité par la capacité du canal



Taux d'erreur

- Ordres de grandeur du taux d'erreur

ligne	taux d'erreur
Disquette	10^{-9} : à 5Mo/s, 3 bits erronés/mn
CD-ROM optique	10^{-5} : 7ko erronés sur 700 Mo
DAT audio	10^{-5} : à 48 kHz, 2 erreurs/s
Mémoires à semi-conducteurs	$< 10^{-9}$
Liaison téléphonique	entre 10^{-4} et 10^{-7}
Télécommande infrarouge	10^{-12}
Fibre optique	10^{-9}
Satellite	10^{-6} (Voyager), 10^{-11} (TDMA)
ADSL	10^{-3} à 10^{-9}
Réseau informatique	10^{-12}

(In : Théorie des codes, Dunod, 2007)

Codes linéaires : généralités

- ce sont des codes de canal
- ils sont basés sur de l'*algèbre linéaire* sur $\mathbb{F}_2$
- ces codes sont linéaires car l'ensemble des mots du code forment un sous-espace vectoriel
- la longueur des mots du code est fixe, elle correspond à la dimension augmentée de la redondance
- ce sont des codes correcteurs d'erreurs
- pour coder ou décoder, on utilise la seule matrice génératrice G
- pour détecter une erreur et la réparer, on utilise la matrice de contrôle H
- le code est entièrement spécifié par l'une ou l'autre de ces deux matrices
- la somme de deux mots du code reste un mot du code

Codes de Hamming (1950)

- les codes de Hamming sont emblématiques des codes linéaires
- un message binaire $\mathcal{M}$ est augmenté d'une redondance de r bits
- la longueur du code est $n = 2^r - 1$
- la dimension (taille des messages d'origine) est égale à $k = n - r$
- la matrice génératrice G est une (k, n)-matrice
- la matrice de contrôle H est une (r, n)-matrice
- les vecteurs-colonnes de H sont les entiers en binaire de 1 à n dans n'importe quel ordre
- un code de Hamming détecte 2 erreurs et corrige 1 erreur.

  (*cf.* TP n° 4)

Exemple : un code $H_{7,4}$

- on définit la matrice génératrice G de l'application linéaire f :
  $f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_4,\ x_1 + x_3 + x_4,\ x_1,\ x_2 + x_3 + x_4,\ x_2,\ x_3,\ x_4)$

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

Exemple : un code $H_{7,4}$

- on définit la matrice génératrice G de l'application linéaire f :
  $f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_4,\ x_1 + x_3 + x_4,\ x_1,\ x_2 + x_3 + x_4,\ x_2,\ x_3,\ x_4)$

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

- par permutation des colonnes on fait apparaître la matrice Id_4 :

$$G' = (P\ \ Id_4) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ avec } P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Exemple : un code $H_{7,4}$

- on définit la matrice génératrice G de l'application linéaire f :
  $f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_4,\ x_1 + x_3 + x_4,\ x_1,\ x_2 + x_3 + x_4,\ x_2,\ x_3,\ x_4)$

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

- par permutation des colonnes on fait apparaître la matrice Id_4 :

$$G' = (P\ \ Id_4) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ avec } P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- en prenant $H' = (Id_3\ {}^tP)$ on obtient que $H' \times {}^tG' = [0]_{3,4}$:

$$H' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Exemple : un code $H_{7,4}$

- on définit la matrice génératrice G de l'application linéaire f :
  $f(x_1,x_2,x_3,x_4) = (x_1+x_2+x_4,\ x_1+x_3+x_4,\ x_1,\ x_2+x_3+x_4,\ x_2,\ x_3,\ x_4)$

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

- par permutation des colonnes on fait apparaître la matrice Id_4 :

$$G' = (P\ \ Id_4) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ avec } P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- en prenant $H' = (Id_3\ {}^tP)$ on obtient que $H' \times {}^tG' = [0]_{3,4}$:

$$H' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$H' \times {}^tG' = (Id_3\ {}^tP) \times {}^t(P\ Id_4) = (Id_3\ {}^tP) \times \begin{pmatrix} {}^tP \\ {}^tId_4 \end{pmatrix} = (Id_3 \times {}^tP) + ({}^tP \times Id_4) = [0]_{3,4}$$

Exemple : un code $H_{7,4}$

- on définit la matrice génératrice G de l'application linéaire f :
  $f(x_1,x_2,x_3,x_4) = (x_1+x_2+x_4,\ x_1+x_3+x_4,\ x_1,\ x_2+x_3+x_4,\ x_2,\ x_3,\ x_4)$

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

- par permutation des colonnes on fait apparaître la matrice Id_4 :

$$G' = (P\ \ Id_4) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ avec } P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- en prenant $H' = (Id_3\ {}^tP)$ on obtient que $H' \times {}^tG' = [0]_{3,4}$:

$$H' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$H' \times {}^tG' = (Id_3\ {}^tP) \times {}^t(P\ Id_4) = (Id_3\ {}^tP) \times \begin{pmatrix} {}^tP \\ {}^tId_4 \end{pmatrix} = (Id_3 \times {}^tP) + ({}^tP \times Id_4) = [0]_{3,4}$$

- on défait la permutation sur les colonnes pour obtenir la matrice de contrôle H :

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

Exemple (suite) : codage et décodage

- à l'issue du codage de source, on groupe la sortie binaire par paquets de longueur 4
- soit le paquet-message $\mathcal{M} = (1001)$ à encoder
- on code $\mathcal{M}$ en un message codé $\mathcal{C}$

$$\mathcal{C} = \mathcal{M}\ .\ G = (0011001)$$

- on envoie $\mathcal{C}$ au récepteur qui reçoit le message reçu $\mathcal{R}$
- pour détecter une éventuelle erreur, il contrôle le message $\mathcal{R}$:

$$H\ .\ {}^t\mathcal{R}\ =\ [0]_{r,1}\ \ ?$$

- si oui, il conclut à l'absence d'erreur car $\mathcal{C} = (0011001)$ est un mot du code
- il peut alors décoder $\mathcal{C}$ en $\mathcal{M}$ conformément à G :

$$\mathcal{M} = \pi_{3,5,6,7}(\mathcal{C}) = (1001)$$

Distance et poids de Hamming

Soient x et y deux mots binaires de longueur n

- distance de Hamming entre $x = x_1 \ldots x_n$ et $y = y_1 \ldots y_n$:

$$d_H(x,y) = card(\{i, 1 \leq i \leq n \text{ et } x_i \neq y_i\})$$

c'est le nombre de positions où 2 mots binaires diffèrent

Distance et poids de Hamming

Soient x et y deux mots binaires de longueur n

- distance de Hamming entre $x = x_1 \dots x_n$ et $y = y_1 \dots y_n$:

$$d_H(x,y) = card(\{i, 1 \leq i \leq n \text{ et } x_i \neq y_i\})$$

c'est le nombre de positions où 2 mots binaires diffèrent

- poids de Hamming du mot x :

$$w_H(x) = d_H(0,x) = card(\{i, 1 \leq i \leq n \text{ et } x_i = 1\})$$

c'est la distance du mot x au mot $(0,\dots,0)$, elle coïncide avec le nombre de 1 dans x

Exemple

Distance et poids de Hamming

Soient x et y deux mots binaires de longueur n

- distance de Hamming entre $x = x_1 \dots x_n$ et $y = y_1 \dots y_n$:

$$d_H(x,y) = card(\{i, 1 \leq i \leq n \text{ et } x_i \neq y_i\})$$

c'est le nombre de positions où 2 mots binaires diffèrent

- poids de Hamming du mot x :

$$w_H(x) = d_H(0,x) = card(\{i, 1 \leq i \leq n \text{ et } x_i = 1\})$$

c'est la distance du mot x au mot $(0,\dots,0)$, elle coïncide avec le nombre de 1 dans x

Exemple $x = (0,1,1,0,1,1,0)$ et $y = (1,1,1,0,0,1,1)$, poids $w_H(x) = 4$ et $w_H(y) = 5$ et distance $d_H(x,y) = 3$

Boules de Hamming

Soit x un mot binaire de longueur n,

- boule de centre x et de rayon e avec $0 \leq e \leq n$:

$$B_e(x) = \{z : z \in \{0,1\}^n,\ d_H(x,z) \leq e\}$$

ce sont les mots binaires de longueur n qui diffèrent de x d'au plus e positions

Boules de Hamming

Soit x un mot binaire de longueur n,

- boule de centre x et de rayon e avec $0 \leq e \leq n$:

$$B_e(x) = \{z : z \in \{0,1\}^n,\ d_H(x,z) \leq e\}$$

ce sont les mots binaires de longueur n qui diffèrent de x d'au plus e positions

- nombre d'éléments de la boule de centre x et de rayon e :

$$|B_e(x)| = \sum_{i=0}^{e} \binom{n}{i}$$

Boules de Hamming

Soit x un mot binaire de longueur n,

- boule de centre x et de rayon e avec $0 \leq e \leq n$:

$$B_e(x) = \{z : z \in \{0,1\}^n,\ d_H(x,z) \leq e\}$$

ce sont les mots binaires de longueur n qui diffèrent de x d'au plus e positions

- nombre d'éléments de la boule de centre x et de rayon e :

$$|B_e(x)| = \sum_{i=0}^{e} \binom{n}{i}$$

- les codes de Hamming sont des codes linéaires parfaits
  en effet, les boules centrées sur les mots de code forment une partition de l'ensemble des suites binaires de longueur n.

Exemple

Boules de Hamming

Soit x un mot binaire de longueur n,

- boule de centre x et de rayon e avec $0 \leq e \leq n$:

$$B_e(x) = \{z : z \in \{0,1\}^n,\ d_H(x,z) \leq e\}$$

ce sont les mots binaires de longueur n qui diffèrent de x d'au plus e positions

- nombre d'éléments de la boule de centre x et de rayon e :

$$|B_e(x)| = \sum_{i=0}^{e} \binom{n}{i}$$

- les codes de Hamming sont des codes linéaires parfaits
  en effet, les boules centrées sur les mots de code forment une partition de l'ensemble des suites binaires de longueur n.

Exemple La boule de centre $x = (0,1,1,0,1,1,0)$ et de rayon 2 contient $z = (0,1,1,0,0,1,1)$ mais pas $y = (1,1,1,0,0,1,1)$.
Cette boule contient $\sum_{i=0}^{2} \binom{7}{i} = 1 + 7 + 21 = 29$ éléments.

Distance minimale, poids minimal

Soit C un code linéaire,

- la distance minimale du code est le minimun des distances entre 2 mots distincts du code :

$$d_{min}(C) = \min(\{d_H(x,y), x \in C, y \in C, x \neq y\})$$

Distance minimale, poids minimal

Soit C un code linéaire,

- la distance minimale du code est le minimun des distances entre 2 mots distincts du code :

$$d_{min}(C) = \min(\{d_H(x,y), x \in C, y \in C, x \neq y\})$$

- le code peut ainsi détecter jusqu'à $d_{min}(C) - 1$ erreurs
- le code peut ainsi corriger jusqu'à $\lfloor \frac{d_{min}(C)-1}{2} \rfloor$ erreurs
  (*cf.* TD n° 2)

Distance minimale, poids minimal

Soit C un code linéaire,

- la distance minimale du code est le minimun des distances entre 2 mots distincts du code :

$$d_{min}(C) = \min(\{d_H(x,y), x \in C, y \in C, x \neq y\})$$

- le code peut ainsi détecter jusqu'à $d_{min}(C) - 1$ erreurs
- le code peut ainsi corriger jusqu'à $\lfloor \frac{d_{min}(C)-1}{2} \rfloor$ erreurs
  (*cf.* TD n° 2)
- le poids minimal est le plus petit des poids des mots du code :

$$w_{min}(C) = \min(\{w(x), x \in C, x \text{ non-nul}\})$$

- les notions de distance minimale et de poids minimal coïncident pour un code linéaire

Exemple

Distance minimale, poids minimal

Soit C un code linéaire,

- la distance minimale du code est le minimun des distances entre 2 mots distincts du code :

$$d_{min}(C) = \min(\{d_H(x,y), x \in C, y \in C, x \neq y\})$$

- le code peut ainsi détecter jusqu'à $d_{min}(C) - 1$ erreurs
- le code peut ainsi corriger jusqu'à $\lfloor \frac{d_{min}(C)-1}{2} \rfloor$ erreurs
  (*cf.* TD n° 2)
- le poids minimal est le plus petit des poids des mots du code :

$$w_{min}(C) = \min(\{w(x), x \in C, x \text{ non-nul}\})$$

- les notions de distance minimale et de poids minimal coïncident pour un code linéaire

Exemple La distance minimale d'un code de Hamming est 3.
Il peut détecter 2 erreurs et en corriger 1.

Détection d'erreur et correction

- un code de Hamming corrige 1 erreur due au bruit
- la matrice de contrôle a pour rôle de vérifier que le message reçu $\mathcal{R}$ n'a pas été altéré, il doit vérifier :

$$H \,.\, {}^t\mathcal{R} = [0]_{r,1}$$

$$\Leftrightarrow \quad \mathcal{R} = \mathcal{C}$$

- sinon, une erreur a altéré $\mathcal{C}$ et le syndrome $\mathcal{S}$ de l'erreur est :

$$\mathcal{S} = H \,.\, {}^t\mathcal{R} \neq [0]_{r,1}$$

- dans H, il existe forcément un vecteur-colonne H_i avec $1 \leq i \leq n$ tel que :

$$\mathcal{S} = H_i$$

- on en déduit que l'erreur a porté sur le i^e bit de $\mathcal{C}$ et on peut corriger $\mathcal{R}$ en $\mathcal{C}$

Codes Crc

- Codes de Redondance Cyclique *(Cyclic Redundancy Check)* inventés en 1961
- utilisés dans les réseaux informatiques (satellites, clef Wep, Gsm, formats `zip` et `rar` *etc*)
- un tel code est caractérisé par un polynôme générateur P_g de degré r :

$$P_g = x^r + \sum_{i=0}^{r-1} g_i x^i$$

Codes Crc

- Codes de Redondance Cyclique *(Cyclic Redundancy Check)* inventés en 1961
- utilisés dans les réseaux informatiques (satellites, clef Wep, Gsm, formats `zip` et `rar` *etc*)
- un tel code est caractérisé par un polynôme générateur P_g de degré r :

$$P_g = x^r + \sum_{i=0}^{r-1} g_i\, x^i$$

- un mot binaire $m = m_{k-1} \ldots m_0$ est représenté par un polynôme P_m :

$$P_m = \sum_{i=0}^{k-1} = m_i\, x^i$$

Codes Crc

- Codes de Redondance Cyclique *(Cyclic Redundancy Check)* inventés en 1961
- utilisés dans les réseaux informatiques (satellites, clef Wep, Gsm, formats `zip` et `rar` *etc*)
- un tel code est caractérisé par un polynôme générateur P_g de degré r :

$$P_g = x^r + \sum_{i=0}^{r-1} g_i\, x^i$$

- un mot binaire $m = m_{k-1} \ldots m_0$ est représenté par un polynôme P_m :

$$P_m = \sum_{i=0}^{k-1} = m_i\, x^i$$

- m codé par $c = m_{k-1} \ldots m_0\, c_{r-1} \ldots c_0$ où $c_{r-1} \ldots c_0$ représente le reste de la division euclidienne de $x^r.\, P_m$ par P_g :

$$P_c = P_m.\, x^r + (P_m.\, x^r \mod P_g)$$

Codes Crc

- Codes de Redondance Cyclique *(Cyclic Redundancy Check)* inventés en 1961
- utilisés dans les réseaux informatiques (satellites, clef Wep, Gsm, formats `zip` et `rar` *etc*)
- un tel code est caractérisé par un polynôme générateur P_g de degré r :

$$P_g = x^r + \sum_{i=0}^{r-1} g_i\, x^i$$

- un mot binaire $m = m_{k-1} \ldots m_0$ est représenté par un polynôme P_m :

$$P_m = \sum_{i=0}^{k-1} = m_i\, x^i$$

- m codé par $c = m_{k-1} \ldots m_0\, c_{r-1} \ldots c_0$ où $c_{r-1} \ldots c_0$ représente le reste de la division euclidienne de $x^r.\, P_m$ par P_g :

$$P_c = P_m.\, x^r + (P_m.\, x^r \mod P_g)$$

- à la réception, on contrôle que $P_c \mod P_g = 0$ (à défaut, l'ARQ opère)
- le décodage consiste à retirer les r bits de redondance

Exemple

Codes Crc

- Codes de Redondance Cyclique *(Cyclic Redundancy Check)* inventés en 1961
- utilisés dans les réseaux informatiques (satellites, clef Wep, Gsm, formats `zip` et `rar` *etc*)
- un tel code est caractérisé par un polynôme générateur P_g de degré r :

$$P_g = x^r + \sum_{i=0}^{r-1} g_i\, x^i$$

- un mot binaire $m = m_{k-1} \ldots m_0$ est représenté par un polynôme P_m :

$$P_m = \sum_{i=0}^{k-1} = m_i\, x^i$$

- m codé par $c = m_{k-1} \ldots m_0\, c_{r-1} \ldots c_0$ où $c_{r-1} \ldots c_0$ représente le reste de la division euclidienne de $x^r.\, P_m$ par P_g :

$$P_c = P_m.\, x^r + (P_m.\, x^r \mod P_g)$$

- à la réception, on contrôle que $P_c \mod P_g = 0$ (à défaut, l'ARQ opère)
- le décodage consiste à retirer les r bits de redondance

Exemple

Soit $P_g = x^2 + x + 1$ avec le mot $m = 10110$ correspondant à $P_m = x^4 + x^2 + x$
alors $x^r.\, P_m = x^2.\,(x^4 + x^2 + x) = x^6 + x^4 + x^3$
comme $(x^6 + x^4 + x^3) \mod x^2 + x + 1 = x$ (*cf. division de polynôme*)

le mot m est codé en $c = 10110\ 10$.

Une multitude de codes correcteurs

Le code du Minitel est le code de Hamming $(128, 120, 4)$ étendu

15 octets transmis plus 1 octet de redondance (rendement : 94%)

En plus de ceux étudiés, voici les codes correcteurs les plus classiques :

- codes de Reed-Solomon (codes CIRC des CD audio)
- codes de Golay
- codes de Reed-Muller
- codes convolutifs
- *etc*

A suivre ...