

Base d'informatique - Fiche de révision

Matéis R.

Octobre 2023

1 Conversion d'une base b vers la base décimal

Pour convertir des nombres codés en base 2, 8, 16 et 20. La méthode est la même, on multiplie chacun des chiffres par la base b élevé au carré d'indice où se trouve le chiffre en question. Voici le tableau des conversions pour aller plus vite :

n	10	9	8	7	6	5	4	3	2	1	0
2^n	1024	512	256	128	64	32	16	8	4	2	1
8^n							4096	512	64	8	1
10^n								1000	100	10	1
16^n								4096	256	8	1
20^n								8000	400	20	1

Ainsi on regarde dans le tableau les valeurs qui nous intéressent et on fait l'addition.

2 Addition en binaire

Voici les règles d'addition en binaire :

$$\begin{aligned}0_2 + 0_2 &= 0_2 \\0_2 + 1_2 &= 1_2 \\1_2 + 0_2 &= 1_2 \\1_2 + 1_2 &= 10_2\end{aligned}$$

Pour la dernière règle ($1_2 + 1_2 = 10_2$), on abaisse le 0 et on génère une retenue de 1.

3 Les entiers relatifs

On s'intéresse à présent à la représentation des entiers relatifs. Pour représenter les nombres négatifs, on utilise la notation dite binaire en complément à deux.

Nb bits	Valeur Minimum	Valeur Maximum	Nb Valeurs
8	-128	127	256
16	-32_768	32_767	65_536
32	-2_145_483_648	2_145_483_647	4_294_967_296

Si on veut convertir un nombre négatif, on doit :

1. On prend la valeur absolue du nombre.
2. On la code en binaire sur le nombre choisis de bits.
3. On utilise le complément à deux de la valeur codée en inversant les 0 et les 1.
4. On ajoute 1 au résultat.

Par exemple :

$$\begin{array}{r}
 -18_{10} \\
 18_{10} \\
 0001_0010_2 \\
 1110_1101_2 \\
 1110_1110_2
 \end{array}$$

Ainsi, -18 s'écrit 1110_1110₂ en binaire.

4 Coder un réel flottant

Pour coder un réel flottant au format IEEE 754 en simple précision (sur 32bits), on commence par s'intéresser au signe du réel, on dit alors que $S = 1$ si le réel est négatif et $S = 0$ sinon. Ensuite, on code la partie entière en valeur absolue, on appelle ça la mantisse. Et pour finir, on code la partie décimale avec des puissances de deux négatives (voici un tableau des puissances de deux négatives, ci-dessous). Plus simplement, pour coder en binaire la partie décimale, il faut multiplier la partie décimale par 2 jusqu'à obtenir 0, a chaque étape on garde le chiffre le plus à gauche du résultat (c'est à dire la partie entière) de la multiplication qui sera 1 ou 0, puis, on réitère la multiplication sur la partie décimale du résultat de la multiplication en supprimant le premier 1 s'il existe.

n	0	-1	-2	-3	-4
2^n	1	$\frac{1}{2} = 0,5$	$\frac{1}{4} = 0,25$	$\frac{1}{8} = 0,125$	$\frac{1}{16} = 0,0625$

Une fois que l'on a converti toutes les parties du nombre réel en binaire, dans la partie entière (codé en binaire) on décale la virgule vers la gauche jusqu'au premier 1 qui compose la mantisse. Une fois ça, on obtient la mantisse tronquée. Maintenant qu'on a déplacé la virgule, on compte le nombre de rang que l'on a déplacé. On ajoute a ce nombre 127 et le convertie en binaire, on obtient alors l'exposant décalé. Après ça, on place le bit du signe, puis l'exposant décalé, enfin

la mantisse tronqué et pour terminer on ajoute des 0 pour que l'on obtienne un 32bits. Une fois tout ça fait, on convertis le résultat en hexadécimal. Voici un exemple :

$$\begin{aligned}
 & -1027,625_{10} \\
 & S = 1 \\
 & 1027_{10} = 0100_0000_0011_2 \\
 & 0,625_{10} = 101_2 \\
 & M = 0100_0000_0011,101_2 \\
 & M_t = 0000_0000_1110_1_2 \\
 & E_d = 1000_1001_2 \\
 \Leftrightarrow & 1100_0100_1000_0000_0111_0100_0000_0000_2 \\
 \Leftrightarrow & C4_80_74_00_{16}
 \end{aligned}$$

5 Unicode et représentation UTF

L'UTF-8 est rétro-compatible en fonction des chaînes de caractères à coder. Voici un tableau avec le nombre d'octets nécessaires en fonction du point de code de chaque caractère :

1. Les 127 premiers caractères de l'ASCII 7 bits ont les mêmes valeurs en UTF-8 et sont donc codés sur un octet
2. Pour coder les caractères de valeurs comprises entre 128 et 2047 on utilise deux octets
3. Puis trois octets pour coder les caractères de valeurs comprises entre 2048 et 65535
4. Enfin, on utilise quatre octets pour les caractères de valeurs supérieures à 65535

6 L'algèbre de bool

Soit deux variables booléennes a et b . On note les opérations booléennes ainsi $a + b$, $a.b$ et \bar{a} , voici leurs table de vérité :

a	b	$a + b$	$a.b$	\bar{a}	\bar{b}
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	0	1
1	1	1	1	0	0

Ainsi on définit une fonction booléenne $f_1(a, b) = a_0b_0 + \dots + a_nb_n$.
Voici les règles de calcul pour les expressions algébriques de bool :

Loi		Forme +		Forme .
élément neutre	R1	$x + 0 = x$	R2	$x.1 = x$
d'idempotence	R3	$x + x = x$	R4	$x.x = x$
d'inversion	R5	$x + \bar{x} = 1$	R6	$x.\bar{x} = 0$
d'absorption	R7	$x + x.y = x$	R8	$x.(x + y) = x$
de De Morgan	R9	$\bar{x + y} = \bar{x} + \bar{y}$	R10	$\bar{x.y} = \bar{x} + \bar{y}$
de commutativité	R11	$x + y = y + x$	R12	$x.y = y.x$
d'associativité	R13	$x + (y + z) = (x + y) + z$	R14	$x.(y.z) = (x.y).z$
de distributivité	R15	$x.(y + z) = x.y + x.z$	R16	$x + y.z = (x + y).(x + z)$