IEEE-754 : Aide Mémoire

1 Représentation des réels

On considère la norme IEEE 754 pour représenter les réels :

Signe (S)	Exposant (E)	Pseudomantisse (P)
1 bit	e bits	p bits

Le nombre ainsi représenté a pour valeur :
$(-1)^S * 2^{E-(2^{e-1}-1)} * (1 + \frac{P}{2^p})$

Valeurs particulières

Les valeurs de l'exposant avec tous les bits à 0 ou 1 ($E = 00000000$ et $E = 11111111$ en 32-bits) sont réservées pour représenter des valeurs particulières, à savoir :

Exposant	Pseudomantisse	Valeur
Tous les bits à 0	Tous les bits à 0	Représentation de la valeur zéro.
Tous les bits à 0	Au moins un bit non nul	Représentation d'une valeur dénormalisée.
Tous les bits à 1	Tous les bits à 0	Représentation d'une valeur infinie
Tous les bits à 1	Au moins un bit non nul	Représentation d'une valeur *NaN*

- Il y a deux représentations possibles pour le zéro, le zéro positif et le zéro négatif.
- Les nombres dénormalisés sont utilisés pour représenter certaines valeurs sortant de l'intervalle autorisé par la représentation utilisée.
- Un nombre dénormalisé a pour valeur :$(-1)^S * 2^{1-(2^{e-1}-1)} * (\frac{P}{2^p})$. Attention pas de 1 implicite devant la pseudo-mantisse.
- Une valeur infinie est utilisée pour indiquer par exemple un dépassement de capacité. Il y a deux valeurs infinies possibles, positive et négative.
- Une valeur *NaN* indique toute valeur erronnée, comme par exemple la racine carrée d'un nombre négatif.

Arrondis

Si un nombre ne peut pas exactement être représenté en binaire flottant, on utilise une valeur arrondie. Il existe plusieurs modes d'arrondi :

- L'arrondi au plus proche arrondit à la valeur la plus proche. En cas d'égalité, on choisit la valeur paire (se terminant par 0 en binaire). C'est le mode d'arrondi par défaut.
- L'arrondi vers 0 arrondit à la valeur la plus proche de 0 (troncature).
- L'arrondi vers $+\infty$ arrondit à la plus proche valeur la plus grande.
- L'arrondi vers $-\infty$ arrondit à la plus proche valeur la plus petite.

Formats courants

	Taille	Signe	Exposant	Pseudomantisse
Simple précision	32	1	8	23
Double précision	64	1	11	52

Dans la suite, pour simplifier les calculs, les réels seront désormais représentés sur 16 bits avec la représentation suivante (ce n'est pas la même que la représentation demi précision de la norme IEEE 754) :

Signe	Exposant	Mantisse
1 bit	4 bits	11 bits

2 Codage d'un flottant

Comment représenter 1.32 en flottant ? $1.32 = \frac{132}{100} = \frac{33}{25}$

```
 100001        | 11001
- 11001        |------        /
-------vv      | 1.01010001111\01...
   100000                    /
  - 11001
   ------vv
      11100
    - 11001
    -------vvvv
         110000
       -  11001
       --------v
          101110
        - 11001
        -------v
           101010
         - 11001
         -------v
            100010
            ....
```

On dispose de 11 bits pour la pseudo-mantisse, 01010001111 est l'arrondi au plus proche.
L'exposant ici est 2^0 et le signe 0, donc ce chiffre s'écrira en flottant :

$$0 \quad 0111 \quad 01010001111$$

3 Additions sur les flottants

L'addition de deux flottants se fait en plusieurs étapes. On prendra comme exemple l'addition des nombres suivants :

$$0 \quad 0111 \quad 10000000000 = +1.1 \times 2^{7-(2^{4-1}-1)} = +1.1_2 \times 2^0 = 1.5 \tag{1}$$

$$0 \quad 0110 \quad 00000000000 = +1.0 \times 2^{6-(2^{4-1}-1)} = +1.0_2 \times 2^{-1} = 0.5 \tag{2}$$

1. On ramène les deux nombres au même exposant, le plus grand des deux. On décale donc les bits de la mantisse du nombre ayant le plus petit exposant d'autant de bits vers la droite que la différence entre les exposants, sans oublier le 1 avant la virgule.

   Dans l'exemple on veut augmenter de 1 l'exposant du deuxième terme. La mantisse complète du second nombre passe donc de 1.00000000000 à 0.10000000000 (on supprime le zéro le plus à droite pour rester sur 11 bits).
2. On ajoute ensuite les deux mantisses, en tenant compte du signe (attention ici les mantisses ne sont pas exprimées en complément à 2) Dans l'exemple les mantisses sont de même signe on peut donc les ajouter directement : $1.10000000000 + 0.10000000000 = 10.00000000000$
3. On renormalise ensuite le nombre obtenu. Dans l'exemple, le résultat a donc pour exposant 0 et pour mantisse 10.000000000 : on renormalise la mantisse, ce qui augmente l'exposant de 1. On a donc le résultat final :

   $$0 \quad 1000 \quad 00000000000 = +1.0_2 \times 2^1 = 2$$

4 Multiplication sur les flottants

Supposons que l'on multiplie les nombres suivants :

$$0 \quad 1000 \quad 01000000000 = +1.01_2 \times 2^1 = 10.1_2 = 2.5 \tag{3}$$

$$0 \quad 1000 \quad 10000000000 = +1.1_2 \times 2^1 = 11_2 = 3 \tag{4}$$

La multiplication se fait en plusieurs étapes :

1. Détermination du bit de signe : il vaut 1 si les bits de signe des deux réels sont différents, et 0 s'ils sont identiques. Ici, les bits de signe sont identiques : le bit de signe du résultat sera donc 0
2. L'exposant est la somme des exposants respectifs des réels.
   Les exposants des deux réels sont tous les deux 1 : l'exposant du résultat sera donc $1 + 1 = 2$, soit 1001_2 en codage par excès. Cela peut aussi se faire en ajoutant les exposants tels qu'ils apparaissent dans la représentation binaire et en y soustrayant l'excédent.

   $$1000_2 + 1000_2 - 111_2 = 8 + 8 - 7 = 9 = 1001_2$$

3. On multiplie les mantisses entre elles. Ceci peut s'effectuer d'une façon similaire à une multiplication en base 10.
   La multiplication des mantisses 1.01000000000 et 1.1000000000 peut s'écrire :

   $$1.01 \times 1.1 = 1.01 \times (1.0 + 0.1) = 1.01 + 0.101 = 1.111$$

   soit 1.11100000000 pour la mantisse du résultat. On en déduit l'écriture du résultat : 0 1001 11100000000, ce qui correspond bien à l'écriture de 7.5