Chapitre 1 Représentation des nombres – Numération

I Système de numération

I.1 - Introduction

Un entier naturel N dans une base b, peut se décomposer tel que :

$$N_b = \alpha_p \alpha_{p-1} \ldots \alpha_1 \alpha_0$$

avec les symboles $\alpha_i \rightarrow 0 < \alpha_i < b - 1$ (i position du symbole -1 car le premier i est 0)

Attention !

$N_b = \alpha_p \alpha_{p-1} \ldots \alpha_1 \alpha_0$ ne signifie pas que N peut s'écrire sous la forme d'une multiplication, c'est bien la suite de symboles qu'on représete.

Par exemple N = 12 avec $\alpha_0 = 1$ et $\alpha_1 = 2$

Donc α_i dépend de la base utilisée, par exemple en base 10 : $0 < \alpha_i < 9$

Pour une base b donnée, on peut la représenter dans une base 10 à chaque fois :

$$N_{(b)} \sum_{i=0}^{p} \alpha_i b^i$$

avec b^i le poids associé au symbole α_i

La signification du symbole dépend de son poids (donc de sa place dans la suite de symboles représentant le nombre) → **position du symbole très importante**

Example

15 $\neq$ 51
même symbole mais avec des positions différentes

Même chose pour la partie fractionnaire avec $b^i \rightarrow i < 0$ donc $b^i < 1$

I.2 - nombres décimaux - base 10

Les symboles en base 10 : $0 < \alpha_i < 9$
Les poids sont des puissances de 10 positives : 10^i

Example

Décomposer 1987 en base 10

$1987 = 7 \times 10^0 + 9 \times 10^1 + 9 \times 10^2 + 1 \times 10^3$
7 est dit le **poids faible** (nombre tout à droite)

Danger

Attention : décalage d'indice entre la position et le poids (dans l'exemple précédent, 9 est à la troisième position mais avec un poids de 2)

Partie fractionnaire (après la virgule)

Donc $i < 0$ et $n_{10} = \sum_{p}^{i-1} \alpha_i 10^i$

Même chose que précédemment mais les puissances de 10 sont négatives (car $i < 0$)

Example

Décomposer 25,308 en base 10

$25,308 = 8 \times 10^{-3} + 0 \times 10^{-2} + 3 \times 10^{-1} + 5 \times 10^{0} + 2 \times 10^{1}$

I.3 - nombres binaires (base 2)

Base la + utlisée en informatique : a deux valeur (0 ou 1), que l'on peut aussi noter (Vrai/Faux)

Les symboles utilisés dans la base binaires = **bits** (binary digit)
8 bits = 1 octect => stockage des informations en octetcs

Donc, l'écriture d'un nombre en base 2 est tel que :

$$N_{(2)} \sum_{i=0}^{p} \alpha_i 2^i = \alpha_0 2^0 + \ldots + \alpha_p 2^p$$

qui en écriture donne : $\alpha_p \ldots \alpha_0$

Danger

Entre la représentation d'un nombre par sa somme de symboles et de poids et son écriture, on inverse le sens !

La **première position** est celle toute à droite eb écriture

- **Most Significant Bit (MSB)** :α_p
- **Less Significant Bit (LSB)** : α_0

Example

Décomposer 101 ("un zéro un" et non "cent-un")

$101_{(2)} = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2$

On peut faire de même qu'en base 10 pour la partie fractionnaire.

Tableau des puissances de 2

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1
1024	512	**256** (octet)	128	64	32	16	8	4	3

2^0	2^{-1}	2^{-2}	2^{-3}
1	0.5	0.25	0.125

I.4 - nombres hexadécimaux (base 16)

Très utilisée en informatique pour l'affichage des nombres binaires car **1 hexadécimal correspond à 4 bits**

$$N_{(2)}\sum_{i=0}^{p}\alpha_i 16^i = \alpha_0 16^0 + \ldots + \alpha_p 16^p$$

avec $0 < \alpha_i < 15$$

- **Symboles :** {0, 1, ..., 9, **A, B, C, D, E, F**}
- **Poids** : puissances positives de 16

Tableau de correspondance

A	B	C	D	E	F
10	11	12	13	14	15

Example

Décomposer AF,D8$_{(16)}$ en base 16

$AF, D8_{(16)} = 8 \times 16^{-2} + D \times 16^{-1} + F \times 16^0 + A \times 16^1$

II Conversions (transcodage)

On cherche les opérations pour changer de base. Plus la base est petite, plus le nombre de symboles équivalent en sortie de conversion est grand.

II.1 - Conversion d'un nombre en base b (2 ou 16) à un nombre en base 10

Conversion la plus simple, toujours la même méthode utilisée --> **écriture polynomiale** :

$$N_b = \sum_{i=0}^{p}\alpha_i b^i = N_{10}$$

Example

Convertir 10011$_{(2)}$ en base 10

$10011_{(2)} = 1 \times 2^0 + 2 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4$

$10011_{(2)} = 1 + 2 + 0 + 0 + 16 = 19_{(10)}$

De même avec la base 16 :

Example

Convertir 5EA$_{(16)}$ en base 10

$5EA_{16} = A \times 16^0 + E \times 16^1 + 5 \times 16^2 = 10 + 14 \times 16 + 5 \times 256 = 1514_{(10)}$

A vaut 10, E vaut 14 en base 16

II.2 - Conversion d'un nombre en base 10 à un nombre en base b (2 ou 16)

Méthode un peu plus complexe que dans l'autre sens. Il existe 2 méthodes :

- par soustraction
- par division

Méthode par soustraction

1. Recherche du poids le plus proche inférieur de N dans la base b : b^{k-1}
2. Soustraction de $N_{(10)}$ du nombre qui multiplie cette puissance
3. Retour à l'étape 2 avec le nombre obtenu
4. Fin quand on arrive à b^0 pour la partie entière (donc $k = 1$)

> **Partie fractionnaire**
>
> Même méthode, poids fort à droite de la virgule (contrairement au nombre entier où le poids fort est à gauche de la virgule)
> Mais **attention**, le procédé peut devenir infini (cf TD_COD)

Nombre de bits nécessaires :

$$k = \lfloor log_2(N_{(10)}) \rfloor + 1$$
$$\text{ou}$$
$$2^{k-1} < N_{(10)} < 2^k$$

> **Example**
>
> ***Convertir 95$_{(10)}$ en base 2***
>
> 1. Puissance de 2 la plus proche de 95 : 64
> 2. On soustrait à 95 une fois 64 (car on a 1 fois 64 max dans 95) : $95 - 64 = 41$
> 3. On répète l'opération : puissance de 2 la plus proche --> 32 donc $31 - 0 \times 32 = 31$
>
> On a donc :
>
> - $95 - 1 \times 64 = 41$
> - $31 - 0 \times 32 = 31$
> - $31 - 1 \times 16 = 15$
> - $15 - 1 \times 8 = 7$
> - $7 - 1 \times 4 = 3$
> - $3 - 1 \times 2 = 1$
> - $1 - 1 = 0$
>
> Donc on a $95_{(10)} = 1011111_{(2)}$
>
> ***Convertir 95$_{(10)}$ en base 16***
>
> - $95 - 5 \times 16 = 15$
> - $15 - 15 \times 16^0 = 0$
>
> Donc $95_{(10)} = 5F_{(16)}$

Méthode par divisions euclidiennes successives par b

Partie entière

Danger

Lecture de bas en haut pour cette méthode contrairement à la méthode précédente qui se fait de haut en bas

Example

Convertir 95$_{(10)}$ en base 2

$$\begin{array}{ll} 95 & |\underline{2} \\ 94 & 47 \\ \ 1 & \end{array}$$

Puis, on prend le quotient de la division \etaon répète l'opération :

$$\begin{array}{ll} 47 & |\underline{2} \\ 46 & 23 \\ \ 1 & \end{array}$$

$$\begin{array}{ll} 23 & |\underline{2} \\ 22 & 11 \\ \ 1 & \end{array}$$

$$\begin{array}{ll} 11 & |\underline{2} \\ 10 & 5 \\ \ 1 & \end{array}$$

$$\begin{array}{ll} 5 & |\underline{2} \\ 4 & 2 \\ 1 & \end{array}$$

$$\begin{array}{ll} 2 & |\underline{2} \\ 2 & 1 \\ 0 & \end{array}$$

$$\begin{array}{ll} 1 & |\underline{2} \\ 0 & 0 \\ 1 & \end{array}$$

Le quotient est égal à 0 donc on s'arrêt à cette étape

Donc on obtient le nombre (**lecture de bas en haut des restes**) : $1011111_{(2)}$

Convertir 95$_{(10)}$ en base 2

$$\begin{array}{ll} 95 & |\underline{16} \\ 80 & 5 \\ 15 & \end{array}$$

$$\begin{array}{ll} 5 & |\underline{16} \\ 0 & 0 \\ 5 & \end{array}$$

Donc les restes sont 5 et 15 donc le nombre est $5F_{(16)}$

Info

- Les multiples de b se terminent par 0 :
  $2 \times 7 = 14_{(10)} = 1110_{(2)}$
- Les multiples de b^2 se terminent par 00 :
  $2^2 \times 3 = 12_{(10)} = 1100_{(2)}$
- Les multiples de b^3 se terminent par 000 :
  $2^3 \times 4 = 24_{(10)} = 11000_{(2)}$

(fonctionne en base 10 : 100 = 10^2)

Pour la conversion en base binaire :
Avec n bits → 2^n (valeurs de 0 à 2^{n-1})

Nombre de bits	4	8	16	32
Plage de valeurs	0 à 15	0 à 255	0 à 65 535	0 à 4 294 672 296

Partie fractionnaire

On peut utiliser la Méthode par soustraction pour la partie fractionnaire

La méthode de division successive ne fonctionne pas donc on utilise la méthode de **multiplication successives par b**
=> Le calcul peut être infini pour la partie fractionnaire

Example

Convertir 0,375$_{(10)}$ en base 2

$$0,375 \times 2 = 0,75$$
$$0,75 \times 2 = 1,5$$
$$0,5 \times 2 = 1,0$$

On s'arrête lorsque l'on atteint x,**0**

On lit les chiffres entiers pour la partie décimale : donc $0,375_{(10)} = 0,011_{(2)}$

Convertir 0,375$_{(10)}$ en base 16

$$0,375 \times 16 = 6,0$$

Donc : $0,375_{(10)} = 0,6_{(16)}$

II.3 - Conversion d'un nombre entre les bases 2 et 16

Base binaire = 2^2

Base hexadécimale = 2^4
Donc la conversion est assez rapide car on peut faire des paquets de bits

Info

Hexadécimal vers binaire
Chaque symbole est remplacé par 4 bits

Binaire vers hexadécimal

- 1 paquet de 4 bits = 1 symbole
- Séparation de la partie entière et de la partie fractionnaire
  - **Partie entière** : de la droite vers la gauche

- **Partie fractionnaire** : de la gauche vers la droite
- **Rajouter des 0 si besoin**

Example

Convertir FA,8$_{(10)}$ en base 2

- F en binaire : 1111$_{(2)}$ (on peut passer par l'intermédiaire de la base 10, où F vaut 15 si besoin)
- A en binaire : 1010$_{(2)}$
- 8 en binaire : 1000 (car $8 = 2^3$)

Donc : $F1,8_{(16)} = 1111\ 1010,1000_{(2)}$

Convertir 10111,01$_{(2)}$ en base 16

On ajoute des 0 à gauche de la virgule et à droite de la virgule pour avoir des paquets de 4 bits
0001 0111, 0100

Donc :

- 0001$_{(2)}$ = 1$_{(16)}$
- 0111$_{(2)}$ = 7$_{(16)}$
- 0100$_{(2)}$ = 4$_{(16)}$

Soit : 0001 0111,0100$_{(2)}$ = 17,4$_{(16)}$

III Opérations

Additions

Comme en décimal (indiquer les retenues)

Règles en base 2

- 1$_{(2)}$ + 1$_{(2)}$ = 10$_{(2)}$
- 1$_{(2)}$ + 0$_{(2)}$ = 1$_{(2)}$
- 1$_{(2)}$ + 1$_{(2)}$ + 1$_{(2)}$ = 11$_{(2)}$

Example

111$_{(2)}$ + 011$_{(2)}$

$$\begin{array}{ccccc} & 1 & 1 & 1 & \\ & & 1 & 1 & 1 \\ + & & 0 & 1 & 1 \\ & & - & - & - \\ & 1 & 0 & 1 & 0 \end{array}$$

Astuce pour la base 16

On complète pour arriver à 10$_{(16)}$

Exemples

- $A_{(16)} + 8_{(16)}$
  $= A_{(16)} + 6_{(16)} + 2_{(16)}$ car A+6 = 16

Donc $A_{(16)} + 8_{(16)} = 10_{(16)} + 2_{(16)} = 12_{(16)}$

- $C_{(16)} + 8_{(16)} = C_{(16)} + 4_{(16)} + 4_{(16)} = 14_{(16)}$

Example

$68_{(16)} + 3A_{(16)}$

$$\begin{array}{ccc} & ① & \\ & 6 & 8 \\ + & 3 & A \\ & - & - \\ & A & 2 \end{array}$$

Soustractions

Info

$1_{(2)} - 1_{(2)} = 0$
$10_{(2)} - 1_{(2)} = 1$

Example

$111_{(2)} - 011_{(2)}$

$$\begin{array}{cccc} & 1 & 1 & 1 \\ - & 0 & 1 & 1 \\ & - & - & - \\ & 1 & 0 & 0 \end{array}$$

$100_{(2)} - 011_{(2)}$

$$\begin{array}{cccc} & 1 & {}_{①}0 & {}_{①}0 \\ - & 0 & {}_{+①}1 & 1 \\ & - & - & - \\ & 0 & 0 & 1 \end{array}$$

Car 0 - (1 + 1) = 0 - 10 --> on doit rajouter une unité

Exemple en base 16

$68_{(16)} - 3A_{(16)}$

$$\begin{array}{cccc} & 6 & {}^{①}8 \\ - & {}_{+①}3 & A \\ & \text{—} & \text{—} \\ & 2 & E \end{array}$$

Tip

Pour faire 18 - A, de la même manière que pour l'addition on complète à $10_{(16)}$. Comme A = 8+2, on a :

$18_{(16)} - A_{(16)} = 18_{(16)} - 8_{(16)} - 2_{(16)} = 10_{(16)} - 2_{(16)} = E_{(16)}$

Autre exemple :

$15_{(16)} - 8_{(16)}$

Pour aller à $10_{(16)}$, on a : $15_{(16)} - 5_{(16)}$

Donc en décomposant $8_{(16)}$:

$15_{(16)} - 5_{(16)} - 3_{(16)} = 10_{(16)} - 3_{(16)} = D_{(16)}$

(car on a en base 16 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12 ... 1A, 1B ... etc.)