Codage des nombres

Laurent Signac – – 05-10-19 1231 4d3981000890095d251b

1 Généralités

L'information est la matière première de l'informatique.

Un **programme**, constitué d'**informations**, stocke, manipule, transforme, d'autres informations, à l'aide de **machines**

Toutes ces informations sont à un moment représentées par des nombres (séquence de 0 et 1).

Le binaire est la base la plus simple utilisable et elle est adaptée à l'électronique.

L'unité d'information est le *bit* : 0 ou 1. Un groupe de 8 bits constitue un octet (*byte* en anglais).

2 Valeur d'un nombre écrit en base b

En base b, on utilise exactement b symboles (les chiffres).

L'écriture en base b : $c_n c_{n-1} ... c_1 c_0$ représente un nombre qui a pour valeur :

$$v(c_n) \times b^n + v(c_{n-1}) \times b^{n-1} + ... + v(c_1) \times b^1 + v(c_0) \times b^0$$

avec $v(c_k)$ la valeur du chiffre c_k.

3 Écriture d'un nombre dont on connaît la valeur

En effectuant des divisions successives de n par la base b, la séquence des restes donne la valeur de chaque chiffre de l'écriture de n en base b.

4 Nombres à virgule

4.1 Base 2 → base 10

$$\begin{aligned} 100,011_2 &= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + \\ &\quad 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 4,375 \end{aligned}$$

4.2 Base 10 → base 2

$$\begin{array}{rcl} 0,6875 & \times 2 = & 1,375 \\ 0,375 & \times 2 = & 0,75 \\ 0,75 & \times 2 = & 1,5 \\ 0,5 & \times 2 = & 1 \end{array}$$

$$0,6875_{10} = 0,1011_2$$

5 Développements binaires infinis

En base b, un nombre $\frac{p}{q}$ (fraction irréductible) a une écriture finie si tous les facteurs premiers de q sont aussi des facteurs premiers de b. En binaire, seuls les nombres de la forme $\frac{p}{2^k}$ ont une écriture finie.

Certains nombres ont une écriture infinie en binaire. Par exemple :

$$0,2_{10} = 0,001100110011..._2 = 0,\overline{0011}_2$$

6 Nombres négatifs

Une fois choisi le nombre de bits du codage (n), on représente les entiers compris entre 0 et $2^{n-1}-1$ comme d'habitude, et on choisit comme code d'un entier $-k$ compris entre -2^{n-1} et -1 : l'écriture binaire de $2^n - k$

Le complément à 2 est involutif.

7 Codage des nombres à virgule

7.1 Codage en virgule fixe

- Retenir un nombre fixe de chiffres.
- Garder simplement un nombre fixe de bits avant et après la virgule.
- L'espace entre 2 nombres qui se succède est toujours le même.
- **Problème** : l'erreur relative peut être grande

7.2 Codage en virgule flottante

- Norme IEEE754
- Idée : Retenir un nombre fixe de **chiffres significatifs**
- Les petits nombres seront plus *serrés* que les grands
- L'erreur relative est **tolérable**
- Simple précision (32 bits) et double précision (64 bits)

7.2.1 Écriture du nombre en notation scientifique en base 2 :

$$25,90625 \Rightarrow 11001,111101 \Rightarrow 1.1001111101 \times 2^4 \times (-1)^0$$

7.2.2 Écriture en simple précision

- 1 bit de signe
- 8 bits d'exposant décalé E ($E = e + 127$)
- 23 bits contenant les chiffres significatifs

$$\underbrace{0}_{s} \quad \overbrace{10000011}^{E\text{:8 bits } (=e+127)} \quad \underbrace{100111110100...0}_{23 \text{ bits}}$$

Types de nombres :

- Normalisés : $0 < E < 255 \Rightarrow (-1)^s \times 1,M \times 2^{E-127}$
- Dénormalisés : $E = 0$ et $M \neq 0 \Rightarrow 0,M \times 2^{-126} \times (-1)^s$
- Zéro : $E = 0$ et $M = 0$
- Infini : $E = 255$ et $M = 0 \Rightarrow (-1)^s\infty$
- Indéfini : $E = 255$ et $M \neq 0$

Flottants $\neq$ réels

Il y a des nombres incodables, donc des erreurs d'arondi.

En simple précision :

- 2^{127} se code $011111110\,\overbrace{0...0}^{22}\,0$
- $2^{127} + 2^{104}$ se code $011111110\,\underbrace{0...0}_{22}\,1$