Parie I - correction d'erreurs ?

- Dans les systèmes électroniques digitaux, l'information est représentée en format binaire : uniquement par des 0 et des 1 (bits)

- Transfert d'information d'un point à un autre : il y a toujours une chance pour qu'un bit soit mal interprété (1 au lieu de 0 et vice versa)
  Cela peut avoir de multiples causes, par exemple :
  - Bruit parasite
  - Défauts au niveau des composants
  - Mauvaise connexion
  - Détérioration due au vieillissement …

- La correction d'erreurs est le procédé utilisé pour :
  - *détecter automatiquement* et *corriger automatiquement* ces bits erronés.

⇒ Au niveau logiciel ou au niveau matériel (pour le haut débit).

Taux d'erreur = Nombre de bits erronés sur le total des bits transférés

- Disquette magnétique : 1 bit erroné tous les milliards de bits transférés
  - Un million de bits/s (125 Ko/s) : 1 bit erroné toutes les 16.6 minutes
  - Lecteurs actuels (5 Mo/s) : 1 bit erroné toutes les 25 secondes

- CD-ROM optique : 1 bit erroné tous les 100 000 bits (12.5 Ko) transférés ⇒ 6300 erreurs dans un disque

- Audio DAT : 10^{-5} bits faux (à 48kHz) ⇒ 2 erreurs chaque seconde

- Ligne téléphonique : 10^{-4} à 10^{-6} bits erronés

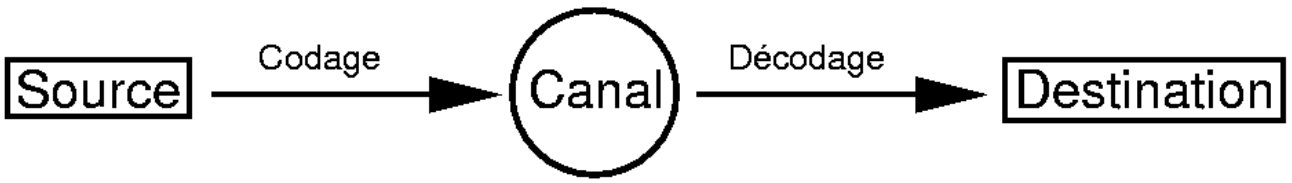
- Communicateurs par fibres optiques : 10^{-9} bits erronés

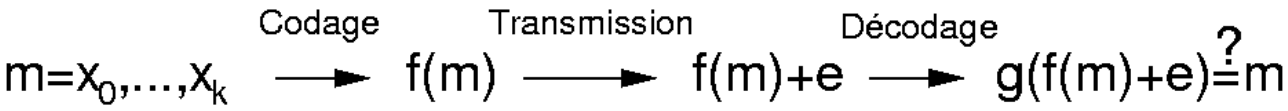
- Mémoires à semi-conducteurs : $< 10^{-9}$

Plan du cours

- Introduction : taux d'erreur
- Notion de Code
- Concepts de base de la correction
- Exemples introductifs
  - Commande automatique d'un bras de robot
  - Contrôle de parité
  - Parité longitudinale et transversale
- Généralisation : Code linéaire
  - Exemple : la parité
  - Code de Hamming
  - Codes cycliques et Reed-Solomon
- Autres codes et applications

Notion de Code





- Le code doit répondre à différents critères :
  - Sécurité de l'information : cryptage + authentification
  - Rentabilité : compression des données
  - Tolérance aux fautes : correction/détection d'erreurs

Concepts de base de la correction

- Un groupe de bits dans un ordinateur est un « mot ».
- Chaque bit est considéré comme étant une « lettre ».

- La langue française nous permet une analogie :
  - Toutes les combinaisons possibles de l'alphabet ne sont pas des mots de la langue. Les seuls mots autorisés sont ceux énumérés dans un dictionnaire.
  - Des erreurs qui se produisent en transmettant ou en stockant des mots français peuvent être détectées en déterminant si le mot reçu est dans le dictionnaire.
  - S'il ne l'est pas, des erreurs peuvent être corrigées en déterminant quel mot français existant est le plus proche du mot reçu.

⇒ Idée pour la correction d'erreurs :
  - Ajouter des lettres supplémentaires (redondantes).
  - Ces lettres supplémentaires donnent une structure à chaque mot.
  - Si cette structure est changée par des erreurs, les changements peuvent être détectés et corrigés.

Commande automatique d'un bras de robot

Code	Haut	Bas	Droite	Gauche
C_2	00	10	01	11

- C_2 : économique

☹ Impossible de détecter une erreur :
  - Si 00 est envoyé et 01 reçu, « droite » est interprété au lieu de « haut »

Commande automatique d'un bras de robot

Code	Haut	Bas	Droite	Gauche
C_2	00	10	01	11
C_3	000	110	011	101

- C_3 : détecte si 1 seul bit est faux car 2 mots distincts diffèrent d'au moins 2 bits (distance de Hamming)
  - ☺ Si 000 est envoyé et 001 est reçu : erreur
  - ☹ Pas de correction : si 001 est reçu, avec une seule erreur il peut tout aussi bien provenir de 000 que 011 ou encore 101 !!!

Commande automatique d'un bras de robot

Code	Haut	Bas	Droite	Gauche
C_2	00	10	01	11
C_3	000	110	011	101
C_6	000000	111000	001110	110011

- C_6 : distance minimale entre deux mots : 3

  ⇒Détecte 2 erreurs

  ☺Corrige 1 erreur :
  - Avec au plus un bit erroné, on choisit le mot de code (du dictionnaire) le plus proche
  - Ex: 000001 est reçu, alors 000000 est le mot admissible le plus proche

Définition - Notation

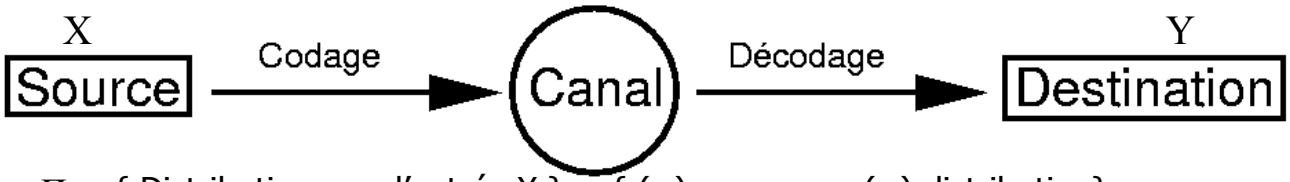
- **V** = **alphabet** = ensemble fini de symboles.
  Ex1: V= {0,1} Ex2: V={octets}

- Code de **longueur *n*** sur V = sous ensemble de V^n.
  - Les éléments du code sont appelés mots de code.

- **Codage** par blocs de source de taille ***k*** (k < n)
  - Φ : V^k --> V^n : fonction de codage, injective
  - $\Phi(x_1, ..., x_k) = y_1, ..., y_k, ..., y_n$
  - **r** = n – k = nombre de symboles de redondance
  - **Rendement**: **R = k/n** (0< R ≤1)

- **Code(n, k) sur V** = sous-ensemble de V^n de cardinal $|V|^k$.

Lien avec entropie: capacité de canal



- $\underline{\Pi}$ = { Distributions sur l'entrée X } = { $(p_i)_{i=1..|V|}$ avec (p_i) distribution}

- $p_{i|k}$ = Prob (Y = s_i | X=s_k) : caractérise les probabilités d'erreurs lors de la transmission sur le canal sans mémoire.
  - Canal sans erreur ssi ($p_{i,i}$=1 et $p_{i,k\neq i}$=0)

- $P(Y=s_i) = \sum_{k=1..|V|} p_k \cdot p_{i|k}$

- **Déf**: ***Capacité de canal*** : $C = Max_{p \in \underline{\Pi}} H(X) - H(X \mid Y)$
  - i.e. ce qu'il reste à découvrir de l'entrée X du canal lorsqu'on connait la sortie Y.

- On a aussi: C = Max H(Y) – H(Y | X). Cas extrêmes:
  - H(Y)=H(Y|X) : sortie indépendante de l'entrée.
  - H(Y | X) = 0 : canal sans erreur.

Deuxième théorème de Shannon

- **Théorème**: Soit un canal de capacité C. Alors pour tout $\varepsilon>0$:
  $\exists$ un code(n, k) de probabilité d'erreur $<\varepsilon$ ssi $0 \leq k/n < C$.

  i.e. la capacité de canal C est une limite supérieure au rendement..

- Exemple : canal binaire symétrique (BSC).
  - $C_{BSC} = 1 + p.\log_2 p + (1-p).\log_2(1-p)$
  - Si $p = 0.5$ => $C_{BSC} = 0$: pas de code correcteur possible.
  - Si $p \neq 0.5$ => il existe un code permettant de communiquer sans erreur
    - Mais son rendement est borné par C.

- Problème fondamental du codage:
  - Construire des codes de rendement maximal pour une longueur n fixée.

PREMIERS EXEMPLES
Codes de Parité

Contrôle de parité

- Une technique de base pour construire un code détecteur
  1. Découper le message en mots de 7 bits $m=[x_0,\ldots, x_6]$
  2. Ajouter aux mots leur parité : $f(m)=[x_0,\ldots, x_6, p]$
     - Le nombre de 1 dans le mot est soit pair (p = 0) soit impair (p = 1)
     - Calculée par : $\mathbf{x_7 = p = \sum_{i=0..6} x_i \bmod 2}$

- Standard n°5 du Comité Consultatif International Télégraphique et Téléphonique (CCITT 5) : le plus populaire, utilisé par exemple aux USA.

Lettre	Codage de base sur 7 bits	Mot de code avec bit de parité
a	1000 001	1000 001**0**
e	1010 001	1010 001**1**
u	0110 101	0110 101**0**

☺ Permet de détecter tout nombre impair d'erreurs

Parité longitudinale et transversale

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}	a_{06}	P_0
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	P_1
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	P_3
C_0	C_1	C_2	C_3	C_4	C_5	C_6	N

1. Mots sur 3*7 = 21 bits
2. Parité par ligne : P_i
3. Parité par colonne : C_j
4. Parité globale : N

- Détecte 2 ou tout nombre impair d'erreurs
- Corrige 1 erreur
  - Un des a_{ij} est faux : le recalcul de P_i et C_j donne l'emplacement ij
  - P_i, C_j et N sont recalculés
- Détecte si il y a 2 erreurs, mais ne permet pas de corriger

Rendement

- Nombre de bits de message / Nombre de bits transmis
  - Parité : Rendement = 7/8 = 87.5 %
  - Parité long. & transv. : Rendement = 21/32 ≈ 65 %

« Parités » usuelles pour la simple détection d'erreur

- LUHN10 pour les cartes bleues
  - Doubler un chiffre sur deux du n° de carte bleue
  - Ajouter les autres chiffres et les (doubles modulo 9) obtenus
  - Le résultat doit être 0 modulo 10 pour une carte valide

- Clefs (sécurité sociale, RIB, etc.)
    - Sécu : clef calculée pour le numéro + la clef soit nul modulo 97
    - RIB : clef calculée pour que (numéro||clef)$_{5+5+11+2 \text{ chiffres}}$ soit nul modulo 97
  - IBAN : lettres + 9 et la somme doit faire 1 modulo 99

- De 1972 à 2077: Code ISBN sur les livres sur 10 chiffres :
  - $\sum_{i=1..10}\ i \times a_i \equiv 0$ modulo 11

Code barre EAN-13

- EAN-13 (European Article Numbering)
- Numéro sur 13 chiffres + motif graphique barres noires/blanches

  c_{12} - c_{11} .. c_6 - c_5 .. c_0
- c_0 chiffre de parité calculé comme suit:
  - Soient a = mod 10
    et $b = c_{11} + c_9 + c_7 + c_5 + c_3 + c_1$ mod 10
  - Alors $c_0 = 10 - (a+3b \bmod 10)$
  - Exemple: a= 3+9+1+3+5+7 mod 10 = 8 ; b= 2+9+2+4+6+8 mod 10 = 2; $c_0 = 10 - (a+3b \bmod 10) = 10 - 4 = 6$
- Le code barre graphique code le même numéro:
  - chaque colonne de 2,31mm code un seul chiffre, par 4 barres de largeur différentes chaque colonne est divisée en 7 barres N/B de largeur élémentaire 0,33 mm
- EAN13 permet de détecter des erreurs mais pas de corriger.
- Depuis 2007: Code ISBN sur les livres=EAN-13: $c_{12}c_{11}c_{10}c_9c_8c_7c_6c_5c_4c_3c_2c_1c_0$
  - Avec pour les livres: $c_{12}c_{11}c_{10}$=978
    - Ex: **978-2-10-050692-7**
      c_0=10-[9+8+1+0+0+9+3x(7+2+0+5+6+2) mod 10]=10- 3 = 7
- Extensions: code barre bidimensionnel PDF417: permet de coder jusqu'à 2725 caractères, grâce à un code correcteur de Reed-Solomon