# Théorie de l'information
# Codes correcteurs

Florent Devin

EISTI

28 avril 2009

# Introduction

# Résumé

1 **Introduction**
- Introduction
- Détection/correction d'erreurs
- Notations mathématiques

2 **Codes détecteurs**
- Bit de parité

3 **Codes correcteurs**

4 **Codes linéaires**

5 **D'autres codes correcteurs**

# Introduction

- Distinction entre code correcteur, cryptographie et compression
- Code correcteur :
  - Détecte une erreur dans un message
  - Peut corriger une erreur dans un message
- Cryptographie
  - Code un message
  - Peut compresser ce message
- Compression
  - diminue le coût de stockage
  - diminue le coût de transmission

# Application

## Auto correction classique

- Barrettes mémoire : Hamming
- Image de sondes spatiales (en niveau de gris) : Reed - Muller
- Image couleur de sondes spatiales : Golay
- Lecteur CD : CIRC / Reed - Solomon
- Communication sans fil : beaucoup d'algorithme

# Restrictions et définitions

- Messages transmis : découper en blocs (**mots**)
- Mots : longueur $n$
- Alphabet : $\{0, 1\}$
- Code : sous ensemble $C$ de l'ensemble $\{0, 1\}^n$ (appelé $V$)
- $n$ : longueur de $C$

# Contexte

- Soit $\mathcal{T}$ : l'ensemble de transmission
- $\mathcal{T} = \{0 \to 0, 0 \to 1, 1 \to 0, 1 \to 1\}$
- Soit $\mathcal{B} = \{0 \to 0, 1 \to 1\}$
- Soit $\mathcal{M} = \{0 \to 1, 1 \to 0\}$ appelé $p$
- $P(\mathcal{B}) + P(\mathcal{M}) = 1$
- Probabilité d'erreur sur une transmission de deux bits : $1 - (1 - p)^2$
- Probabilité de $q$ erreurs sur une transmission de $t$ bits : $p^q(1 - p)^{t-q}$

# Résumé

1 **Introduction**
- Introduction
- Détection/correction d'erreurs
- Notations mathématiques

2 **Codes détecteurs**
- Bit de parité

3 **Codes correcteurs**

4 **Codes linéaires**

5 **D'autres codes correcteurs**

# Introduction

- Réception : suite de bits
- Besoin de découper pour extraire les mots
- Utilisation "délicate" des codes de longueur variable

## Exemple : Huffman

Soit la chaîne : "Bonjour monde", un code de huffman est :

```
11000001111111001011010100011100 00111101100110100
```

Que se passe-t-il si on reçoit :

```
11111001111111001011010100011100 00111101100110100
```

Le message reçu est alors tout autre : "jejBur monde"

# Principe de codage/décodage

- Codage
  - Découpage du message en blocs de $m$ bits
  - Codage de chaque blocs en bloc de $n$ bits
  - Concaténation des blocs obtenus
  - Transmission
- Décodage
  - Découpage du message en blocs de taille $n$
  - Décodage des blocs selon la méthode de la distance minimale
  - Extraction des blocs de $m$ bits
  - Assemblage du message

# Distance

- Soit deux mots :
  - $x = (x_1, x_2, \ldots, x_n)$
  - $y = (y_1, y_2, \ldots, y_n)$
  - distance de **Hamming** $d(x, y)$ : nombre d'indice $i | x_i \neq y_i$
- $d$ : distance minimale

# Propriétés

- Positivité :
  - $d(x,y) \geq 0 \quad \forall (x,y)$
  - $d(x,y) = 0 \iff x = y$
- Symétrie :

$$d(x,y) = d(y,x)$$

- Inégalité :

$$d(x,y) \leq d(x,z) + d(z,y) \quad \forall (x,y,z)$$

# Principe du décodage

## Décodage à distance minimum

- Mot reçu : $r$
- Mot code : $c$
- $c$ est le plus proche de $y$ quand $d(x, y) = min$
- Application du maximum de vraisemblance

# n-correcteur

Un code est **n-correcteur** quand toute erreur d'au plus $n$ bits est corrigée correctement

# Distance et capacité

Soit

- C un code de distance $d$
- $w'$ un mot reçu
- $e$ nombre d'erreur de $w'$ par rapport à $w$

- Si $e < d$ alors C peut détecter l'erreur
- Si $e < \frac{d}{2}$ alors C peut corriger l'erreur

# Résumé

1 Introduction
- Introduction
- Détection/correction d'erreurs
- Notations mathématiques

2 Codes détecteurs
- Bit de parité

3 Codes correcteurs

4 Codes linéaires

5 D'autres codes correcteurs

# Règles de calcul dans $(\mathbb{Z}/2\mathbb{Z})^n$

- $\mathbb{Z}/2\mathbb{Z}$ : corps (muni des opérations binaire $+$, x)
- $(\mathbb{Z}/2\mathbb{Z})^n$ : mots binaires de longueur $n$
- Définitions de deux lois :
  - L'addition : $+ : (\mathbb{Z}/2\mathbb{Z})^n x (\mathbb{Z}/2\mathbb{Z})^n \rightarrow (\mathbb{Z}/2\mathbb{Z})^n$

$$(a_1, a_2, \cdots, a_n) + (b_1, b_2, \cdots, b_n) = (a_1 + b_1, a_2 + b_2, \cdots, a_n + b_n)$$


Pas de report de retenue

  - La multiplication : .

# Règles de calcul dans $(\mathbb{Z}/2\mathbb{Z})^n$

- $\mathbb{Z}/2\mathbb{Z}$ : corps (muni des opérations binaire $+$, x)
- $(\mathbb{Z}/2\mathbb{Z})^n$ : mots binaires de longueur $n$
- Définitions de deux lois :
  - L'addition $+$
  - La multiplication : $: . : \mathbb{Z}/2\mathbb{Z}x(\mathbb{Z}/2\mathbb{Z})^n \rightarrow (\mathbb{Z}/2\mathbb{Z})^n$

$$\lambda.(a_1, a_2, \cdots, a_n) = (\lambda.a_1, \lambda.a_2, \cdots, \lambda.a_n)$$

# Remarques

- $0_n = (0, 0, \cdots, 0)$
- $1_n = (1, 1, \cdots, 1)$
- $\forall x \in (\mathbb{Z}/2\mathbb{Z})^n x + x = 0_n \Rightarrow x = -x$
- $x + y = z \iff x = y + z$
- Multiplication matricielle :

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

# Applications linéaires

## Notation

- $\mathcal{M}_{p,n}$ : matrice de taille $p \times n$ dans $(\mathbb{Z}/2\mathbb{Z})^n$

## Définition

Une application f : $(\mathbb{Z}/2\mathbb{Z})^p \to (\mathbb{Z}/2\mathbb{Z})^n$ est **linéaire** si et seulement si pour tous mots $x$,$y$ de $(\mathbb{Z}/2\mathbb{Z})^p$ on a $f(x+y) = f(x) + f(y)$

# Codes détecteurs

# Résumé

1 Introduction
- Introduction
- Détection/correction d'erreurs
- Notations mathématiques

2 Codes détecteurs
- Bit de parité

3 Codes correcteurs

4 Codes linéaires

5 D'autres codes correcteurs

# Bit de parité (paire)

- Signature : nombre total de bit à 1 : pair
- Transmission du vocabulaire cible + signature
- Réception : calcul du nombre de bit à 1
  - Nombre pair : pas de problème détecté
  - Nombre impair : problème détecté

# Exemple

## Parité paire

```
Message initial : 0000000100100011
Phase de découpage (bloc de 4 bits) :
0000 0001 0010 0011
Phase de codage (rajout du bit de parité) :
00000 00011 00101 00110
Phase de transmission (message assemblé)
0000000011001010 0110
```

# Exemple simple

## Répétition

- Code à transmettre : 0110
- Code transmis : 000111111000
- Code reçu : $xxyzyzyzzyxx$
- Permet de corriger les erreurs dans la plupart des cas
- Augmente considérablement le temps de transmission

# Codes correcteurs

# Codes correcteurs

- Plusieurs familles
  - Codes en bloc : codage/décodage ne dépend que du bloc en cours (Famille de type linéaires, cycliques ou non)
  - Codes entrelacé : codage/décodage dépend d'informations déjà transmises (famille de type convolutifs, récursifs ou non)
- Préférence pour les codes linéaires dans les transmissions (quand ceux ci sont utilisés...)

# Codage de double parité

- Association à chaque bloc, d'un bit de parité
- Association d'un bloc de parité pour $x$ bloc transmis
- Codage peu utilisé

00000000 01010101 11111111
11111111 00000000 01010101
01010101 11111111 00000000
10101010 10101010 10101010

# Codes linéaires

# Codes linéaires

## Notation : C(n,k,d)

- $F_2$ : corps à deux éléments (0, 1)
- $F_2^n$ mots de longueur $n$
- $C \subset F_2^n$ : code linéaire de longueur $n$
- $k$ : dimension du sous espace vectoriel $C$
- $2^k$ : nombre de mots du code $C$
- $w(x)$ : poid du mot $x$
- $d$ : minimum des poids de $C$

# Code linéaires

## Explications : code(n,k,d)

- $n$ : dimension du vocabulaire transmis (nombre de symbole des mots)
- $k$ : dimension du vocabulaire initial
- $d$ : distance minimale entre deux mots cibles
- $\frac{k}{n}$ : rendement du code
- $\frac{d}{n}$ : fiabilité du code
- Permet de détecter $d-1$ erreurs
- Permet de corriger $\lfloor \frac{d-1}{2} \rfloor$ erreurs

# Borne de Singleton

$$d + k \leq n + 1$$

On ne peut avoir une bonne correction et un vocabulaire riche sur une longueur fixe de code

# Matrice génératrice

- Code linéaire : représentable par une matrice génératrice $G$
- $G$ de taille $k \times n$ ($k$ lignes, $n$ colonnes)
- Lignes de $G$ : base du code $C$
- Si $m$ vecteur ligne de $k$ composantes (mot original)
- $mG \in C$ : mot de $n$ composantes (mot codé)
- $d$ : distance de Hamming la plus petite de toutes les lignes
- Permutation de deux colonnes : code équivalent

# Matrice sous forme systématique

- Si $G = [I_k|A]$ alors $G$ est sous forme systématique
- N'existe pas toujours
- Si existence alors l'écriture est unique
- Début du code : mot initial

## Exemple de matrice

$$\left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right)$$

# Méthodes de corrections

## Deux possibilités

- Par génération de toutes les combinaisons possibles, puis comparaison du mot reçu avec ceux possibles (très long, très volumineux, lent)
- Par utilisation d'une matrice de contrôle et des syndromes associés

# Matrice de contrôle

## Code dual (noté $C^{\perp}$)

- $y = (y_1, y_2, \ldots, y_n) \in \mathbb{F}_q^n$
- $x = (x_1, x_2, \ldots, x_n) \in C$
- $x.y = x_1y_1 + x2y2 + \cdots + x_ny_n = 0$
- $y$ : ensemble des vecteurs de $C^{\perp}$
- $H$ : matrice génératrice de $C^{\perp}$
- $H$ : matrice de contrôle de $C$

# Matrice de contrôle et syndrome

- $G$ : Matrice sous forme systématique ($G = [I_k|A]$)
- $H$ : Matrice de contrôle ($H = [{}^tA|I_{n-k}]$)
- Syndrome de $x$ (noté $\sigma$) : vecteur colonne $\in \mathbb{F}_q^n$
  - $\sigma(x) = H{}^tx$
  - $\sigma(x) = \sum_{i=1}^{n} x_i h_i$
  - $\sigma(x) = 0 \iff x \in C$

# Intérêt du syndrome

- $y$ : mot reçu
- $x$ : mot émis
- $e = y - x$ : erreur de transmission
- $\sigma(y) = \sigma(x + e) = \sigma(e)$
- Application en TD : Hamming

**Propriété importante**

Le syndrome ne dépend que de l'erreur

# D'autres codes correcteurs

# D'autres codes correcteurs

- Reed Muller : Code linéaire permettant de corriger 7 erreurs
- Golay : code cyclique (topologie particulière des codes linéaires)
- BCH (Bose, Ray-Chaudhuri, Hocquenghem) : Code cyclique à plusieurs niveaux
- Reed Solomon : Code par bloc (basé sur les corps Galois)
- Goppa : code de géométrie algébrique (extension des code de Reed - Solomon)
- ...