

Représentation des données : types et valeurs de base

Représentation approximative des réels

J. Boucher

Lycée Pierre-Paul RIQUET, Première NSI

21 novembre 2024

Plan

- 1 II. Représentation des entiers relatifs
 - 1. Représentation des nombres fractionnaires
 - 2. Encodage en virgule fixe
 - 3. Encodage en virgule flottante

1. Représentation des nombres fractionnaires

Décomposition d'un nombre fractionnaire

La **partie fractionnaire** de tout nombre peut se décomposer, de manière unique, dans une base $b \geq 2$, en une **somme de puissances inverses** de b .

Spé. Maths : Pour tout nombre d tel que $0 \leq d < 1$, on a :

$$d = a_1 \times b^{-1} + a_2 \times b^{-2} + \dots + a_n \times b^{-n}$$

avec $n \in \mathbb{N}^*$ et $0 \leq a_i < b$.

1. Représentation des nombres fractionnaires

Exemple : Décomposition de $\frac{1}{32}$

- en base 10 :

$$\begin{aligned}\frac{1}{32} &= 0 \times 10^{-1} + 3 \times 10^{-2} + 1 \times 10^{-3} + 2 \times 10^{-4} + 5 \times 10^{-5} \\ &= 0,03125\end{aligned}$$

- en base 2 :

$$\begin{aligned}\frac{1}{32} &= 0 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} \\ &= (0,00001)_2\end{aligned}$$

- en base 16 :

$$\begin{aligned}\frac{1}{32} &= 0 \times 16^{-1} + 8 \times 16^{-2} \\ &= (0,08)_{16}\end{aligned}$$

1. Représentation des nombres fractionnaires

Certain nombre ne peuvent pas être représenté de manière exacte dans une base donnée.

■ Représentaion en base 10 :

■ exacte : $\frac{1}{10} = 0,1$

■ approximative : $\frac{1}{3} = 0,3333\dots$

■ Représentaion en base 2 :

■ exacte : $\frac{1}{4} = (0,01)_2$

■ approximative : $\frac{1}{10} = (0,0001100110011001100\dots)_2$

2. Encodage en virgule fixe

Principe

Un codage en **virgule fixe** consiste à représenter la partie entière et la partie fractionnaire sur un nombre fixé de bits.

Encodage en virgule fixe sur 16 *bit*

$$51,1796875 = \underbrace{00110011}_{8 \text{ bits}}, \underbrace{00101110}_{8 \text{ bits}}$$

Sur 16 *bits* on peut représenter tous les nombres tels que :

- la **partie entière** soit comprise entre -2^7 et $2^7 - 1$ (en complément à deux) ;
- la **partie fractionnaire** appartienne à l'intervalle $[0; 1 - 2^{-8}]$ en commettant une erreur d'approximation d'au plus 2^{-8} .

→ **Cette représentation n'est pas adaptée aux nombres très petits ainsi qu'aux nombres très grands.**

3. Encodage en virgule flottante

Principe

La représentation en virgule flottante est similaire à l'écriture scientifique d'un nombre décimal. Elle consiste à décomposer un nombre en trois parties telles que :

$$\textit{signe} \times (1 + \textit{mantisse}) \times 2^{\textit{exposant}}$$

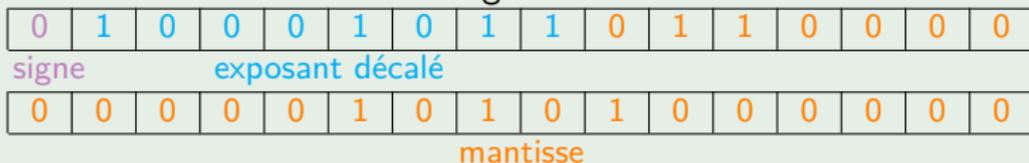
- *signe* : encodé par 0 pour les positifs, 1 pour les négatifs ;
- *mantisse* : partie fractionnaire du nombre ($0 \leq \textit{mantisse} < 1$) ;
- *exposant* : entier relatif (souvent décalé pour pouvoir être stocké sous la forme d'un entier positif).

3. Encodage en virgule flottante

Encodage en virgule flottante sur 32 bits

$$\begin{aligned}
 5632,65625 &= (1011000000000,10101)_2 \\
 &= (1,01100000000010101)_2 \times (10)_2^{(1100)_2} \\
 &= +(1 + (0,01100000000010101)_2) \times (10)_2^{(1100)_2}
 \end{aligned}$$

Sur 32 bits obtient donc l'encodage suivant :



- L'exposant décalé code sur 8 bits un entier entre 0 et 255. Il est décalé de $2^7 - 1 = 127$ pour représenter des valeurs d'exposant comprise entre -127 et 128 . Ici on a bien $135 - 127 = 12 = (1100)_2$
- La mantisse code sur 23 bits la partie fractionnaire, en excluant le « zéro » de la partie entière.