Travaux Dirigés d'algorithmique n°2

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

La représentation des nombres en machine

Alors que pour les entiers, la représentation en machine ne présente pas d'énormes difficultés puisqu'elle correspond globalement à la décomposition en base 2 de l'entier en question, il faut un peu plus de "travail" pour faire "rentrer un nombre réel en machine". Le but de cette séance est donc de présenter le standard IEEE qui permet un tel codage.

Bibliographie : "Concepts fondamentaux de l'informatique", *A. AHO et J. Ullman,* Dunod, pages 219–229.

---

1 La représentation des entiers

Dans un premier temps, rappelons le codage des entiers : la manière la plus simple et la plus immédiate est celle qui représente un **entier non signé** : une chaîne binaire de 32 bits est traitée comme étant un nombre binaire.

Exemple : Dans cet exemple (et les suivants), nous utiliserons des mots de six bits (au lieu de 32) pour simplifier les notations. La chaîne $abcdef$ représente l'entier

$$32\,a + 16\,b + 8\,c + 4\,d + 2\,e + f\ .$$

Concrètement, $(101100)_2 = (44)_{10}$.

On manipule généralement des nombres positifs **ou** négatifs, ceci nous oblige à représenter le signe d'une manière ou d'une autre.

1.1 Notation en grandeur signée

La manière la plus simple est d'utiliser le premier bit comme **bit de signe**, 1 signifie que le nombre est négatif et 0 signifie que le nombre est positif. (Les bits restants représentent la valeur absolue de l'entier).

Exemple : 101100 représente $(-12)_{10}$ et 001100 représente $(12)_{10}$.

Cette notation pose plusieurs problèmes :

- premièrement zéro possède deux représentations ($00\cdots0$ et $10\cdots0$),
- et les opérations d'addition et de soustraction sont non triviales.

1.2 Complément à deux

- Pour interpréter une chaîne de n bits comme un nombre en complément à deux, nous l'interprétons d'abord comme un entier non signé. Si le bit de tête de la chaîne est 1, nous soustrayons 2^n.
- Pour coder un entier N en complément à deux : si N est positif ou nul, nous le convertissons en un nombre binaire, mais si N est négatif, nous calculons $N + 2^n$, puis convertissons cette valeur en nombre binaire.

1.3 Code k-excès

Pour interpréter une chaîne de n bits en code k-excès, il faut traiter la chaîne comme un entier non signé et soustraire k. De façon équivalente, pour représenter N en code k-excès, il faut convertir $N + k$ en un entier non signé. De cette manière, il est évident que nous ne pouvons représenter uniquement les nombres allant de $-k$ à $2^n - k - 1$ (bornes incluses). Cette méthode est utilisée par le standard IEEE.

▸ **Exercice 1.** *Représenter les entiers 13, 50, 59, -3, -9, -20 dans les représentations suivantes : notation grandeur signée, complément à deux, code* 8*-excès.*

▸ **Exercice 2.** *Qu'imprime le programme suivant ?*

```
#include<stdio.h>

void main(void)
{
  int x=3, y=2, z=1;
  printf("%d\n", x | y & z );
  printf("%d\n", x | y & ~ z );
  printf("%d\n", x ^ y & ~ z );
  printf("%d\n", x & y && z );
}
```

▸ **Exercice 3.** *Que pouvez-vous dire du bit le plus à droite de* `x & (x-1)` *?*

2 La représentation des réels

Soit x un nombre réel. Le standard IEEE propose de l'écrire de la façon suivante :

$$x = (-1)^S \times 2^{E-127} \times 1.M$$

- où **S** est le bit de signe ;
- **E** est l'exposant (codé sur 8 bits) ;
- **M** est la mantisse (codé sur 23 bits) .

S	E	M
0	1 2 3 4 5 6 7 8	9 10 11 12 13 $\cdots$ 30 31

▸ **Exercice 4.** *Donner la représentation des nombres* ***-14.4*** *et* ***1.25*** *sur 32 bits dans le standard IEEE.*

▸ **Exercice 5.** *Donner le réel dont la representation IEEE (codée en hexadécimale) est la suivante :*

`C 1 4 C 0 0 0 0`