2 – Représentation des entiers relatifs

Le signe d'un nombre peut prendre 2 valeus (+ ou -), il suffit donc d'un bit pour le représenter. Mais les ordinateurs représentent les entiers négatifs à l'aide d'un codage plus approprié qui facilite les opérations arithmétiques : le complément à 2^n

1 – Représentation des nombres entiers – convention

1 – Le binaire non signé

Les entiers naturels sont codés sur machine en base 2, sur un nombre arbitraire de bits. Pour simplifier nos illustrations, nous allons considérer des entiers codés sur 8 bits

On peut travailler sur 8 bits en Python, en utilisant la fonction int8 de numpy

Dans la représentation binaire non signé, les nombres entiers naturels sont écrit en base 2

Ex : Le nombre 10 en base 10, s'écrit 1010 en base 2.
Dans la mémoire, sur 8 bits, il est codé 00001010.

Pour une mémoire sur 8 bits, tous les entiers naturels de 0 à 255 peuvent donc être représenter de cette façon.

2 – Le binaire signé

Dans la représentationen binaire signé, le bit de poid fort (le plus à gauche), sert à représenter le signe :

- 0 pour un entier positif
- 1 pour un entier négatif

Il y a donc moyen de représenter

•128 codes avec le bit de signe à 1
ce sont 128 nombres négatifs
(de –1 à – 128)

•128 codes avec le bit de signe à 0
le nombre 0 et 127 nombres positifs
(de 1 à +127)

Utiliser les n autres bits pour représenter la valeur absolue du nombre en binaire non signé pose problème.
Ex : sur 4 bits, si l'on utilise l'algorithme d'addition habituel, 3 + -2 serait égal à -5 :

```
  0 0 1 1 → 3
+ 1 0 1 0 → -2
  1 1 0 1 → -5
```

On code les nombres signés avec le complément à 2^n

Exo : Parmi les entiers suivant, écrit en utilisant le complément à 2^8, lequel ou lesquels sont négatifs ?

a. 0110 1001 b. 1010 1001 c. 10 1001

b car sur 8 bits, on rajoute deux 0 devant pour compléter, comme ça commence par 0, il est positif.

2 – Le complément à 2^n

Ce codage permet de représenter les entiers relatifs tout en ne changeant pas d'algorithme d'addition.

1 – Définitions

Prendre le complément à 1 d'un nombre, c'est inverser les 0 et les 1 de son écriture binaire. On notera b barre le complément à 1 du bit b.

Ex : le complément à 1 de 1001 sera 0110
1 barre = 0 et 0 barre = 1

L'opposé d'un nombre x est le nombre y vérifiant x + y = 0
Ex : l'opposé de 3 sera -3 car 3 + -3 = 0

Le complément à 2^n d'un nombre, c'est le nombre qu'il faut lui ajouter pour obtenir 2^n

2 – Opposé d'un nombre en 8 bits

On peut remarquer qu'un nombre d'un octet et son complément à 1, va s'écrire 1111 1111
Ex : sur 8 bits 1001 1010 + 0110 0101 = 1111 1111

On note b barre le complément à 1du bit b
Alors pour un nombre sur 8 bits :

b7b6b5b4b3b2b1b0 + (b7b6b5b4b3b2b1b0)barre + 1 = 1111 1111 + 1 = 1 0000 0000

Par convention, l'opposé d'un nombre positif est son complément à 2^taille des entiers, c'est à dire son complément à 1, plus 1.

Ex : on cherche l'opposé du nombre positif 3
3 = 0000 0011 sur 8 bits, son complément à 1 est donc 1111 1100
l'opposé (-3) s'écrit alors 1111 1100 + 1 = 1111 1101

Pour connaître le nombre que représente un entier négatif, on effectue la démarche inverse : on lui retranche 1 puis on prend son complément à 1.

Ex : Quel entier relatif correspond à 1011 0101, il est négatif car son premier bit vaut 1.

On lui retranche 1 : 1011 0101 – 1 = 1011 0100

On prend en suite son complément à 1 : 0100 1011

On le convertit en base 10 : On trouve 75

1011 0101 est donc l'écriture sur 8 bits de -75

Pour continuer

http://villemin.gerard.free.fr/Wwwgvmm/Numerati/BINAIRE/Negatif.htm
https://www.courstechinfo.be/MathInfo/NbrSignes.html