TD 3

Codage des nombres

Dans cette partie, le petit encadré « **à savoir !** » est utilisé au début des paragraphes qui peuvent être considérés comme du cours.

Exercice 1 : Représentation positionnelle des entiers naturels

à savoir ! Soit $\beta \in \mathbf{N}$, $\beta > 1$, une base. Tout $n \in \mathbf{N}$ peut être représenté de manière unique par sa *représentation positionnelle en base* β :

$$(x_{p-1}x_{p-2}\cdots x_1x_0)_\beta := \sum_{i=0}^{p-1} x_i\beta^i.$$

Les $x_i \in \{0, 1, \ldots, \beta-1\}$ sont les *chiffres* de l'écriture de n en base β, p est le nombre de chiffres nécessaires pour écrire de l'entier naturel n. On attribue un symbole à chaque chiffre : chiffres 0 et 1 en binaire, chiffres de 0 à 9 en décimal, chiffres de 0 à F en hexadécimal.

1) La définition permet directement d'effectuer des changements de base : si on connaît l'écriture $(x_{p-1}\ldots x_0)_\beta$ de n, et qu'on veut connaître cette écriture en base γ, il suffit de convertir les chiffres x_i en base γ, puis de calculer $\sum_{i=0}^{p-1} x_i\beta^i$ en base γ. En utilisant cette technique, convertissez $(11011)_2$ puis $(56)_9$ en décimal.
2) Une autre technique de changement de base est celle des divisions euclidiennes successives. Rappelez et justifiez cette méthode. Utilisez là pour convertir $n = (423)_{10}$ en binaire, puis $(3452)_{10}$ en base 8.
3) Entre les bases 2, 8 et 16, des méthodes plus directes peuvent être utilisées : par exemple, tout chiffre octal est représenté par un entier sur trois bits, et tout entier sur trois bits est représenté par un chiffre octal. Justifiez cette méthode de convertion entre les bases 2 et 8. Convertissez $(34521)_8$ en base 2 puis 16.

Exercice 2 : Représentation en complément à deux sur p bits

Partie 1 : Définition

à savoir ! Soit un entier relatif n à coder en machine sur p bits. On adopte la définition suivante pour le codage de n en complément à 2 sur p bits :

$$(c_{p-1}c_{p-2}\ldots c_1c_0)_{\bar{2}} := -c_{p-1}2^{p-1} + \sum_{i=0}^{p-2} c_i 2^i.$$

Une manière d'interpréter le codage en complément à 2 est donc de considérer que le bit le plus à gauche a un poids négatif (-2^{p-1}). On peut montrer que $n \geq 0$ ssi $c_{p-1} = 0$, et $n < 0$ ssi $c_{p-1} = 1$.

1) Quelle est la valeur décimale codée par $(10000011)_{\bar{2}}$? Par $(00001010)_{\bar{2}}$?
2) Comment coder $(-120)_{10}$ en complément à 2 sur 8 bits ?
3) Quelle est le plus grand entier représentable en complément à 2 sur p bits ? Le plus petit ?

Partie 2 : Addition et calcul de l'opposé

à savoir ! Soient $m = (c_m)_{\bar{2}}$ et $n = (c_n)_{\bar{2}}$ en complément à 2 sur p bits. On dit qu'il y a dépassement de capacité dans une opération en complément à 2 sur p bits lorsque le résultat de l'opération ne peut pas être représenté sous cette même forme (il est soit trop petit, soit trop grand).

- En l'absence de dépassement de capacité, le codage de $m+n$ en complément à 2 sur p bits est le codage de l'entier naturel $(c_m + c_n) \mod 2^p$. Si m et n sont de même signe, il y dépassement ssi le signe du résultat calculé diffère du signe des opérandes ; s'ils sont de signes opposés, aucun dépassement n'est possible.
- En l'absence de dépassement de capacité, le codage de $-n$ en complément à 2 sur p bits est le même que celui de l'entier naturel $(\bar{c}_{p-1}\bar{c}_{p-2}\ldots\bar{c}_1\bar{c}_0)_2 + 1 \mod 2^p$. Il y dépassement ssi le signe du résultat calculé est le même que celui de l'opérande.

1) Posez, en complément à deux sur 8 bits, les additions suivantes : $(10001010)_{\overline{2}} + (00001011)_{\overline{2}}$, $(10001010)_{\overline{2}} + (10001011)_{\overline{2}}$, $(01001010)_{\overline{2}} + (11001010)_{\overline{2}}$.

2) Calculez l'opposé de $(10001010)_{\overline{2}}$ en complément à 2 sur 8 bits, et vérifiez que votre résultat est correct.

3) En complément à 2 sur p bits, quel est le seul cas produisant un dépassement de capacité pour le calcul de l'opposé ?

Partie 3 : Extension de signe en complément à 2

1) Comment sont représentés $(34)_{10}$ et $(-42)_{10}$ en complément à 2 sur 8 bits (complément à 2^8) ?

2) Comment sont représentés $(34)_{10}$ et $(-42)_{10}$ en complément à 2 sur 12 bits (complément à 2^{12}) ?

3) Proposez et justifiez une règle permettant de passer de l'écriture d'un entier relatif en complément à 2 sur p bits à son écriture en complément à 2 sur $p+k$ bits (en conservant l'égalité des valeurs bien-entendu).

Exercice 3 : Codage de nombres en machine

1) Convertir les nombres suivants en base 10 : $(1011)_2$, $(10110)_2$, $(101.1)_2$, $(0.1101)_2$, $(110.01)_2$.

2) Convertir les nombres suivants en base 10 : $(FF)_{16}$, $(1A)_{16}$, $(789)_{16}$, $(0.13)_{16}$, $(ABCD.EF)_{16}$.

3) Convertir les nombres suivants (exprimés en base 10) en base 2 et en base 16 : 12, 24, 192, 2079, 0.25, 0.375, 0.376 et 17.150.

Exercice 4 : Conversion base 10 vers base 2 d'entiers naturels

Le principe de l'algorithme conversion de Horner de l'entier naturel $n = (x_3x_2x_1x_0)_\beta$ est le suivant. On a :

$$n = x_3{\cdot}\beta^3 + x_2{\cdot}\beta^2 + x_1{\cdot}\beta^1 + x_0, = (x_3{\cdot}\beta^2 + x_2{\cdot}\beta^1 + x_1){\cdot}\beta + x_0, = ((x_3{\cdot}\beta + x_2){\cdot}\beta + x_1){\cdot}\beta + x_0.$$

Si on exécute l'algorithme suivant,

$r_3 = x_3$
$r_2 = r_3 \times \beta + x_2$
$r_1 = r_2 \times \beta + x_1$
$r_0 = r_1 \times \beta + x_0$

alors par construction on a l'égalité $r_0 = x_3{\cdot}\beta^3 + x_2{\cdot}\beta^2 + x_1{\cdot}\beta^1 + x_0$. Si on effectue tous les calculs dans une base d'arrivée γ, on obtient l'écriture de n en base γ.

1) Testez l'algorithme de Horner pour convertir l'entier $n = (11010101)_2$ en décimal.

2) Plus fastidieux à faire à la main : testez l'algorithme de Horner pour convertir l'entier $n = (567)_{10}$ en binaire. Combien de multiplications avez vous posées ? Combien de multiplications auriez vous posées si vous aviez utilisé la définition de la notation positionnelle pour faire cette conversion ?

3) Ecrivez maintenant deux fonctions en `C` pour convertir une chaîne de caractères (tableau de codes ASCII terminé par un `0`) contenant l'écriture d'un entier positif en décimal, en un entier non-signé en machine [1] : l'une utilisera naïvement la définition de la notation positionnelle, l'autre l'algorithme de Horner. Dans les deux cas, combien de multiplications sont utilisées ?

4) Pour poursuivre : dans les deux fonctions de conversion, peut-il se produire un dépassement de capacité ? Le cas échéant, que vont retourner ces fonctions ? Dans la même veine, mettre au point une fonction pour convertir un entier naturel écrit en hexadécimal vers un entier naturel du type `unsigned int`.

Exercice 5 : Représentation positionnelle des rationnels

à savoir ! Les rationnels sont les nombres de la forme $\frac{p}{q}$, avec $p \in \mathbf{Z}$ et $q \in \mathbf{N} - \{0\}$. Tout $x \in \mathbf{Q}$ positif peut être décomposé en une partie entière $\lfloor x \rfloor \in \mathbf{N}$ ($\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$) et une partie fractionnaire $\{x\} = x - \lfloor x \rfloor$ ($0 \leq \{x\} < 1$). On utilise la notation positionnelle pour l'écriture de $\{x\}$: s'il existe $q \in \mathbf{N}$ t.q.

$$\{x\} = (0, x_{-1} \dots x_{-q})_\beta = \sum_{i=1}^{q} x_{-i}\beta^{-i},$$

1. Vous avez peut-être déjà utilisé les fonctions `sscanf` ou `atoi` du C99 pour faire cela, mais il s'agit ici de s'en passer.

alors $x = (x_{p-1}x_{p-2}\dots x_0,x_{-1}x_{-2}\dots x_{-q})_\beta$ est l'écriture x en base β. Le problème est que l'écriture d'un rationnel en base β n'est pas forcément finie : en machine, il faut souvent se contenter d'une approximation. Par contre, on sait que l'écriture d'un rationnel est nécessairement périodique.

à savoir ! Soit un rationnel $0 \leq x < 1$. On veut déterminer son écriture $(0,x_{-1}x_{-2}x_{-3}\dots)_2$ en binaire. On a $2 \times x = (x_{-1},x_{-2}x_{-3}\dots)_2$, donc $x_{-1} = \lfloor 2 \times x \rfloor$. En procédant par des multiplications successives par 2, on peut ainsi extraire un par un les bits de l'écriture (binaire) de x.

1) Quelle est l'écriture de 13/7 en base $\beta = 10$? L'écriture est périodique, on soulignera la période.

2) Convertir $1/10 = (0,1)_{10}$, puis $(5,3)_{10}$ en écriture binaire.

3) Donnez l'écriture décimale de $(11,1001)_2$.

4) Donnez une écriture sous la forme d'une fraction de deux nombres décimaux de $(0,0\underline{101})_2$.