

Logique Combinatoire et Séquentielle

Codage de l'information

Pierre Héroux

Pierre.Heroux@univ-rouen.fr
Université de Rouen

L1 Informatique – EEEA

Plan

1 Définitions

2 Représentation des nombres

- Codage des entiers naturels
- Représentation des entiers négatifs
- Représentation des réels

3 Codage des informations alphanumériques

Définitions

- Soit I un ensemble d'informations

$$I = \{i_1, i_2, i_3, \dots\}$$

- Soit $A = \{a_1, a_2, \dots, a_b\}$ un ensemble fini de symboles (chiffres). A est appelé *alphabet*.
- Une suite ordonnée de symboles $a_i \in A$ est appelé *mot*.
- On appelle *base*, le cardinal de A .

Définitions

- Coder I , consiste à faire correspondre à chaque élément de I un mot m construit sur l'alphabet A .

$$\forall i \in I, m = f(i)$$

- La fonction de codage est bijective. Le décodage consiste à identifier l'information i à partir du mot m

$$i = f^{-1}(m)$$

- Dans un codage à longueur fixe n , tous les mots ont la même longueur n .
- On peut construire b^n mots différents.
- On peut coder b^n informations différentes

Plan

1 Définitions

2 Représentation des nombres

- Codage des entiers naturels
- Représentation des entiers négatifs
- Représentation des réels

3 Codage des informations alphanumériques

Notation positionnelle

- La représentation d'un nombre est un codage
- Un même nombre peut être représenté dans plusieurs bases
- Dans une base b , les symboles de l'alphabet A , représentent les quantités $0, 1, \dots, (b - 1)$
- Un mot de longueur n exprimé une base b

$$(a_{n-1} \dots a_1 a_0)_b = \sum_{i=0}^{n-1} a_i \cdot b^i \quad \forall a_i \in A$$

- Cette formule permet de décoder le nombre exprimé dans la base b .
- Les chiffres de gauche sont dits de poids forts car ils correspondent aux puissances élevées de b .
- Les chiffres de droite sont dits de poids faible.

Codage du nombre A en base b

$$\begin{aligned}
 A &= (a_{n-1}a_{n-2}\cdots a_1a_0)_b \\
 &= \sum_{i=0}^{n-1} a_i.b^i \\
 &= a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots + a_1b + a_0 \\
 &= (a_{n-1}b^{n-2} + a_{n-2}b^{n-3} + \cdots + a_1).b + a_0 \\
 &\vdots \\
 &= ((\cdots (a_{n-1}b + a_{n-2})b + \cdots)b + a_1)b + a_0
 \end{aligned}$$

- Utilisé pour coder A dans la base b .
- On divise le nombre de façon itérative par b .
- À chaque division, le reste représente un chiffre de la représentation, dans l'ordre des poids croissants.

Codage du nombre A en base b

Exemple : $(125)_{10}$ à convertir en base 7

$$125 = 17 \times 7 + 6$$

$$17 = 2 \times 7 + 3$$

$$2 = 0 \times 7 + 2$$

$$(125)_{10} = (236)_7$$

Binaire naturel

Dans le code binaire naturel,

$$A = \{0, 1\} \Rightarrow b = 2$$

Un codage de longueur n , permet de coder 2^n informations (les entiers de 0 à $2^n - 1$).

$$(a_{n-1} \dots a_1 a_0)_2 = \sum_{i=0}^{n-1} a_i \cdot 2^i$$

Exemple

Un codage de longueur 3 permet de coder les entiers naturels de 0 à 7.

entier représenté	a_2	a_1	a_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Codage hexadécimal

L'alphabet comporte 16 symboles

$$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

entier représenté	code
0	0
1	1
...	...
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
...	...
32	20
...	...
100	64

Codage hexadécimal

Permet de condenser la représentation binaire

binnaire	hexadécimal	binnaire	hexadécimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Un mot binaire de 4 bits est converti en un mot de 1 chiffre hexadécimal.

Un mot de n octets ($8n$ bits) est converti en un mot de $2n$ chiffres hexadécimaux.

$$(0110\ 1110\ 0010\ 0011\ 0101\ 1010)_2 = (6E\ 23\ 5A)_{16}$$
$$(7A\ 3F\ 6B)_{16} = (0111\ 1010\ 0011\ 1111\ 0110\ 1011)_2$$

Décimal codé binaire

Chaque chiffre de la représentation décimale est codé en binaire sur 4 bits.

entier représenté	code		
0		0000	
1		0001	
...	...		
10	0001	0000	
11	0001	0001	
...	...		
65	0110	0101	
...	...		
859	1000	0101	1001

Codage utilisé pour les afficheurs 7 segments.

Code de Gray

Codage construit de telle sorte que les représentations de 2 entiers consécutifs ne diffèrent que d'un bit.

entier représenté	code	entier représenté	code
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

Plan

1 Définitions

2 Représentation des nombres

- Codage des entiers naturels
- **Représentation des entiers négatifs**
- Représentation des réels

3 Codage des informations alphanumériques

Bit de signe

Pour coder un entier signé sur n bits, on utilise :

- $n - 1$ bits pour coder la valeur absolue de 0 à $2^{n-1} - 1$
- 1 bit pour coder le signe
 - 0 : nombre positif
 - 1 : nombre négatif

Le codage sur n bits permet donc de coder les entiers de $-2^{n-1} + 1$ à $2^{n-1} - 1$.

Exemple sur 8 bits :

- $(0000\ 0101)_2$ représente +5 ;
- $(1000\ 0101)_2$ représente -5 ;
- $(0000\ 0000)_2$ représente +0 ;
- $(1000\ 0000)_2$ représente -0 ;

Ne respecte pas l'arithmétique binaire

Complément à 1

Pour coder l'opposé d'un entier naturel, on inverse tous ses bits.

Un codage sur n bits permet de coder les entiers de $-2^{n-1} + 1$ à $2^{n-1} - 1$.

- $(0000\ 0101)_2$ représente $+5$;
- $(1111\ 1010)_2$ représente -5 ;
- $(0000\ 0000)_2$ représente $+0$;
- $(1111\ 1111)_2$ représente -0 ;

Respecte l'arithmétique binaire

Complément à 2

Pour obtenir le complément à deux, on ajoute 1 au complément à 1.

Ce codage permet de coder les entiers de -2^{n-1} à $2^{n-1} - 1$.

- $(0000\ 0101)_2$ représente +5 ;
- $(1111\ 1011)_2$ représente -5 ;
- $(0000\ 0000)_2$ représente 0 ;

Une seule représentation du zéro.

Respecte l'arithmétique binaire.

Notation excédentaire

Dans cette représentation, on code en binaire naturel le nombre auquel on ajoute un décalage (offset).

L'offset vaut en général 2^{n-1} ainsi quel que soit le nombre x à coder, $x + \text{offset} \geq 0$.

- bit de poids fort à 1 : nombre positif
- bit de poids fort à 0 : nombre négatif

Plan

1 Définitions

2 Représentation des nombres

- Codage des entiers naturels
- Représentation des entiers négatifs
- **Représentation des réels**

3 Codage des informations alphanumériques

Représentation des réels

Plusieurs représentations possibles :

Virgule fixe : Principe identique au binaire naturel mais des puissances négatives de 2 à droite de la virgule.

Couple numérateur,dénominateur : complique l'arithmétique

Virgule flottante : $X = M \times b^E$, où M est la mantisse, b est la base et E est l'exposant.

Notation en virgule flottante

Il existe une infinité de couples (M, E) tels que

$$X = M \times 2^E$$

Par exemple,

$$\begin{aligned} 12,5 &= 6,25 \times 2^1 \\ &= 3,125 \times 2^2 \\ &= 1,5625 \times 2^3 \\ &= 25 \times 2^{-1} \\ &= 50 \times 2^{-2} \end{aligned}$$

Parmi toutes ces représentations, une seule, dite normalisée, est telle que $|M| \in [1, 2[$.

Une notation particulière est utilisée pour coder 0.

Norme IEEE754

- Les réels sont codés sur 32 bits (simple précision) ou 64 bits (double précision).
- Le couple (M, E) est tel que $|M| \in [1, 2[$.
- M a seul chiffre avant la virgule. C'est toujours 1. Il n'est pas inclus dans la représentation. $M = 1 + m$
- $e = E + \text{offset}$
- Représentation sur 32 bits $(-1)^s \times |(1 + m)| \times 2^{e-127}$

bit de signe	exposant	mantisso
s	e	m
1 bit	8 bits	23 bits

Norme IEEE754

Exemple :

1011 1111 0010 0000 0000 0000 0000 0000

- $s = 1 \Rightarrow (-1)^s = -1 \Rightarrow$ le nombre est négatif
- $e = (01111110)_2 = (126)_{10} \Rightarrow E = e - 127 = -1$
-

$m = 010\ 0000\ 0000\ 0000\ 0000\ 0000$

$$|M| = |1 + m| = (1,01)_2 = (1,25)_{10}$$

$$(-1)^1 \times 1,25 \times 2^{-1} = -0,625$$

Codage des informations alphanumériques

Pour coder des informations alphanumériques, on fait correspondre un mot binaire à chaque caractère.

Il existe plusieurs tables de correspondance.

Les plus utilisées ont été standardisées :

ASCII, ASCII étendu et EBCDIC : sur 7 ou 8 bits, caractères latin.

UNICODE : sur 16 bits, permet de représenter tous les caractères (arabe, latin, cyrillique, grec, hébreu, latin,...)

ASCII

valeur décimale	caractère
48	0
49	1
⋮	⋮
57	9
⋮	⋮
65	A
66	B
⋮	⋮
90	Z
⋮	⋮
97	a
98	b
⋮	⋮