Représentation des nombres

Algo & Prog avec R

Arnaud Malapert

2 octobre 2020

Université Côte d'Azur, CNRS, I3S, France
firstname.lastname@univ-cotedazur.fr

Cours en AUTOFORMATION

Représentation des nombres

- Système positionel : binaire, décimal, octal, et héxadécimal.
- Nombre non signé seulement (positif).
- Représentation en machine des entiers naturels seulement !

Prérequis
Savoir additionner, soustraire, multiplier, et diviser ! Surtout par 2 !

Évaluation : Gagner un max de points en peu de temps !

- 3 points du même QCM dans le contrôle terminal.
- Exercices de programmation autour des algorithmes de conversion.
- Activité de programmation d'une appli web de conversion.

Table des matières

Système positionnel

Représentation des nombres

On représente les nombres grâce à des symboles.

Représentation unaire : un symbole de valeur unique.

- I=1, II = 2, III = 3, let IIIIIIIII = 10.
- Le calcul est facile.
  - I + III = IIII ;
  - II × III = II II II = IIIIII
- mais cela devient vite *incompréhensible*

Chiffres Romains : plusieurs symboles ayant des valeurs différentes.

- Le nombre de symboles est théoriquement infini.
  - I=1, V=5, X = 10, L = 50 . . ..
- Le calcul est impossible.

Représentation des nombres

Système positionnel : symboles dont la valeur dépend de la position

- $999 = 900 + 90 + 9$
- À Babylone, système sexagésimal (60) (IIe millénaire av J-C).
- Transmission de l'orient vers l'occident avec le zéro (env. 825 ap. J-C)[1]

Un brin de cynisme

> *Les hommes sont comme les chiffres, ils n'acquièrent de la valeur que par leur position.*
>
> *Napoléon Bonaparte*

1. « Al-jabr wa'l-muqâbalah » Muhammad ibn Müsä al-Khuwärizmï

Système positionnel

Utilisation d'une base b

- Les nombres sont représentés à l'aide de b symboles distincts.
- La valeur d'un chiffre dépend de la base.

Un nombre x est représenté par une suite de symboles :

$$x = a_n a_{n-1} \dots a_1 a_0.$$

$$\begin{aligned} \textbf{Décimale} &\ (b = 10),\ a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ \textbf{Binaire} &\ (b = 2),\ a_i \in \{0, 1\} \\ \textbf{Hexadécimale} &\ (b = 16),\ a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\} \end{aligned}$$

Les bases les plus utilisées sont : 10, 2, 3, 2^k , 12, 16, 60, $\frac{\sqrt{5}-1}{2}$ ...

Notation

$(x)_b$ indique que le nombre x est écrit en base b.

Système positionnel

Entiers naturels

Représentation des entiers naturels

En base b

$$x = a_n a_{n-1} \dots a_1 a_0 = \sum_{i=0}^{n} a_i b^i$$

- a_0 est le chiffre de poids faible,
- a_n est le chiffre de poids fort.

En base 10

$$(1998)_{10} = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 8 \times 10^0.$$

En base 2

$$\begin{aligned}(101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 4 + 0 + 1 = 5.\end{aligned}$$

Traduction vers la base 10

Méthode simple

- On applique simplement la formule.
- Cela revient donc à une simple somme.
- En pratique, on peut utiliser la multiplication égyptienne.

Schéma de Horner

- Méthode générale pour calculer l'image d'un polynôme en un point.
- Moins d'opération que la méthode simple.
- Plus efficace pour une machine, pas nécessairement pour un humain.

Schéma de Horner

Une reformulation judicieuse de l'écriture en base b

$$\begin{aligned} a_n a_{n-1} \ldots a_1 a_0 &= \sum_{i=0}^{n} a_i b^i \\ &= ((\ldots((a_n b + a_{n-1})b + a_{n-2})\ldots)b + a_1)b + a_0 \end{aligned}$$

Algorithme simple et efficace

- Initialiser l'accumulateur : $v = 0$.
- Pour chaque chiffre a_i en partant de la gauche : $v \leftarrow (v \times b) + a_i$.

Schéma de Horner

Algorithme simple et efficace

- Initialiser l'accumulateur : $v = 0$.
- Pour chaque chiffre a_i en partant de la gauche : $v \leftarrow (v \times b) + a_i$.

$(10110)_2 = (22)_{10}$

a_i	v	Calcul de v
1	1	$2 \times 0 + 1$
0	2	$2 \times 1 + 0$
1	5	$2 \times 2 + 1$
1	11	$2 \times 5 + 1$
0	22	$2 \times 11 + 0$

$(12021)_3 = (142)_{10}$

a_i	v	Calcul de v
1	1	$3 \times 0 + 1$
2	5	$3 \times 1 + 2$
0	15	$3 \times 5 + 0$
2	47	$3 \times 15 + 2$
1	142	$3 \times 47 + 1$

Traduction entre des puissances de 2

Du binaire vers une base 2^k
Regrouper les bits par k en partant de la droite et les traduire.

D'une base 2^k vers le binaire
Traduire chacun des symboles en un nombre binaire.

Du binaire vers l'héxadécimal (2^4)
Regrouper les bits par 4 en partant de la droite.

(00)10	0101	1110	0001	1101	1111
2	5	E	1	D	F

Du binaire vers l'octal (2^3)
Regrouper les bits par 3 en partant de la droite.

(00)1	001	011	110	000	111	011	111
1	1	3	6	0	7	3	7

D'une base 2^k vers une base 2^p

1. Passer par le binaire : $(x)_{2^k} \to (x)_2 \to (x)_{2^p}$.
2. Généraliser la méthode précédente si k est un diviseur/multiple de p.
3. Appliquer un algorithme de traduction vers une base quelquonque.

Traduction vers une base quelquonque

Nombre entier
On procède par divisions euclidiennes successives :

- On divise le nombre par la base,
- puis le quotient par la base,
- ainsi de suite jusqu'à obtenir un quotient nul.

La suite des restes obtenus correspond aux chiffres de a_0 à a_n dans la base visée.

$(44)_{10} = (101100)_2$

$$\begin{array}{rlll} 44 &= 22 \times 2 + 0 & & a_0 = 0 \\ 22 &= 11 \times 2 + 0 & & a_1 = 0 \\ 11 &= 5 \times 2 + 1 & & a_2 = 1 \\ 5 &= 2 \times 2 + 1 & & a_3 = 1 \\ 2 &= 1 \times 2 + 0 & & a_4 = 0 \\ 1 &= 0 \times 2 + 1 & & a_5 = 1 \end{array}$$

$(44)_{10} = (1122)_3$

$$\begin{array}{rlll} 44 &= 14 \times 3 + 2 & & a_0 = 2 \\ 14 &= 4 \times 3 + 2 & & a_1 = 2 \\ 4 &= 1 \times 3 + 1 & & a_2 = 1 \\ 1 &= 0 \times 3 + 1 & & a_3 = 1 \end{array}$$

Traduction depuis une base quelquonque

Il faut diviser dans la base d'origine.
Les calculs sont donc difficiles pour un humain !

$(101100)_2 = (1122)_3$

$$\begin{aligned} 101100 &= 1110 \times 11 + 10 & a_0 &= 2 \\ 1110 &= 100 \times 11 + 10 & a_1 &= 2 \\ 100 &= 1 \times 11 + 1 & a_2 &= 1 \\ 1 &= 0 \times 11 + 1 & a_3 &= 1 \end{aligned}$$

Passez par le décimal !
$(x)_b \to (x)_{10} \to (x)_{b'}$.

Exercices

- Traduire $(10101)_2$ en écriture décimale.
- Traduire $(10101101)_2$ en écriture décimale.
- Traduire $(1010100111010101)_2$ en écriture hexadécimale.
- Traduire $(1AE3F)_{16}$ en écriture binaire.
- Traduire $(1AE3F)_{16}$ en écriture octal.
- Traduire $(927)_{10}$ en écriture binaire.
- Traduire $(1316)_{10}$ en écriture binaire.

Système positionnel

Nombres fractionnaires

Représentation des nombres fractionnaires

En base b
La formule est la même, mais il existe des exposants négatifs.

$$x = a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-k} = \sum_{i=-k}^{n} a_i b^i$$

En base 10

$$19.98 = 1 \times 10^1 + 9 \times 10^0 + 9 \times 10^{-1} + 8 \times 10^{-2}.$$

En base 2

$$\begin{aligned}(101,01)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 4 + 0 + 1 + 0 + 0.25 = 5,25.\end{aligned}$$

Traduction vers une base quelquonque

Nombre fractionnaire

- On décompose le nombre en partie entière et fractionnaire si $x > 0$:

$$x = E[x] + F[x].$$

- On convertit la partie entière par la méthode précédente :

$$E[x] = a_n a_{n-1} \dots a_1 a_0.$$

- On convertit la partie fractionnaire :

$$F[x] = 0, a_{-1} a_{-2} \dots a_{-m}.$$

- Finalement, on additionne la partie entière et fractionnaire :

$$x = a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} \dots.$$

Partie fractionnaire

- on multiplie $F[x]$ par b. Soit a_{-1} la partie entière de ce produit,
- on recommence avec la partie fractionnaire du produit pour obtenir a_{-2},
- et ainsi de suite ...
- on stoppe l'algorithme si la partie fractionnaire devient nulle.

$(0,734375)_{10} = (0,BC)_{16}$

$$\begin{aligned} 0,734375 \times 16 &= 11,75 & a_{-1} &= B \\ 0,75 \times 16 &= 12 & a_{-2} &= C \end{aligned}$$

Traduction de 0,3 en base 2

$(0,3)_{10} = (0,01001100110011\ldots)_2$

$0,3 \times 2 = 0,6 \quad a_{-1} = 0$

$0,6 \times 2 = 1,2 \quad a_{-2} = 1$

$0,2 \times 2 = 0,4 \quad a_{-3} = 0$

$0,4 \times 2 = 0,8 \quad a_{-4} = 0$

$0,8 \times 2 = 1,6 \quad a_{-5} = 1$

$0,6 \times 2 = 1,2 \quad a_{-6} = 1$

$0,2 \times 2 = 0,4 \quad a_{-7} = 0$

La conversion d'un nombre fractionnaire ne s'arrête pas toujours.

- En base b, on ne peut représenter exactement que des nombres fractionnaires de la forme X/b^k
- La conversion d'un nombre entier s'arrête toujours.

Il faudra arrondir ...

Exercices

- Traduire $(10101101)_2$ en écriture décimale.
- Traduire $(10101001110101101)_2$ en écriture hexadécimale.
- Traduire $(1AE3F)_{16}$ en écriture décimale.
- Traduire $(1AE3F)_{16}$ en écriture binaire.
- Traduire $(13.1)_{10}$ en écriture binaire.

Multiplication et division égyptiennes

Multiplication égyptienne

Principe

- Décomposition d'un des nombres en une somme.
  On décompose généralement le plus petit.
- Création d'une table de puissance pour l'autre nombre
- Très souvent, traduction du décimal vers le binaire.
  Il existe des variantes en fonction de la complexité de l'opération.
- Il suffit de savoir multiplier par deux et additionner !

Multiplication égyptienne : $x \times y$

- On construit la ligne $i + 1$ en multipliant par deux 2^i et $x \times 2^i$ tant que $2^{i+1} < y$.

$189 \times 21 = 3969$

i	a_i	2^i	189×2^i
0		1	189
1		2	378
2		4	756
3		8	1512
4		16	3024

Multiplication égyptienne : $x \times y$

- On construit la ligne $i+1$ en multipliant par deux 2^i et $x \times 2^i$ tant que $2^{i+1} < y$.
- On traduit 21 en binaire en remontant dans le tableau.
- On calcule $\sum_0^n a_i(x \times 2^i)$.

$189 \times 21 = 3969$

i	a_i	2^i	189×2^i		
0	1	1	189	✓	(1 -1 = 0)
1	0	2	378		
2	1	4	756	✓	(5 -4 = 1)
3	0	8	1512		
4	1	16	3024	✓	(21 -16 = 5)
			3969		

Méthode genérale

Des opérations plus complexes faisant intervenir par exemple des fractions exigeaient une décomposition avec :

- les puissances de deux,
- les fractions fondamentales,
- les dizaines.

La technique est rigoureusement la même mais offre plus de liberté au scribe quant à la décomposition du petit nombre.

$243 \times 27 = 6561$

1	243
2	486
4	972
20	4860
27	6561

Papyrus Rhind : $\frac{1}{14} \times \frac{7}{4} = \frac{1}{8}$

1	$\frac{1}{14}$	
$\frac{1}{2}$	$\frac{1}{28}$	
$\frac{1}{4}$	$\frac{1}{56}$	
$\frac{7}{4}$	$\frac{1}{8}$	$\left(\frac{4+2+1}{56}\right)$

Division égyptienne : $x \div y$

Par quoi doit-on multiplier y pour trouver x ?

- On construit la ligne $i + 1$ en multipliant par deux 2^i et $y \times 2^i$ tant que $y \times 2^{i+1} < x$.

$539 \div 7 = 77$

i	2^i	7×2^i
0	1	7
1	2	14
2	4	28
3	8	56
4	16	112
5	32	224
6	64	448

Division égyptienne : $x \div y$

Par quoi doit-on multiplier y pour trouver x ?

- On construit la ligne $i+1$ en multipliant par deux 2^i et $y \times 2^i$ tant que $y \times 2^{i+1} < x$.
- On décompose x par les $y \times 2^i$ en remontant dans le tableau.
- On calcule la somme des 2^i de la décomposition.

$539 \div 7 = 77$

i	2^i	7×2^i		
0	1	7	✓	$(7 - 7 = 0)$
1	2	14		
2	4	28	✓	$(35 - 28 = 7)$
3	8	56	✓	$(91 - 56 = 35)$
4	16	112		
5	32	224		
6	64	448	✓	$(539 - 448 = 91)$
	77			

Division dont le résultat est fractionnaire

$234 \div 12 = 19.5$

i	2^i	12×2^i
0	1	12
1	2	24
2	4	48
3	8	96
4	16	192

Division dont le résultat est fractionnaire

$234 \div 12 = 19.5$

i	2^i	12×2^i		
0	1	12	✓	(6)
1	2	24	✓	(18)
2	4	48		
3	8	96		
4	16	192	✓	(42)

Division dont le résultat est fractionnaire

$234 \div 12 = 19.5$

i	2^i	12×2^i		
-1	$\frac{1}{2}$	6	✓	(0)
0	1	12	✓	(6)
1	2	24	✓	(18)
2	4	48		
3	8	96		
4	16	192	✓	(42)
	19.5			

Exercices

1. Multipliez 187 par 11.
2. Multipliez 2012 par 1515 (indice : utilisez la méthode genérale).

Exercices

1. Multipliez 187 par 11.
2. Multipliez 2012 par 1515 (indice : utilisez la méthode genérale).

187×11

i	a_i	2^i	11×2^i
0	1	1	187
1	1	2	374
2	0	4	748
3	1	8	1496
		11	2057

2012×1515

5	10060
10	20120
500	1006000
1000	2012000
1515	3048180

Arithmétique binaire

Représentation de l'information en machine

- Informations en général représentées et manipulées sous forme binaire.
- L'unité d'information est le chiffre binaire ou bit (binary digit).
- Les opérations arithmétiques de base sont faciles à exprimer en base 2.
- La représentation binaire est facile à réaliser : systèmes à deux états obtenus à l'aide de transistors.

Addition binaire

Tables d'addition

- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $1 + 1 = 10$ (0 et on retient 1)

$91 + 71 = 162$

$$
\begin{array}{r}
{\scriptstyle 1\;\;\;\;1\;1\;1\;1\;1\;\;\;\;} \\
1\,0\,1\,1\,0\,1\,1 \\
+\,1\,0\,0\,0\,1\,1\,1 \\
\hline
1\,0\,1\,0\,0\,0\,1\,0
\end{array}
$$

Soustraction binaire

Tables de soustraction

- $0 - 0 = 0$
- $1 - 0 = 1$
- $(1)0 - 1 = 1$ (1 et on retient 1)
- $1 - 1 = 0$

$83 - 79 = 4$

```
     1 1
 1 0 1 0 0 1 1
-1 0 0 1 1 1 1
    1 1
--------------
 0 0 0 0 1 0 0
```

Limitation : résultat négatif
On ne peut traiter $x - y$ que si $x \geq y$.

Multiplication binaire

Tables de multiplication

- $0 \times 0 = 0$
- $1 \times 0 = 0$
- $1 \times 1 = 1$

$23 \times 11 = 253$

$$
\begin{array}{rr}
 & 1\,0\,1\,1\,1 \\
\times & 1\,0\,1\,1 \\
\hline
 & {\scriptstyle 1\,1\,1\,1\,1} \\
 & 1\,0\,1\,1\,1 \\
+ & 1\,0\,1\,1\,1 \\
+ & 1\,0\,1\,1\,1 \\
\hline
 & 1\,1\,1\,1\,1\,1\,0\,1
\end{array}
$$

Division binaire

Soustractions et décalages comme la division décimale

- sauf que les digits du quotient ne peuvent être que 1 ou 0.
- Le bit du quotient est 1 si on peut soustraire le diviseur, sinon il est 0.

$231 \div 11 = 21$

$$\begin{array}{r|l}
1\,1\,1\,0\,0\,1\,1\,1 & 1011 \\ \cline{2-2}
-\underline{1\,0\,1\,1} & 10101 \\
1\,1\,0\,1 & \\
-\underline{1\,0\,1\,1} & \\
1\,0\,1\,1 & \\
-\underline{1\,0\,1\,1} & \\
0 &
\end{array}$$

Limitation : division entière
Pour l'instant, on ne peut pas calculer la partie fractionnaire.

Exercices

1. Traduire $(1100101)_2$ et $(10101111)_2$ en écriture décimale.
2. Calculez la somme $(1100101)_2 + (10101111)_2$.
3. Calculez le produit $(10101111)_2 \times (1100101)$.
4. Calculez la soustraction $(10101111)_2 - (1100101)_2$.
5. Vérifier tous les résultat en les traduisant en écriture décimale.

Représentation des nombres en machine

Représentation des nombres en machine

Précision finie

- Codés généralement sur 16, 32 ou 64 bits.
- Un codage sur n bits permet de représenter 2^n valeurs distinctes.

Un ordinateur ne calcule pas bien !

- Pour un ordinateur, le nombre de chiffres est fixé.
- Pour un mathématicien, le nombre de chiffres dépend de la valeur représentée.
- Lorsque le résultat d'un calcul doit être représenté sur plus de chiffres que ceux disponibles, il y a dépassement de capacité.

Représentation des entiers en machine

Entiers naturels

- Un codage sur n bits : tous les entiers entre 0 et $2^n - 1$.
- La conversion d'un nombre entier s'arrête toujours.
- On ne peut traiter $x - y$ que si $x \geq y$.

Entiers relatifs : plusieurs représentations existent.
Valeur absolue signée ; complément à 1 ; complément à 2.

Tous les processeurs actuels utilisent le complément à 2.

- il y a un seul code pour 0 ;
- l'addition de deux nombres se fait en additionnant leurs codes ;
- et il est très simple d'obtenir l'opposé d'un nombre.
- les processeurs disposent de fonctions spéciales pour l'implémenter.

Représentation des nombres réels

- Les ressources d'un ordinateur étant limitées, on représente seulement un sous-ensemble $\mathbb{F} \subset \mathbb{R}$ de cardinal fini.
- Les éléments de $\mathbb{F}$ sont appelés nombres à virgule flottante.
- Les propriétés de $\mathbb{F}$ sont différentes de celles de $\mathbb{R}$.
- Généralement, un nombre réel x est tronqué par la machine, définissant ainsi un nouveau nombre $fl(x)$ qui ne coïncide pas forcément avec le nombre x original.

Problèmes et limitations

- les calculs sont nécessairement arrondis.
- il y a des erreurs d'arrondi et de précision
- On ne peut plus faire les opérations de façon transparente

```
> 0.1 + 0.1 + 0.1 == 0.3
[1] FALSE
> 10^20 + 1 == 10^20
[1] TRUE
```

Questions?

Retrouvez ce cours sur le site web

www.i3s.unice.fr/~malapert/R