SOLUTIONS *Travaux dirigés n^o 2*

1 Exercice n^o 1 : Codage des entiers

1. Combien de nombres entiers naturels peut-on représenter en binaire sur n bits ?
Solution : Chacun des n bits peut stocker un “ 0 ” ou un “ 1 ”, soit deux valeurs. On a donc 2^n possibilités d'écritures différentes. Il en résulte que l'on peut représenter, en binaire sur n bits, 2^n entiers naturels.
2. Quel est le plus petit entier relatif codable sur $n+1$ bits ?
Solution : Le bit de poids fort contient le signe de l'entier (0 pour le signe “ + ” et 1 pour le signe “ $-$ ”). Le codage de la valeur absolue s'effectue donc sur les n bits non utilisés. L'entier le plus petit est l'entier négatif $-x$ où x est la plus grande valeur absolue possible que l'on peut coder sur n bits, soit les n bits tous égaux à 1. Donc x vaut en base dix $\sum_{i=0}^{n-1} 2^i = 2^n - 1$. Le plus petit entier relatif que l'on peut ainsi coder est $-(2^n-1) = 1-2^n$.
Quel est le plus grand entier relatif codable sur $n+1$ bits ?
Solution : On effectue le même raisonnement qu'à la question précédente (il y a simplement le signe qui change). On obtient alors $2^n - 1$ comme étant ce plus grand entier.
Donner la représentation de zéro. Que remarquez-vous ?
Solution : Zéro peut être interprété soit comme -0, soit comme $+0$, il a donc deux représentations $\underbrace{0\cdots0}_{n+1\ fois}$ ou $1\underbrace{0\cdots0}_{n\ fois}$.
(Pour ces questions, on suppose que l'on utilise le bit de poids fort pour le signe d'un nombre entier relatif - 0 pour le signe “ + ” et 1 pour le signe “ $-$ ” - puis que les n autres bits permettent de coder la valeur absolue de l'entier.)
3. Supposons que l'on souhaite coder en binaire les fractions, *i.e.*, les écritures de la forme “ $\pm\frac{a}{b}$” où a et b sont deux entiers naturels et $b \neq 0$.
   (a) Imaginer un codage systématique en machine de ce type de nombres.
   Solution : On peut imaginer utiliser le bit de poids fort comme bit de signe, puis m bits pour le numérateur (en tant qu'entier naturel) et n bits pour le dénominateur (en tant qu'entier naturel, non nul). C'est donc un codage sur $m+n+1$ bits.
   (b) À partir de maintenant, on se donne $2n+1$ bits pour coder les fractions. Le bit de poids fort correspond au bit de signe (0 pour “ + ”, 1 pour “ $-$ ”). Les n bits suivants (de gauche à droite) correspondent à l'écriture binaire (sur n bits) du nombre naturel a et enfin les n derniers bits correspondent à la représentation binaire de b. Donner la représentation de $-\frac{22}{3}, \frac{5}{4}, -\frac{100}{25}, -\frac{4}{1}, \frac{16}{32}, \frac{1}{2}$, et, dans tous les cas, donner le nombre minimum n de bits permettant de représenter ces fractions.
   Solution :
   – $(22)_{10} = (10110)_2$, $(3)_{10} = (11)_2$, donc $-\frac{22}{3} = -\frac{(10110)}{(11)_2}$. Comme le numérateur est de longueur 5 bits, il nous faut $n = 5$, et donc 11 bits pour coder la fraction. On obtient $\underbrace{1}_{signe}\underbrace{10110}_{num}\underbrace{00011}_{denom}$.

- $(5)_{10} = (101)_2$, $(4)_{10} = (100)_2$, $n = 3$ donc il faut 7 bits, et $\frac{5}{4} = 0101100$.
- $(100)_{10} = (1100100)_2$, $(25)_{10} = (11001)_2$, $n = 7$ donc il faut 15 bits, et $-\frac{100}{25} = 111001000011001$.
- $(4)_{10} = (100)_2$, $(1)_{10} = (1)_2$, $n = 3$ donc il faut 7 bits, et $-\frac{4}{1} = 1100001$.
- $(1)_{10} = (1)_2$, $(2)_{10} = (10)_2$, $n = 2$, il faut 5 bits, et $\frac{1}{2} = 01001$.

Donner la (les) représentation(s) de zéro.
Solution : zéro est représenté par différentes notations sous forme de fraction, en effet il suffit d'avoir zéro au numérateur (et bien entendu un dénominateur non nul), le signe peut être positif ou négatif, peu importe. On a donc une multitude de différentes représentations.
Que pensez-vous d'un tel système ?
Solution : Ce système n'est pas efficace car on code un même nombre de plusieurs façons différentes.

2 Exercice n^o 2 : Représentation des nombres à virgule

1. Donner la représentation en base dix des nombres suivants.**Solutions :**
   (a) $(10011, 11101)_2 = 2^4 + 2 + 1 + 2^{-1} + 2^{-2} + 2^{-3} + 0 \times 2^{-4} + 2^{-5} = (19, 90625)_{10}$;
   (b) $(A23C, CC09)_{16} = 10 \times 16^3 + 2 \times 16^2 + 3 \times 16 + 12 + 12 \times 16^{-1} + 12 \times 16^{-2} + 0 \times 16^{-3} + 9 \times 16^{-4} = (41532, 7970123291015625)_{10}$ (On a : $16^{-1} = 0, 0625$, $16^{-2} = 0, 00390625$, $16^{-4} = 0, 0000152587890625$) ;
   (c) $(414, 23)_5 = 4 \times 5^2 + 5 + 4 + 2 \times 5^{-1} + 3 \times 5^{-2} = (109, 52)_{10}$;
   (d) $(8, 5)_{60} = 8 \times 60 + 5 \times 60^{-1} = (480, 08333)_{10}$;
   (e) $(102, 221)_3 = (11, 92592593)_{10}$.
2. Donner la représentation dans la base cible des nombres décimaux suivants. **Solution :**
   (a) $(89, 0625)_{10}$ vers la base deux ; $(89)_{10} = (1011001)_2$. Pour convertir la partie fractionnaire $(0, 0625)_{10}$, on fait comme suit : $0, 0625 \times 2 = 0, 125$ donc $a_{-1} = 0$, $0, 125 \times 2 = 0, 25$ donc $a_{-2} = 0$, $0, 25 \times 2 = 0, 5$ donc $a_{-3} = 0$ et $0, 5 \times 2 = 1, 0$ donc $a_{-4} = 1$ et on arrête. On a donc $(89, 0625)_{10} = (1011001, 0001)_2$.
   (b) $(110, 23046875)_{10}$ vers la base seize ; $(110)_{10} = (6E)_{16}$. Puis on conversion de la partie décimale : $0, 23046875 \times 16 = 3, 6875$, on conserve $a_{-1} = 3$, et on continue avec $0, 6875$, soit $0, 6875 \times 16 = 11, 0$, on arrête et on conserve $a_{-2} = (11)_{10} = (B)_{16}$. On a donc $(110, 23046875)_{10} = (6E, 3B)_{16}$.
   (c) $(55, 616)_{10}$ vers la base cinq. $(55)_{10} = (210)_5$. Puis $0, 616 \times 5 = 3, 08$ ($a_{-1} = 3$), $0, 08 \times 5 = 0, 4$ ($a_{-2} = 0$), $0, 4 \times 5 = 2, 0$ donc $a_{-3} = 2$. On a $(55, 616)_{10} = (210, 302)_5$.

3 Exercice n^o 3 : Codage en complément à deux - passage de la base dix vers le binaire

Donner les représentations en complément à deux des nombres décimaux suivants. **Solution :** Rappelons la définition du codage des entiers relatifs en complément à deux. Pour les entiers positifs, le bit de poids fort est 0 (signe +), puis on utilise les $n-1$ autres bits pour coder en binaire la valeur de l'entier. Pour les entiers négatifs, on code sur n bits (quitte à mettre des zéros à gauche) la valeur absolue de l'entier en binaire, puis on complémente tous les bits ($0 \leftrightarrow 1$), et enfin on ajoute 1 (on effectue une addition entre nombres binaires, donc attention aux retenues !).

1. 122 sur un octet ; $(122)_{10} = (1111010)_2$, donc le codage en complément à deux sur un octet est (01111010).
2. 2025 sur seize bits. $(2025)_{10} = (11111101001)_2$, donc en complément à deux sur 16 bits son codage est (0000011111101001).
   Peut-on coder ce nombre sur douze bits ? **Oui**, puisque on peut coder l'entier 2025 sur 11 bits. Sa représentation est alors (011111101001).
   sur onze bits ? **Non** car on a besoin d'un bit pour le signe !
3. -78 sur deux octets ; $(78)_{10} = (1001110)_2$. Sur 16 bits, on a donc (0000000001001110). On complémente les bits (1111111110110001), puis on ajoute 1 (somme binaire) : $(1111111110110001) + 1 = (1111111110110010)$.
4. -700 sur deux octets. $(700)_{10} = (1010111100)_2$. Donc sur 16 bits on a (0000001010111100). On complémente les bits (1111110101000011), et on ajoute 1 : $(1111110101000011) + 1 = (1111110101000100)$.

4 Exercice n^o 4 : Codage en complément à deux - passage du binaire vers la base dix

Donner les représentations décimales des nombres binaires suivants codés en complément à deux. **Solution :**

1. (00110101) (codé sur un octet) ; Le codage commence par 0 (bit de poids fort pour le signe +), c'est donc un entier positif qui est représenté. Sa valeur est codée en binaire sur les 7 autres bits. On a $(0110101)_2 = (53)_{10}$. Il s'agit donc de l'entier +53.
2. (0111010110001101) (codé sur deux octets) ; Le codage commence par 0 à gauche donc c'est un entier positif. Sa valeur en base dix est donnée par $(111010110001101)_2$. On peut la trouver par exemple en passant par la base 16, on regroupe les bits par paquets de quatre (de la droite vers la gauche) $(111010110001101)_2 = (0111\ 0101\ 1000\ 1101)_2$, puis on convertit chaque paquet en hexadécimal $(1101)_2 = (C)_{16}$, $(1000)_2 = (8)_{16}$, $(0101)_2 = (5)_{16}$ et $(0111)_{16} = (7)_{16}$. On a donc $(111010110001101)_2 = (758C)_{16} = 7 \times 16^3 + 5 \times 16^2 + 8 \times 16 + 13 = (30093)_{10}$. On a donc représenté l'entier +30093.
3. (10100110) (codé sur un octet). La représentation commence par 1 : c'est donc un entier négatif qui est codé. On soustrait 1, c'est-à-dire on calcule (dans la base deux) $(10100110) - 1 = (10100101)$. Puis on complémente les bits de façon à obtenir (01011010). Ceci est la représentation en base deux de la valeur absolue de l'entier recherché. Or $(01011010)_2 = (90)_{10}$. On a donc la représentation en complément à deux (sur un octet) de -90.