Architecture des Ordinateurs

Mounir T. El Araki

mounir.elarakitantaoui@uic.ac.ma

Plan du cours

- Historique
- Présentation de l'architecture des ordinateurs
- Représentation interne des informations
- Encodage/décodage de l'information
- Circuits logiques
- Mémoires
- Unité centrale de traitement

Encodage

- Encoder une information en assurant son intégrité et sa compression avec des méthodes simples.
- Utilisation des codes pour représenter l'information pour résoudre les problèmes suivants:
  - **Assurer l'intégrité de l'information**
    - (détection et correction d'erreurs)
  - **Minimiser la taille de l'information (compression),**
  - **Garantir la sécurité de l'information** (encryptage/chiffrement).

Code détecteurs et correcteurs d'erreur

- Une information peut subir des modifications involontaires lors de sa transmission ou lors de son stockage en mémoire.
- → Utiliser des codes permettant de détecter ou même de corriger les erreurs.
- → Utiliser des bits supplémentaires (de contrôle) à ceux nécessaire pour coder l'information.
  - Codes auto-vérificateurs (e.g., contrôle de parité),
  - Codes auto-correcteurs (e.g., double parité, hamming, codes polynomiaux).

Contrôle de parité

- Code auto-vérificateur le plus simple.
- A un mot de taille ‘m’, on ajoute 1 bit de parité.
- "parité paire": la valeur du bit de parité est a 1 si le nombre de bit a 1 du mot ‘m+1’ est pair.
  - Exemple en parité paire : 7+1 bits
  - Valide :
    - **1** 1 0 0 1 1 0 0
    - **0** 1 0 0 1 1 0 1
  - Erreur :
    - **0** 1 0 0 1 1 0 0
- Si un bit est change par erreur la parité n'est plus vérifiée. L'erreur est détectée (mais pas corrigée).
  - →Retransmission de l'information.

Contrôle de double parité

- Code obtenu en effectuant un double contrôle de parité.
- A un mot de taille m, on ajoute 1 bit de parité transversal.
- Apres une série de mot, on ajoute 1 mot de parité (longitudinal).
  - **Exemple en double parité impaire : 7+1 bits et série de 4 mots.**

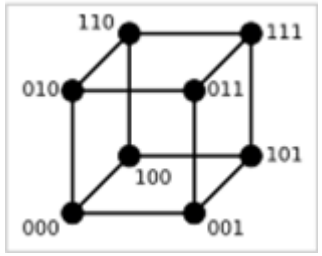
	No de bit							bit de	contrôle
	1	2	3	4	5	6	7	parité	transversal
1. car.= 1	0	1	1	1	0	0	1	0	← faux
2. " = 9	0	1	1	1	0	0	1	1	OK
3. " = 6	0	1	1	0	1	1	0	1	OK
4. " = 8	0	1	1	1	0	0	0	0	OK
bit de parité	1	1	1	1	0	0	1		
contrôle longitudinal	OK	OK	OK	↑ faux	OK	OK	OK		

  - Si un bit est change, l'erreur est détectée et corrigée.

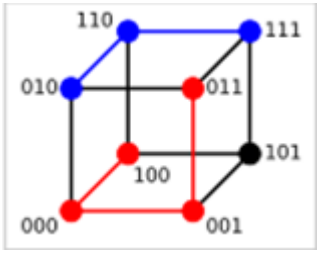
La distance de Hamming

- La distance de Hamming est le nombre de bits à changer pour passer d'une configuration de bis à une autre.
  - Exemple 10010101 & 1001**10**01 à une distance de 2

- Pour n'importe quelle code incluant des membres ayant des distances de Hamming de 2, une erreur d'1 bit peut être détectée. Pourquoi?



Cube binaire de 3 bits pour trouver les distances de Hamming



100 → 011 a une distance de 3 (rouge)
010 → 111 a une distance de 2 (bleu)

Codes de Hamming

- Les codes de Hamming sont utilisés télécommunication et en compression
- [7,4] codes de Hamming binaire
  - Soit notre mot (x_1 x_2 … x_7)
  - x_3, x_5, x_6, x_7 représente le message lui-même.
  - $x_4 := x_5 + x_6 + x_7 \pmod 2$
  - $x_2 := x_3 + x_6 + x_7$
  - $x_1 := x_3 + x_5 + x_7$

$$
\begin{array}{lcl}
(0\,0\,0\,0) & \rightarrow & (0\,0\,0\,0\,0\,0\,0) \\
(0\,0\,0\,1) & \rightarrow & (1\,1\,0\,1\,0\,0\,1) \\
(0\,0\,1\,0) & \rightarrow & (0\,1\,0\,1\,0\,1\,0) \\
(0\,0\,1\,1) & \rightarrow & (1\,0\,0\,0\,0\,1\,1) \\
(0\,1\,0\,0) & \rightarrow & (1\,0\,0\,1\,1\,0\,0) \\
(0\,1\,0\,1) & \rightarrow & (0\,1\,0\,0\,1\,0\,1) \\
(0\,1\,1\,0) & \rightarrow & (1\,1\,0\,0\,1\,1\,0) \\
(0\,1\,1\,1) & \rightarrow & (0\,0\,0\,0\,1\,1\,1) \\
 & \vdots &
\end{array}
$$

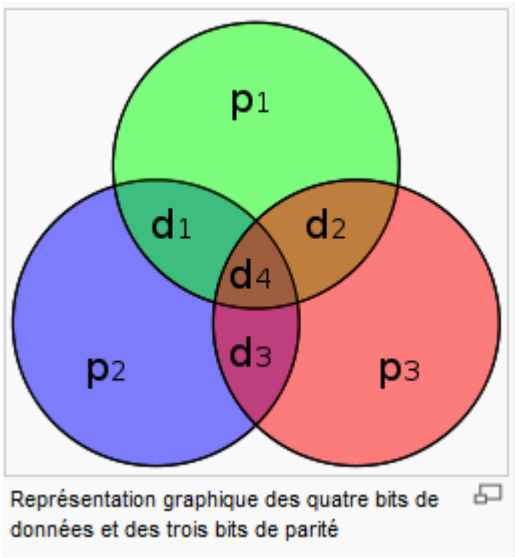
Le code de Hamming [7,4]

- Soit a = $x_4 + x_5 + x_6 + x_7$ (=1 Ssi un de ces bits est en erreur)
- Soit b = $x_2 + x_3 + x_6 + x_7$
- soit c = $x_1 + x_3 + x_5 + x_7$

- Si erreur (assume en plus une) alors

'abc' est la représentation binaire

de l'indice de bit d'erreur.



Représentation graphique des quatre bits de données et des trois bits de parité

- Si (y_1 y_2 … y_7) est le résultat et abc ≠ 000, alors on assume que le bit 'abc' est une erreur et on le change. Si 'abc'=000, on assume pas d'erreur.

Exemple utilisation de L_3

- Suppose (1 0 1 0 0 1 0) est réceptionné.

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

100 est 4 en binaire, donc le message initial était (1 0 1 1 0 1 0).

Compression

- Objectif

  - Diminuer le nombre de bits utilisés pour le stockage et la transmission des informations. Les algorithmes de compression se caractérisent par les facteurs suivants :

    - **le taux de compression,**
    - **la qualité de compression** (avec ou sans perte d'information),
    - **le temps de compression.**

Codage de huffman

- Algorithme de compression sans perte, très connu.
- Réduit le nombre de bits utilisés pour représenter les caractères les plus fréquent,
- Augmente le nombre de bits utilisés pour représenter les caractères peu fréquent,
- Un arbre binaire donne le codage pour chaque caractère.

Codage de Huffman: Exemple simple

- Suppose on a un message qui consiste de 5 symboles, e.g. [►♣♣♠☻►♣☼►☻]
- Comment peut on coder ce message utilisant des 0/1 pour que le message ait une longueur minimale (pour transmission ou sauvegarde)

- 5 symboles → au moins 3 bits
- Pour un encodage simple,
  - la longueur du code est 10*3=30 bits

►	000
♣	001
☻	010
♠	011
☼	100

Codage de Huffman: Exemple simple

- Intuition: Les symboles les plus fréquents doivent avoir des codes plus petits. Seulement, on doit distinguées chaque code, puisqu'ils n'auront pas la même longueur.

- Pour le code de Huffman

Le message sera►♣♣♠☻►►♣☼►☻

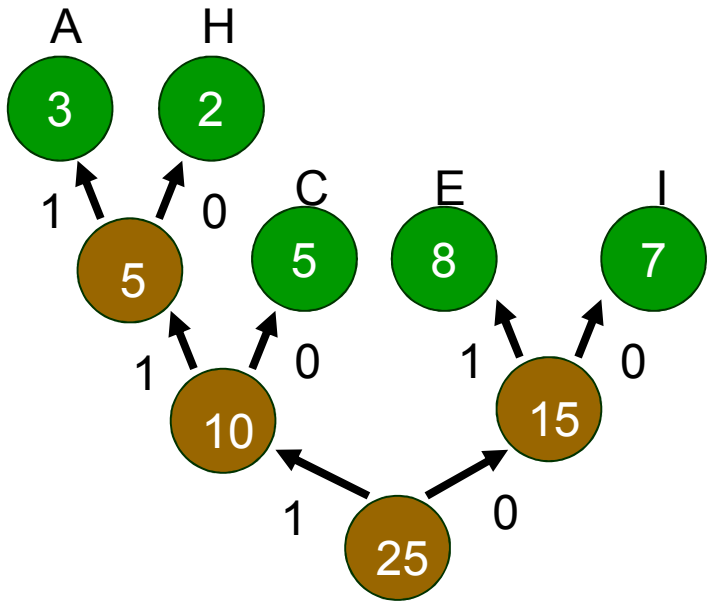
=3*2 +3*2+2*2+3+3=24bits

Symbol	Freq.	Code
►	3	00
♣	3	01
☻	2	10
♠	1	110
☼	1	111

Codage de l'algorithme de Huffman

1. Prendre les symboles les mois probables de l'alphabet

2. Combiner ces deux symboles en un seul symbole , et on répète.



E	=	01
I	=	00
C	=	10
A	=	111
H	=	110

Codage de l'algorithme de Huffman

Caractère	Fréquence	Fixé	Huffman
E	125	0000	110
T	93	0001	011
A	80	0010	000
O	76	0011	001
I	73	0100	1011
N	71	0101	1010
S	65	0110	1001
R	61	0111	1000
H	55	1000	1111
L	41	1001	0101
D	40	1010	0100
C	31	1011	11100
U	27	1100	11101
Total	838	4.00	3.6229